BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF RADIO ELECTRONICS

ÚSTAV RADIOELEKTRONIKY

DEVELOPMENT OF ALGORITHMS FOR GUNSHOT DETECTION

VÝVOJ ALGORITMŮ PRO ROZPOZNÁVÁNÍ VÝSTŘELŮ

DOCTORAL THESIS
DIZERTAČNÍ PRÁCE

AUTHOR
AUTOR PRÁCE
Ing. Martin Hrabina

SUPERVISOR
ŠKOLITEL
prof. Ing. Milan Sigmund, CSc.

BRNO 2019

ABSTRAKT

Táto práca sa zaoberá rozpoznávaním výstrelov a pridruženými problémami. Ako prvé je celá vec predstavená a rozdelená na menšie kroky. Ďalej je poskytnutý prehľad zvukových databáz, významné publikácie, akcie a súčasný stav veci spoločne s prehľadom možných aplikácií detekcie výstrelov. Druhá časť pozostáva z porovnávania príznakov pomocou rôznych metrík spoločne s porovnaním ich výkonu pri rozpoznávaní. Nasleduje porovnanie algoritmov rozpoznávania a sú uvedené nové príznaky použiteľné pri rozpoznávaní. Práca vrcholí návrhom dvojstupňového systému na rozpoznávanie výstrelov, monitorujúceho okolie v reálnom čase. V závere sú zhrnuté dosiahnuté výsledky a načrtnutý ďalší postup.

KĽÚČOVÉ SLOVÁ

Výber príznakov, šum, rozpoznávanie výstrelov, linearárne prediktívne kódovanie, kepstrálne koeficienty, zvuková databáza

ABSTRACT

This work deals with gunshot recognition and problems connected to it. Firstly, the problem is briefly introduced and broken down to smaller steps. Next, overview of datasets is provided, relevant information sources and publications in this field, and state-of-the-art along with possible applications of gunshot recognition. The second part consists of feature selection and performance comparison. Next, sound recognition algorithms are introduced and compared, along with novel features suitable for gunshot detection. The work culminates in creating two stage gunshot detection system, with real time audio event detection. The conclusion sums up achieved results and sketches possible steps to consider for hardware realization.

KEYWORDS

Feature selection, noise, gunshot recognition, linear predictive coding, mel-frequency cepstral coefficients, audio dataset

HRABINA, M. *Development of algorithms for gunshot detection*: doctoral thesis. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Radio Electronics, 2019. 85 p. Supervised by: prof. Ing. Milan Sigmund, CSc.

DECLARATION

I declare that I have written my treatise on doctoral thesis on theme of “Development of algorithms for gunshots detection” independently, under the guidance of the treatise on doctoral thesis supervisor and using technical literature and other sources of information which which are all quoted in the treatise and detailed in the list of literature at the end of the treatise.

As the author of the treatise on doctoral thesis I furthermore declare, that as regards the creation of this treatise on doctoral thesis, I have not infringed and copyright. In particular, I have not unlawfully encroached on anyone’s personal and/or ownership rights and I am fully aware of the consequences in the case of breaking Regulation § 11 and the following of the Copyright Act No 121/200 Sb., and of the rights related to intellectual property right and changes in some Acts (Intellectual Property Act) and formulated in later regulations, inclusive of the possible consequences resulting from the provisions of Criminal Act No 400/2009 Sb., Section, Head VI, Part 4.

Brno ....

(author’s signature)

ACKNOWLEDGEMENT

I would like to thank to my supervisor prof. Ing. Milan Sigmund, CSc. For his support and encouragement during my study and work.

Brno …………   ………………………….

(author’s signature)

SIX research centre
sensor, information and communication systems

Faculty of Electrical Engineering
and Communication

Brno University of Technology
Technicka 12, CZ-61600 Brno, Czech Republic

http://www.six.feec.vutbr.cz

Research described in this treatise on doctoral thesis has been implemented in the laboratories supported by the SIX project; reg. no. CZ.1.05/2.1.00/03.0072, operational program Výzkum a vývoj pro inovace.

Brno ....

(author’s signature)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

SPL	Sound Pressure Level
MFCC	Mel-Frequency Cepstral Coefficient
HMM	Hidden Markov Models
LPC	Linear Prediction Coefficients
SNR	Sound to Noise Ratio
ZCR	Zero-Crossing Rate
SVM	Support Vector Machine
GMM	Gaussian Mixture Models
PLP	Perceptual Linear Predictive [coefficients]
LPCC	Linear Prediction Cepstral Coefficients
PPV	Positive Predictive Value
TP	True Positives
FP	False Positives
FN	False Negatives
MI	Mutual Information
FT	Fourier Transform
TPR	True Positive Rate
TNR	True Negative Rate
BIC	Bayesian Information Criterion
SED	Sound Event Detection
NN	Neural Networks
AMDF	Average Magnitude Difference Function
LFCC	Linear Frequency Cepstral Coefficients
TDF	Time Domain Features

INTRODUCTION

Sound classification is a process of categorization of different sounds into classes that share common features. It is used with various types of sounds, ranging from automatic recognition of music genre, speaker and spoken content recognition to acoustic analysis of industrial processes and recognition of natural and artificial sounds (such as disturbances in environment or gunshot detection).

The main motivation for this work was an effort to develop a reliable gunshot detection algorithm with low computational demands. This algorithm would then be incorporated into tracking collars for protected wildlife and is supposed to prevent poaching by alerting authorities about illegal activities. Similar efforts were undertaken by other researchers using microphone arrays in protected parks.

Sound recognition comes in multiple steps. First of all, dataset representing sounds to be distinguished must be obtained. After the data is acquired and properly labeled, suitable features should be calculated. Those are supposed to sufficiently distinguish between various classes, this equals to low variability inside class and high variability between classes, which can be expressed as mutual information. Among frequently used features in sound event detection are mel-frequency cepstral coefficients (MFCC), linear predictive coefficients (LPC), various spectral characteristics, such as spectral band energy, and recently also MPEG-7 descriptors. While many features have high mutual information between them and class label, they can also have high mutual information between themselves, resulting in high redundancy and low added information with increasing feature count. Many feature extraction and selection methods exist, related to this is also dimensionality reduction, which aims to reduce the number of features while preserving information content. As an example of such dimension reduction techniques, we can name linear discriminant analysis (LDA) or principal component analysis (PCA). Ultimately, features are fed to recognition algorithm, which assorts input data into classes. Examples of commonly used algorithms are support vector machine (SVM), artificial neural networks (ANN) or Naive Bayes classifier.

The rest of the thesis is structured into two major parts. The first part consists of introducing basic theory, demands and methods used. These include basics of acoustics, important sources of information and publications in the field of sound event detection, demands on datasets and some frequently used datasets. Next, frequently used features are introduced, along with various methods of comparing them, and a comparison of their effectiveness under clean and noisy conditions. The first part concludes by introducing frequently used algorithms in sound event recognition. The second part consists of the contributed work itself. It presents newely proposed features and compares them to some previously used features. It also proposes a system for real-time event detection with preliminary categorization into isolated gunshots and gunshot bursts which uses fast and well established algorithms. Secondly, it proposes advanced algorithms, which use new features and are also more computationally demanding, to further refine preliminary results and increase their accuracy and reliability.

1 ACOUSTICS

This chapter is divided in two sections, the first one introduces basics of acoustics, its units, fundamental laws and behaviour of sounds in natural environment. The other describes gunshot acoustics in more detail.

1.1 General Acoustics

Sound is characterized as disturbances in pressure level, where pressure is measured in Pascal (Pa). Since human hearing is approximately logarithmic, pressure change is often measured as sound pressure level (SPL) in decibel (dB) with reference value $p_0 = 20$ µPa. Equation (1) shows conversion from Pascal do dB.

$$L_p = 20 \cdot \log_{10}\left(\frac{p}{p_0}\right), \tag{1}$$

where L_p is sound pressure level in dB and p is acoustic pressure in Pascal.

Since this work deals with gunshot detection, it is important to know sound levels at different distances from source. Given sound level L_{P1} at distance r_1, we can calculate sound level L_{P2} at distance r_2 with formula (2):

$$L_{p2} = L_{P1} + 20 \cdot \log_{10}\left(\frac{r_1}{r_2}\right), \tag{2}$$

formula (3) provides sound pressure p_2 at distance r_2 given pressure p_1 at distance r_1:

$$p_2 = p_1\left(\frac{r_1}{r_2}\right). \tag{3}$$

Fig. 1 and Fig. 2 depict dependencies from (2) and (3).

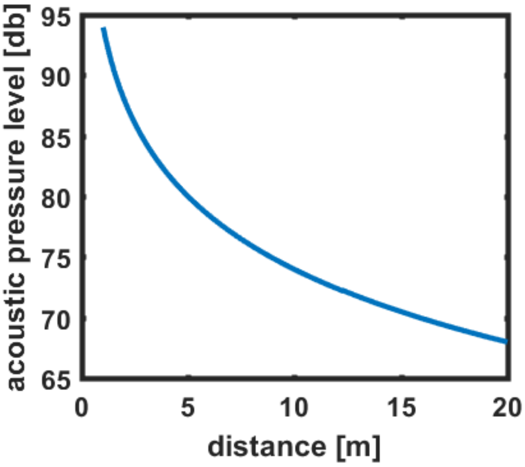

Fig. 1 Sound pressure level (dB) vs. distance

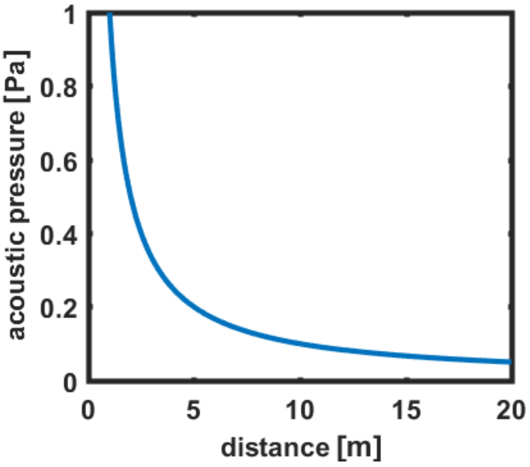

Fig. 2 Sound pressure (Pa) vs. distance

As noted in [1], the sound level at receiver depends also on other factors apart from the source (which entails both initial power and directional characteristic) and distance. These are environmental factors (i.e. medium, such as air, water or even solid objects), this work focuses on sound propagated in the air. Propagation of audio waves is influenced mainly by speed, which itself is influenced among others by: wind, temperature (and its gradient), humidity, and obstacles (along with floor), where we have to factor in reflections, diffraction, dispersion and absorption.

Sound of speed in dry air (0%) and 0°C is 331.2 m/s. The speed increases with growing density (i.e. propagates faster in water than in air). Due to non-ideal gas in which sound propagates, its speed is also decreased with increasing altitude, which causes its upward refraction. Formula (4) describes dependence of sound speed on air temperature in dry air (0%):

$$c_{air} = (331.2 + 0.606 \cdot T), \tag{4}$$

where T is temperature in °C.

Perception of sound is also characterized by frequency, with human audible spectrum lying in between 20 Hz and 20 kHz. The actual audible spectrum is influenced mainly by age and also by hearing damage. Sounds below this range are called infrasounds and sounds with higher frequency are called ultrasounds. Later in this work we examine influence of frequency range considered on accuracy of gunshot detection, and we will show that even more strictly limited frequency range achieves very good accuracy in gunshot detection. Apart from frequency limitations of human hearing, the audible events are also limited by loudness of an event, i.e. pressure variation. Lower limit is called "threshold of hearing" and is usually around 20 μPa or 0 dB_{SPL}. Maximum perceived loudness is limited by thresholds of pain, around 130 dB_{SPL} and subsequent hearing damage.

Analogous to human hearing, there are various specifications for microphones. First of all, we have to consider frequency range in which microphone reliably records sounds along with frequency characteristic. There is also microphone sensitivity, where we want to know either how much mV will be on the output for a given acoustic pressure in Pa, or the pressure can be compared to dB level referenced to 1 volt (i.e. dBV). The upper limit for microphones is given by amplifier overload in capacitor microphones, dynamic microphones do not have to deal with this issue, since they cannot be overloaded. Additionally, we have to consider directivity of microphones. This work is going to assume the use of omnidirectional microphones, since the acoustic event can come from any direction.

1.2 Gunshot Acoustics

This chapter captures the details of gunshot acoustics, describes gunshot waveform and provides details about sound levels.

Gunshot acoustics is being explored for several decades due to its use in forensics, while analyzing recordings from crime scenes to determine gunshot signatures and possibly also identify weapons used. It is also used in exact gunshot localization, for example in sniper localization systems, using microphone arrays.

The primary source of sound in gunshots is muzzle blast, which is the sound of expanding gases produced after discharging a weapon, this can be partially supressed using silencer. Typical muzzle blast lasts around 3 milliseconds and most of the acoustic energy is directed the same way as weapon barrel [2]. Weapons with supersonic bullets also produce shock wave, which has characteristic waveform, also known as N-wave. Shock wave propagates in the shape of cone trailing the bullet (which means no shock-wave is detected when recording behind the shooter), the angle of cone depends on Mach number of the bullet. Above mentioned publication also consider mechanical action of the weapon, which include sound of trigger and hammer, ejection of cartridges, etc. Mechanical action will not be considered in this work, as these signals are detectable only at a very short distance, and this work is dealing with gunshot detection over longer distances. Fig. 3 depicts and example of a gunshot (AK-47 assault rifle), featuring shock wave and muzzle blast, relative position of shock wave and muzzle blast is given by geometry of recording.

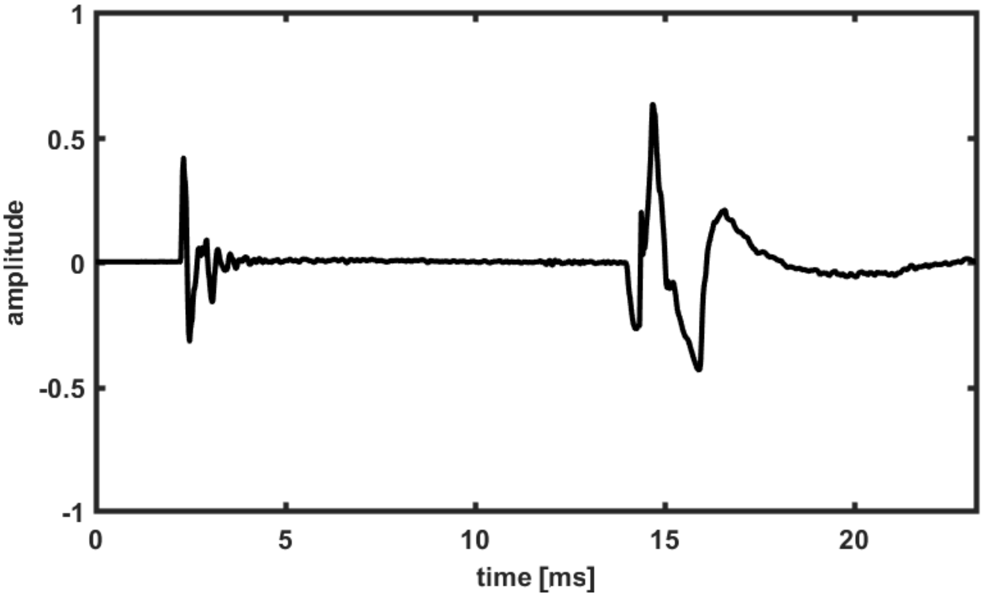

Fig. 3 Gunshot (AK-47) waveform

The duration and shape of muzzle blast depends both on ammunition and weapon. Shock wave shape is given only by bullet geomoetry and by its speed. More detailed look on gunshot waveforms can be found in [3]. However, in reverberant environments, the recording can provide more information about the terrain itself than about the weapon, since due to impulsive nature of muzzle blast and acoustic shockwave, we basically obtain convolution of gunshot signature and surrounding environment [2].

As to the sound pressure levels, gunshots are very loud. At around 1m distance, most surpass 150 dB [3] with some achieving as much as 167 dB. The measured sound level depends not only on distance, but also on azimuth.

2 SOUND EVENTS DATABASES

As we have already mentioned, first step in sound recognition is obtaining a properly labeled dataset. The following section briefly describes requirements in such datasets and then lists several datasets compiled for various purposes. It also separately lists datasets containing mainly gunshots, which are less common, and shows our choice of recordings for subsequent research. After the data is acquired, suitable features should be calculated which sufficiently distinguish between various classes, this equals to low variability inside class and high variability between classes, which can be expressed as mutual information. While many features have high mutual information between them and class label, they can also have high mutual information between themselves, resulting in high redundancy and low added information with increasing feature count. Many feature extraction and selection methods exist, these will be described and some of them applied in chapter 5. Ultimately, features are fed to recognition algorithm, which labels input data into classes.

The following section mentions multiple audio datasets compiled either for various sound recognition (such as sound event detection or audio scene classification) purposes or compiled for other purposes (such as movie industry), but also usable in the field of sound recognition. One subsection is solely dedicated to gunshot datasets, which are much rarer. The chapter concludes with information on what dataset used in this work consists of and technical information about the dataset, such as sampling frequency.

2.1 Existing sound event datasets

In order to make accurate automatic classification algorithm, broad database of sounds to be classified is needed. Some databases already exist for this purpose, others are created with different motivations (e.g. for movie sound effects). Automatic sound classification needs sounds as close to original as possible, this means without compression and additional modifications. Some existing databases will be mentioned below.

For purposes of sound recognition, database focused on urban sounds [4] was compiled from freely available, crowdsourced sound effects database - Freesound [5]. The compiled urban sound database consists of 10 sound classes (air conditioner, car horn, dog bark, drilling, engine, gunshots, playing children, jackhammer, siren and street music) with 27 hours of audio, including silent elements. This publication also offers taxonomy of urban sounds due to lack of common vocabulary. Another crowdsourced internet database, similar to [5], is Soundbible [6], this may contain, apart from real sounds, also synthetic and fantasy sounds, so care is necessary when selecting suitable recordings.

Some of the databases are dedicated to domestic and indoor sounds, for example in case of fall detection in elderly care. Supported by Netcarity project, this database consists of daily activities such as ironing, eating, watching television and their combinations [7]. Another database under this project is described in [8], it consists of 450 events with approx. 210 falls performed by 13 different actors. In this work, accelerometer and 3D camera data were collected for multimodality as well. The latest database in this category is based on 6.8 hours of various noises (bangs, crashes, household appliances) and speech (adult male and female, child) labeled as 9 different categories [9]. A special case of database is [10], it is focused purely on music. In order to avoid copyright issues, it does not contain songs or clips, but only extracted features (such as mel-frequency cepstral coefficients - MFCC) associated with individual songs. Due to size of this database (approx. 500 GB), user can directly download only 1% sample of randomly selected data, full database is accessible as amazon snapshot with detailed instruction provided on database website.

Apart from databases explicitly for scientific purposes, others are compiled and accessible. As a fist example, there are “British Library Sounds" [11], which present 80 000 recordings in various categories (from speech and dialects through world music to weather and natural sounds), available recordings are part of 3,5 million sounds held in the British Library. Collection of natural sounds in British Library is also described in [12]. This database has its origins around year 1900 and many analog recordings were subsequently digitized, so recording quality is difficult to estimate, however most recordings are digitized with 96 kHz sampling frequency and 24 bit quantization (mostly music and speech). This database offers many sounds to listen to for almost everybody (depending on copyright laws in given countries for certain recordings), but download is limited to staff and students of UK higher and further education institutions. Another commercial database [13] is created for movie making purposes, there are both free and paid collections consisting of recordings of crowds in different places and ambience sounds. Along with [9], DCASE 2016 Challenge used [14] and [15], databases of indoor and outdoor sounds and events, both described in [16].

Iranian authors Ghaderi and Kabiri use their own sound database in several publications dedicated to acoustic fault analysis of combustion engines, such as [17], [18]. Dataset consists of 4 cars with 60 recordings of both faulty and normal sounds, resulting in 480 engine recordings. Tab. 1 summarizes all of the above described databases.

2.1.1 Gunshots Databases

Specialized gunshot sound databases are much scarcer, they are mostly produced for military or civil security purposes, in which cases the access is quite difficult. First encountered database consists of approximately 800 gunshots and other sounds evoking danger (e.g. explosions, car crashes …) [19], but it is available only to INDECT project partners, project dealing with intelligent security described in chapter 3. Another database consists solely of gunshots and mechanical sounds produced by weapons [20]. It was compiled for movie making purposes, however apart from postprocessed version, there is also raw version that does not incorporate any modifications and so is viable for our purposes. The dataset consists of around 1500 recordings of gunshots and additional mechanical sounds, recorded in WAV format with high quality (two audio channels, sampling frequency of 192 kHz and 24-bit quantization), as a part of unifying our dataset, this was later downsampled as described in the following section. Other works incorporating gunshots usually have few isolated examples, usually sourced from open sources such as [5] or [6] without compiling dedicated datasets.

Table 1 Overview of sound datasets

Title	Date	Types	Size
A Dataset and Taxonomy for Urban Sound Research [4]	2014	Air conditioner, car horn, dog bark, drilling, engine, gunshots, playing children, jackhammer, siren, street music	27 hours audio, 18.5 hours annotated sounds, 1302 recordings
Freesound [5]	Since 2005	Recorded and synthetic, Various classes	230 000+ recordings
Soundbible [6]	Since 2006	Recorded and synthetic, Various classes	2000+ recordings
The Joint Database of Audio Events and Backgrounds for Monitoring of Urban Areas [19]	2011	Speech events, non-speech events (from birds to gunshots), ambient noises	~ 800 recordings
Netcarity Multimodal Data Collection [7]	2009	Daily activities (eating, ironing, watching TV…)	23.5 hours, 200 examples/activity
A hardware-software framework for high-reliability people fall detection [8]	2008	Falls, door slams, background noises …	~ 450 events
Chime-home: A dataset for sound source recognition in a domestic environment [9]	2015	Speech, percussive sounds, broadband noise, video games/ TV, background noise	6.8 hours
The Million Song Dataset [10]	2011	Music – features only	1 million contemporary popular music tracks
British Library Sounds [11]	-	Various (speech, ambient noises, music …)	80 000
Airborne Sound [13]	-	Ambience, crowds	50 recordings, 113 minutes (free recordings)
The free firearm sound effects library [20]	-	Gunshots, gun mechanic sounds	~ 1500 gunshots, ~ 1000 gun mechanics sounds
TUT Acoustic scenes 2016, Development dataset [14]	2016	Indoor and outdoor ambient noises (Cafe, library, park, beach …)	~ 10 hours
TUT Sound events 2016, Development dataset [15]	2016	Indoor events (dishes, glass, object impact …), outdoor events (birds, cars, banging …)	108 minutes, 54 recordings
Automobile Independent Fault Detection based on Acoustic Emission Using FFT [17]	2011	Healthy and faulty engine sounds	480 recordings (4 cars, 2 states, 60 recordings each)

2.1.2 Our Dataset

Our dataset consists of selected audio data from previously mentioned datasets, as well as some recordings made by us. The database is divided in two parts. Firstly ambient noise, which contain outdoor noises, such as construction site, crowded place or rain and indoor noises, for example air conditioning. Second part consists of specific events, these include gunshots, breaking glass, cracking wood, barking dogs etc. Tab. 2 summarizes sounds in our dataset. Some of the sounds are too subtle or out of context to be used as context sounds for our purposes (such as dropping keys or ringing phone), their presence is for possible future use. However they can still be used in place of general impulsive sounds in absence of other, more suitable sounds.

All selected sounds were in lossless format (such as .wav or .flac) and using various sampling frequencies. For the sake of unity, all sounds were subsampled to 44.1 kHz with 16-bit quantization, multiple channels were averaged to mono signal and the resulting audio was saved in wav format.

Table 2 Dataset compiled from other datasets and our recordings

Type	Noise / Event	Quantity
Crowds indoor	Noise	85 minutes
Outdoor noises (mostly crowds)	Noise	37 minutes
Air conditioning	Noise	72 minutes
Drilling	Noise	90 minutes
Car engine	Noise	65 minutes
Playground	Noise	232 minutes
Jackhammer	Noise	90 minutes
Siren	Noise	75 minutes
Street music	Noise	315 minutes
Car horn	Event	108 recordings
Coughing, throat cleaning	Event	40 recordings
Wood cracking	Event	17 recordings
Barking dog	Event	258 recordings
Door slams	Event	20 recordings
Drawers	Event	20 recordings
Keys dropping	Event	20 recordings
Elephant trumpeting	Event	13 recordings
Gunshots	Event	1532 gunshots
Keyboard	Event	20 recordings
Knocking	Event	20 recordings
Laughter	Event	20 recordings
Phone ringing	Event	20 recordings
Page turning	Event	20 recordings

Our database is divided into three sections as follows: 1) gunshots from hunting weapons, 2) other acoustic events potentially occurring in the elephants' environment, and 3) mixtures of gunshots with other acoustic events. The single category sounds in the first two sections are intended for training algorithms while the mixed sounds in the third section are more suitable for testing. Gunshot sounds were recorded at different angles and distances from the microphone. In total, the largest subset of sounds of the same type is represented by 374 gunshots from the assault rifle AK-47. We have investigated the similarity of individual gunshots within various classes comparing gunshots from the same weapon, the same category (caliber) of weapons and all gunshots together, both in the time domain and spectral domain. In all cases, individual gunshots were extracted from the recordings using a rectangular window with a length of 30 ms and then, each extracted gunshot signal was normalized so that maximum absolute amplitude was equal to one, in order to eliminate effects of different intensity of sounds. Subsequently, all gunshots were time synchronized by setting the maximum point to be at a specific time location, and finally limited to a length of 1024 samples (approx. 23 ms). These synchronized gunshot waveforms were stored together in a time-amplitude distribution matrix. In statistical processing the mean μ(t) and standard deviation σ(t) were estimated sample by sample throughout the whole gunshot duration. Fig. 4 shows a graphical interpretation of the distribution matrix, displayed as a grey scale image together with the statistical parameters obtained for a subset of 308 gunshots within the class of AK-47s. As a part of preprocessing, gunshots originating very far away, having low original maximum amplitude (before normalization) were not considered – they would be ignored by the sound detector in real signal processing. The darker shade in Fig. 4 means that the waveforms are more concentrated around the average waveform. Fig. 5 shows two examples of gunshots from AK-47

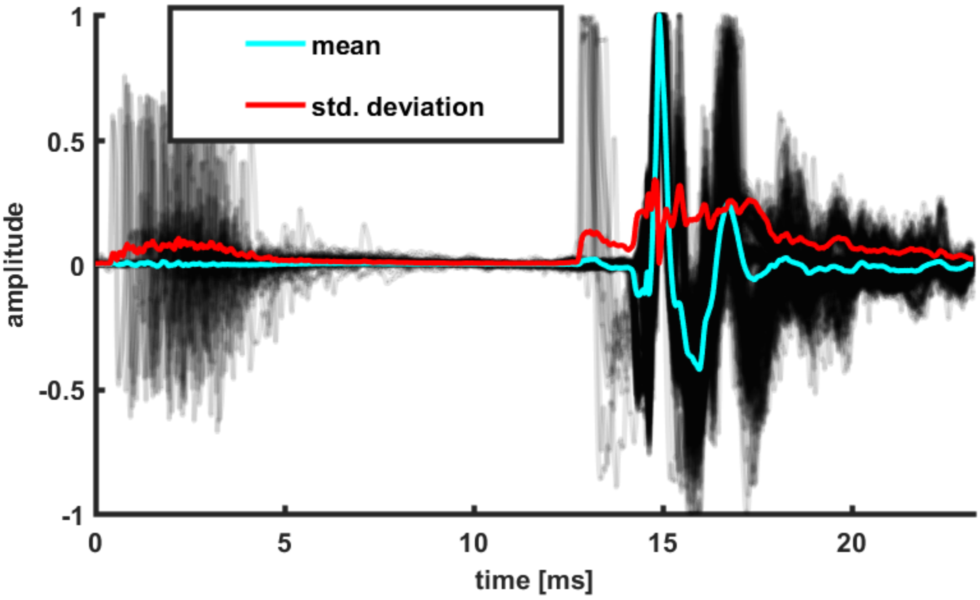

Fig. 4 Gunshot waveforms stacked on top of each other

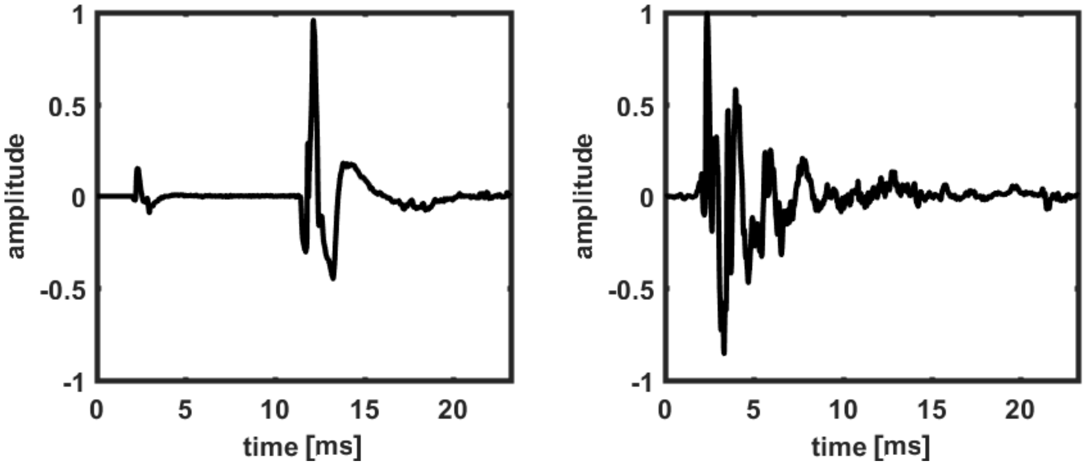

Fig. 5 AK-47 waveform with (left) and without (right) shock wave

Generally, the muzzle blast has quite a distinctive shape which is known as the N-wave [8]. In order to characterize this shape, we have statistically observed two significant features – the two zero-crossing points close to the main positive peak. Fig. 6 shows the distribution of the zero-crossings immediately before and after the peak for AK-47 shots in relation to the position of the maximum point given as the zero point on the horizontal axis.

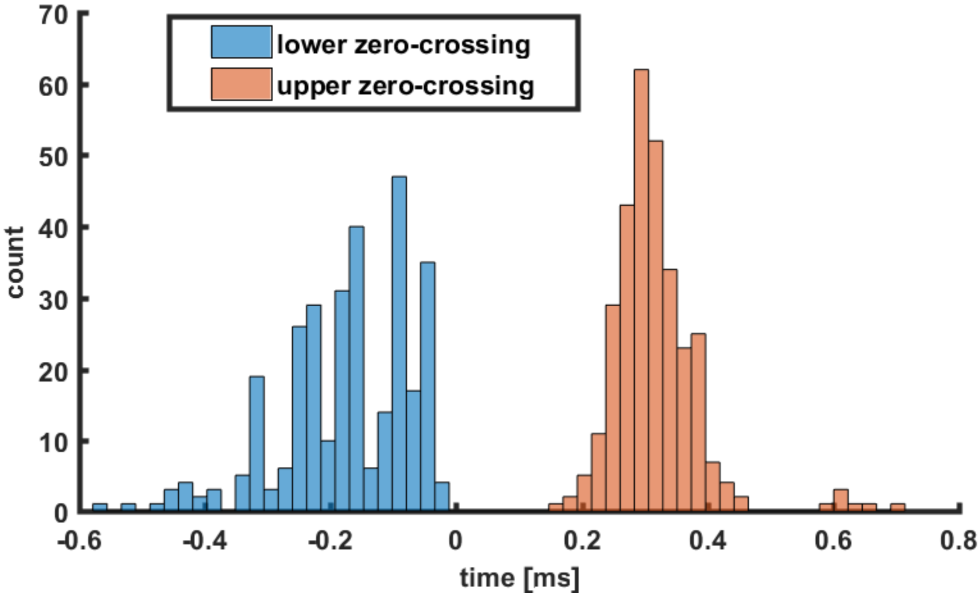

Fig. 6 Distribution of zero-crossings relative to peak

To evaluate overall similarity of waveforms between gunshot classes, standard deviation curves σ(t) corresponding to the muzzle blast were compared. Tab. 3 shows examples of cumulative standard deviations calculated for different gunshot sets both including and excluding gunshots originating far away (i.e. ca. 16% of all gunshots). Note that the far-away gunshots having overall low amplitude were also normalized into the range –1 to +1. Thus the standard deviations depend exclusively on the shape of gunshot waveforms. A lower value of standard deviation means that the individual waveforms are more similar to each other.

Table 3 Cumulative standard deviations as similarity criterion

Weapon	**Without far gunshots**	**With far gunshots**
Assault rifle AK-47	68.7	67.0
All assault rifles	79.1	79.4
All weapons	86.3	95.0

These results indicate, as expected, that with increasing variability of weapons included, gunshots are less and less similar, the method above aimed to quantify those differences.

In order to investigate spectral characteristics of gunshots, the conventional Fourier transform was applied. Fig. 7 shows the magnitude spectra of gunshots from an AK-47 displayed in the same manner as the waveforms in Fig. 4. In this case, each spectrum was calculated only from the muzzle blast without considering acoustic shock. In general, gunshot energy is significant at low frequencies. For instance, the average spectrum (blue line) covers 90% of energy in the range from 1 to 2024 Hz. In all cases there is a very obvious spectral peak located at 561 Hz.

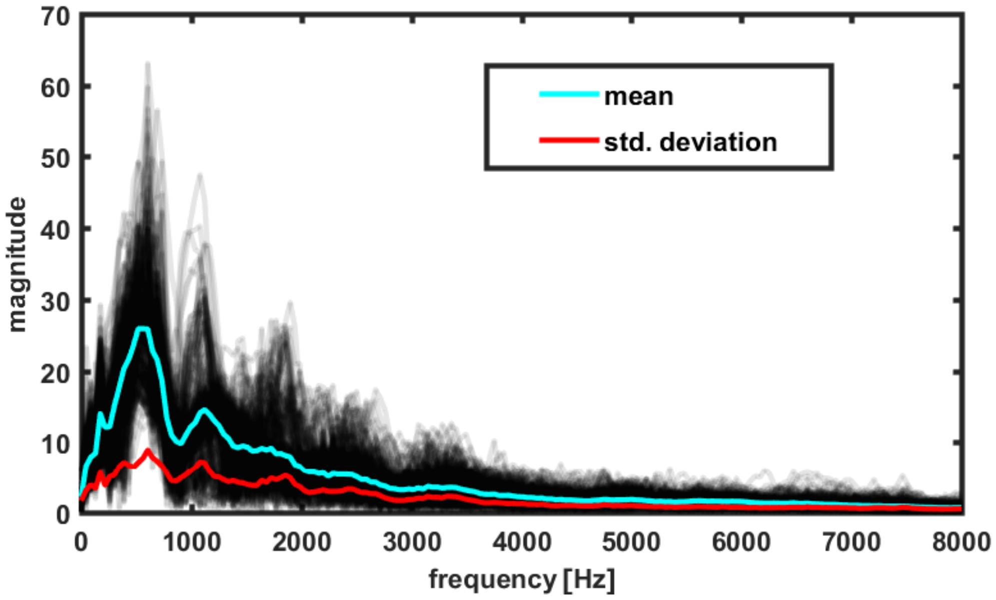

Fig. 7 Spectra of gunshots stacked on top of each other

Among all tested non-gunshot sounds, the ones with more impulsive character (such as cracking branches or thunderstorm) are expected to be more similar to gunshot sounds, as opposed to those with stationary character (e.g. rain, idling engine, etc.). However these sounds also sometimes contain unpredictable events which can change otherwise stationary characteristics. Figures 8-11 illustrate some similarities between a branch crack and a gunshot from an AK-47.

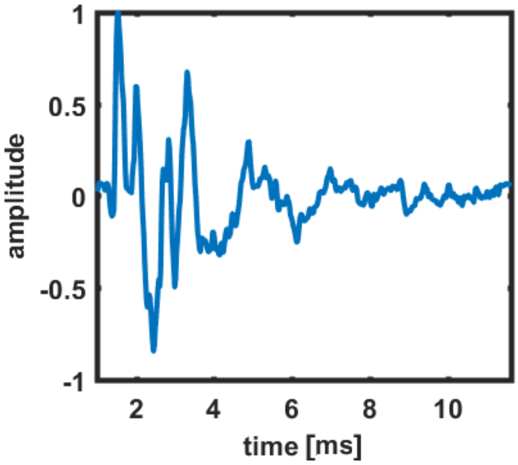

Fig. 8 Gunshot waveform

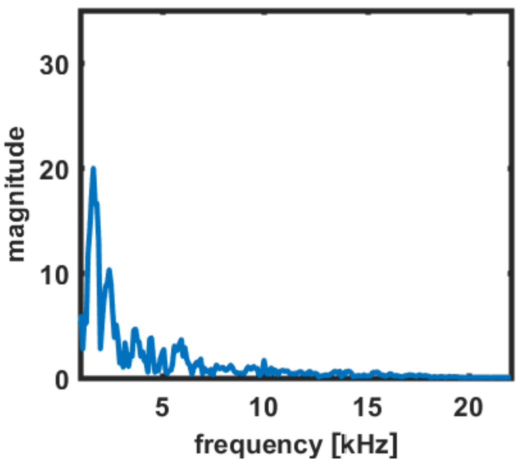

Fig. 9 Gunshot spectrum

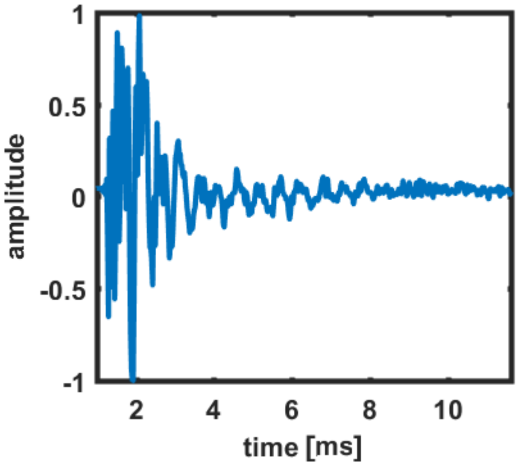

Fig. 10 Waveform of branch crack

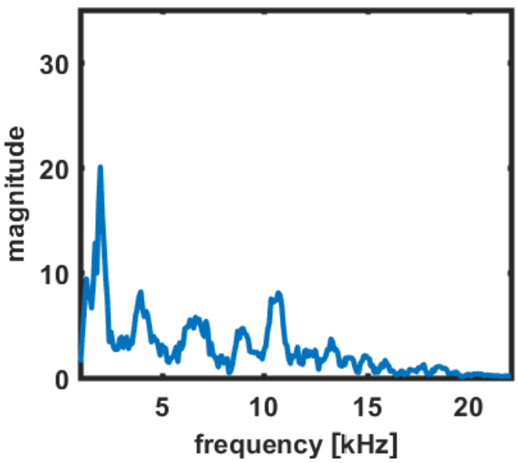

Fig. 11 Spectrum of branch crack

It can be seen, that both gunshot and cracking branch have visually similar waveform and also spectrum. This example in particular shows gunshot spectrum with more dence peaks (which is consistent with Fig. 7 where multiple gunshots are shown), while cracking branch has peaks spread more far apart.

3 INFORMATION SOURCES

This chapter presents information sources relevant to topics of sound event detection and gunshot detection. It briefly lists important conferences and journals, furthermore it mentions national and international projects under which gunshot detection or sound event detection systems were developed. Final section provides overview of important publications beginning in 2000 along with authors and institutions dealing with sound event detection or acoustic scene classification.

3.1 Conferences and Journals

Papers dedicated to gunshot and audio recognition are usually presented on general audio and signal processing conferences or, secondly, on conferences dedicated to machine learning and pattern recognition. These conferences present topics such as audio scene classification, sound event detection, source separation, audio event localization but also speech recognition. Recently, also topics of sound generation are more and more frequent with the advent of generative adversarial neural networks (GANs) and Wavenet [21] in particular. Below is a short list of conferences at which these papers are published:

- IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)
- IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)
- Interspeech
- Detection and Classification of Acoustic Scenes and Events (DCASE)
- International Workshop on Machine Learning and Music
- European Signal Processing Conference (EUSIPCO)
- International Workshop on Acoustical Signal Enhancement
- AES International Conference on Audio Forensics
- AES International Conference on Semantic Audio

Among journals, three notable were found, apart from numerous machine learning journals:

- IEEE Transactions on Audio, Speech and Language Processing
- IEEE Signal Processing Magazine
- Eurasip Journal on Advances in Signal Processing
- Eurasip Journal on Audio, Speech, and Music Processing
- IEEE-ACM transactions on audio speech and language processing

3.2 Projects

Increasing number of national and international projects are dedicated to sound events and music recognition. The aims range from allowing multimedia search by audio content, recognition of environments where recording took place, but also security reasons such as detecting dangerous situation or automated captioning in case of speech recognition.

Gunshot and dangerous sound recognition was part of comprehensive European project INDECT, which was dedicated to "development of solutions to and tools for automatic threat detection". Project duration was since January 2009 until June 2014. The project was solved by a number of european universities, part of this project was also solved in Košice University of Technology, Slovakia . Dangerous sounds dataset was recorded there and several important publications dealing with dangerous sound identification (gunshots and breaking glass) or specific weapon identification based on audio recording originated from the department.

Netcarity project, which ran for four years since 2007, was dedicated to increasing quality of life for elderly people. One of the aims was to develop fall detection algorithm, this task was also connected with creating multiple sound effects datasets.

Ongoing project (2015-2020) number 637422, funded by European Research Council, Computational Analysis of Everyday Soundscapes (EVERYSOUND). It is solved by audio research group from Tampere University of Technology and its aim is „to develop computational methods which will automatically provide high-level descriptions of environmental sounds in realistic everyday soundscapes such as street, park, home, etc. This requires developing several novel methods, including joint source separation and robust pattern classification algorithms to reliably recognize multiple overlapping sounds, and a hierarchical multilayer taxonomy to accurately categorize everyday sounds."

3.3 National Projects

Sound and music processing and recognition is topic of many national grants and projects. Below is a list of few examples from USA and United Kingdom, where national projects on this topic are abundant.

- (USA) Sounds of New York City (SONYC) project
- (USA) Data-Driven Music Audio Understanding
- (USA) An Integrated Framework for Multimodal Music Search and Discovery
- (UK) Structured machine listening for soundscapes with multiple birds
- (UK) Making Sense of Sounds
- (UK) Audio Data Exploration: New Insights and Value
- (UK) Machine Listening using Sparse Representations

Sounds of New York City (SONYC) project is still ongoing project, dedicated to monitoring and combating noise and also to "accurate description of acoustic environments in terms of its composing sources". This project collected dataset [4] and established taxonomy of urban sounds. The team also develops sound classification algorithms and acoustic monitoring devices. [22]

Data-Driven Music Audio Understanding is concerned with “extracting information from music audio and discovering deeper patterns and structure within it”. Project was solved between 2007 and 2011 by the Laboratory for the Recognition and Organization of Speech and Audio (LabROSA) at Columbia University (estimated) but last publications are from 2008. [23]

An Integrated Framework for Multimodal Music Search and Discovery is also dedicated to music, namely annotation of non-speech audio with descriptive keywords and development of methods improving music discovery and search. Project is running from 2011 until 2017 (estimated). [24]

Structured machine listening for soundscapes with multiple birds develops methods for recognition of birds in multisource recordings and interactions between them. Project duration is 2014-2019. [25]

Making Sense of Sounds, project “on how to convert these recordings into understandable and actionable information: specifically how to allow people to search, browse and interact with sounds.” It is focused on on general description of sound in order to improve ability to search in comprehensive sound databases. Project duration is 2016-2019. [26]

Audio Data Exploration: New Insights and Value, this project dealt with automatic environmental sound recognition. It was solved by Audio Analytics Ltd. In cooperation with Queen Mary University. Project ran from 2014 until 2015. [27]

Machine Listening using Sparse Representations aimed to understand ability to hear and recognize sounds and introduce new methods for machine listening of general audio scenes. Project duration was from 2008 until 2014. [28]

3.4 Relevant Publications

Acoustic signals recognition and classification is well established discipline with many publications. However most publications are dedicated to speech recognition. This section will mention some papers dedicated to event detection and recognition which includes feature selection, algorithm design and comparison and also acoustic characterization (in gunshots). This summary will deal with works published in 2000 or later.

A. Dufaux published influential paper dedicated to sound recognition in noisy environments [29], followed by dissertation which compares different features and classifiers under different noise levels [30].

Several papers were published from Polytechnic University of Catalonia, first author A. Temko published papers dealing with event classification, focused on human produced, non-speech events in office settings (such as cough, applause, door slams etc.). Papers published range from 2005 [31] until 2009 [32]. Next author from the same institution is T. Butko, who continues in publishing papers focused on office settings, but for the sake of improved accuracy includes also video features. As a first author, his first publications date from 2008 [33] and most recent in 2011 [34], in 2013 he co-authored paper on footsteps gait analysis [35]. Both authors publish together with C. Nadeu, who deals mostly with speech signals.

One influential paper on event detection [36] was published by an author who usually deals with speech processing. Authors A. Pikrakis and T. Giannakopoulos from University of Athens authored (together or separately) several papers on violent content detection, including gunshots ranging from 2006 [37] to 2010 [38]. An important author in gunshot acoustics is R. Maher from Montana State University. His works date from mid-1980's and deal mostly with music. Works dealing with gunshot acoustics range from 2006 [39] to 2016, later he was co-authoring with T. Routh [40]. In 2007, two works on gunshot detection were published from Polytechnic University of Milan, one deals also with localization [41], the other with noisy environments [42]. In 2008 and 2009 a collective of authors from Portuguese INSEC-ID published several papers on non-speech audio event detection, among others [43] [44] [45]. Since 2008 University of Illinois Urbana-Champaign authors, most notably X. Zhuang published papers on event detection. Papers deal with HMM-based detection [46], feature analysis [47], speech and non-speech audio events detection [48] and event detection based on visual saliency in spectrograms [49]. In the framework of EU project INDECT, team from Technical University of Košice published several papers dealing with sound event detection, focused on events indicating threat, mostly including gunshots. First publication from 2010 [50] was continued by others, most notably dealing with features [51][52]. Works on this topic continue, with latest at the time of writing being [53]. First notable publication from Carnegie Mellon University on acoustic event classification was from 2005 [54], followed only after 2011 by authors A. Kumar [55][56] and S. Chaudhuri [57]. A big number of important papers both on sound event detection and speech processing along with other tasks, such as audio source separation or localization come from audio research group on Tampere University of Technology, led by T. Virtanen. Among the most recent papers dealing with event detection are [58][59] or [60].

4 APPLICABILITY OF GUNSHOT RECOGNITION

Gunshot detection and recognition has wide applicability, apart from obvious military purposes. Gunshot sounds are connected with threat, so gunshot detection in populated areas can be connected with law enforcement such as in INDECT project [50], be it in general area detection to alert police or connected with localization (angle and distance) for forensic purposes. Extended gunshot recognition, such as gun type or ammunition used can also be used for forensic purposes [61]. Apart from open-air detection, detection from recordings (such as telephone calls) was considered [39][41].

Illegal use of weapons is problem not only in cities but also in the wilderness. Some papers deal with this problem in order to protect wild animals from poachers. It can be in the form of microphone arrays [62] or special modules used together with tracking collars [63]. These devices differ slightly in their requirements from those used in cities in their need for lower power consumption (due to limited ability to change batteries or inability to use power grid), which comes from lower computational demands and low false alarm rate (due to increased difficulty of checking frequent false alarms).

Apart from these uses, acoustic events detection can be used in general audio discovery in media [64] or content annotation and classification in action movies or recordings [37].

Fig. 12 presents wide applicability diagram for real-time detection and recognition. Fig. 13 shows possible uses for recordings, military purposes are not included.

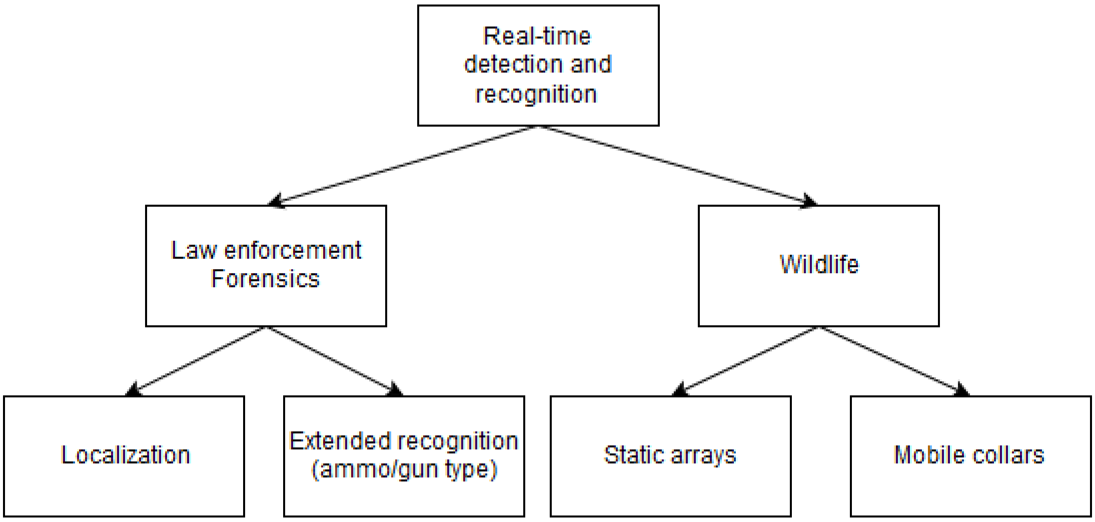
Real-time detection and recognition
Law enforcement Forensics
Wildlife
Localization
Extended recognition (ammo/gun type)
Static arrays
Mobile collars

Fig. 12 Real-time applicability

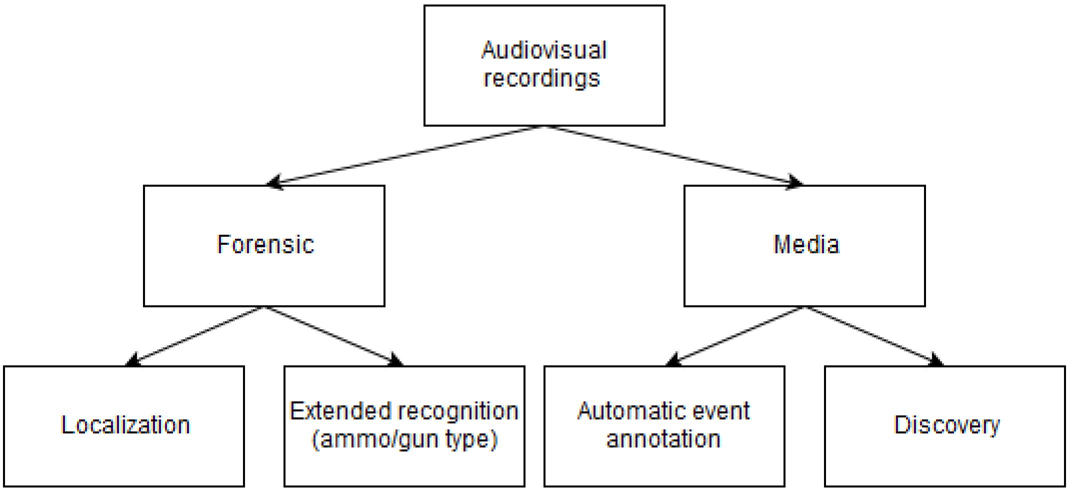
Audiovisual recordings
Forensic
Media
Localization
Extended recognition (ammo/gun type)
Automatic event annotation
Discovery

Fig. 13 Applicability in recordings

4.1 State-of-the-Art in Gunshot Recognition

The comparison of results achieved in audio event detection, and specifically gunshot recognition is quite complicated due to isolated nature of work in various institutions. First problem is the absence of big, accessible and widely used audio datasets. Various authors and research groups record their own datasets or assemble them from crowdsourced data or multiple other datasets. Even though they usually continue using this dataset and compare the results with their new proposed methods, other authors use different datasets and so the results are not consistent. Another problem is, there are multiple metrics, such as overall accuracy, recall and precision, or specificity and sensitivity, this makes direct comparison of results even more complicated and sometimes outright impossible. One big exception to both problems is DCASE challenge, which presents dataset for multiple tasks (such as acoustic scene classification or sound events detection) and calls upon authors to come up with algorithms that perform best on this data using single metric. This chapter will try to list some notable results regardless of just presented problems.

The first work compares several features and measures (correlation against template, 8th order LPC coefficients, 13 MFCC coefficients and impulsivity measure), the features are extracted with 25ms windows over the whole recording, however correlation provides just one number per recording [65]. The features that have temporal information (MFCC, LPC and impulsivity – since they are extracted over whole recording) are fed to HMM with 20 observable states and 8 hidden states. The probability distributions were modelled by a mixture of three Gaussians. The correlation is used with simple threshold. This paper compares 4 sound classes (gunshot, balloon, speech and clapping) with 22 instances each, under different Sound to Noise ratios (SNR), from clean signal to 20 dB SNR. In conditions without noise, each measure correctly detected all gunshots, though correlation and impulsivity made errors in classifying non-gunshot sounds, sometimes causing false positives. Correlation was chosen by the authors as the best measure under noisy conditions, at 20 dB SNR gunshot detection was 91% successful with 23% false alarms.

Paper [37] focuses on violent content classification in movies using audio (presence of gunshots, screams etc.). It is important to note that this approach might not be the best in our case, since audio effects in movies are heavily edited and so might be very different from real-life gunshots, but the paper can still provide some interesting ideas. The method works with individual segments (scenes) divided into W time frames (each 400 ms). Scenes are subsequently classified based on feature statistics (6 features with 8 derived statistics) extracted from these W frames. Features used include energy entropy, amplitude, short-time energy, Zero-Crossing Rate (ZCR), spectral flux and spectral roll-off. As an algorithm, SVM with 4 different kernel functions were tested, along with varying C parameter, which expresses trade-off between training error and margin. The polynomial kernel achieved the best performance, achieved results are 90.5% correct detection and precision of 82.4% on a dataset of total duration 20 minutes with 50% used for training and 50% used for testing. Half of the dataset consisted of positive class (violent content, such as gunshots and fights) and half of negative class (non-violent content such as speech, music or fireworks).

Gerosa and Valensize [42] achieved successful detection rate of 92% with 10% false alarms. Their work aimed to distinguish scream and gunshots from ambient noise. Test database consisted of recordings from movies and internet, some live recordings of shouting people, ambient noise consisted of recording public square in Milan. Part of experiment consisted also of training and testing under different SNR (0 dB to 20 dB with 5 dB steps). System uses two parallel Gaussian Mixture Models (GMM) classifiers (one for gunshots and one for screams) the number of components for each GMM is automatically chosen using Figueiredo-Jain algorithm. Each classifier uses subset of features chosen from a set of 47 (such as MFCC, PLP coefficients, ZCR or spectral features). Feature selection used in this work is a mix between filter methods (that evaluate features without considering the classifier) and wrapper methods (that use classifier performance as a metric on how useful the feature is).

Another classifier uses two-stage method, with first stage serving as event detection using normalized signal energy. The second, gunshot recognition stage, is using Gaussian radial basis function kernel for SVM [66] and achieved 97% True positive rate and 0 false negatives in a dataset of 332 gunshots and 102 outsider signals (claps, door slams, talking and ticks). Using only 8th order LPC coefficients and correlation. MFCC and linear kernel for SVM were also compared, achieving inferior results.

Paper [67] focused on audio event detection (gunshots (463) and breaking glass (150)) in background noise (53 minutes of traffic). This work compares different setups of HMM classifier and subsets of features (based on first 13 MFCC and MPEG-7 descriptors). Feature selection was used considering mutual information of class labels and features and also mutual information between individual features. Best results were achieved for 3-state HMM with 38 features, achieving 100% recall and 98,33% precision. In this work, 90 % of data was used during traning and validation and 10 % during testing stages.

5 COMPARISON OF FREQUENTLY USED FEATURES

This chapter will present comparison of various features under different conditions, including variation of frame length, event position within frame and various noise conditions. Observed features include mel-frequency cepstral coefficients (MFCC), linear prediction coefficients (LPC) and linear prediction cepstral coefficients (LPCC). Performance was evaluated using Matlab Neural Net pattern recognition tool, using neural network with one hidden layer with 10 neurons. Data was divided into training, validation and testing sets in default proportion 70%, 15% and 15% respectively, using random permutations for data division during each training round.

To represent results, we will be using recall (also known as true positive rate), precision (PPV – positive predictive value) and F-score, calculated as shown in equations (5), (6) and (7) respectively:

$$recall = \frac{TP}{P} = \frac{TP}{TP + FN}, \tag{5}$$

$$precision = \frac{TP}{TP + FP}, \tag{6}$$

$$F - score = 2 \cdot \frac{precision \cdot recall}{precision + recall}, \tag{7}$$

where TP (True Positives) is number of gunshots classified as gunshots, P is number of all gunshots (Positives), FN is number of gunshots misclassified (False Negatives) and FP (False Positives) is number of non-gunshots classified as gunshots.

To establish baseline performance, Tab. 4 below shows performance (precision was used to indicate performance) of features of different orders using frame length of 1024 samples (approx. 23 ms at 44.1 kHz) which is frequently used frame length in similar applications. This table can be used as a starting point to compare results from subsequent sections [68].

Table 4 Precision (6) of various features with frame length 23 ms [68]

Number of coefficients	**Feature set**		
	LPC	LPCC	MFCC
8	83.3 %	84.6 %	84.4 %
12	87.8 %	86.3 %	86.9 %
16	88.5 %	87.4 %	83.4 %
20	89.3 %	88.4 %	83.2 %
Average performance	87.2 %	86.9 %	84.5 %

5.1 Effects of Frame Length and Position on Feature Variability

This section compares effects of different frame lengths on gunshot recognition and explores the effect of frame length and position of audio event in frame on variability of features. The aim of this approach is to reveal relevance of given feature to gunshot class. Preliminary experiments were conducted using only small number of gunshots, after obtaining results, we proceeded to include the whole dataset. In order to investigate influence of frame length, gunshot recordings were segmented into frames of lengths 3 ms, 5 ms, 8 ms and 11 ms, as shown in Fig. 14. As to the event position, gunshots were segmented into frames of length 3 ms with 50% overlap as shown in Fig. 15. Variability/stability observation consisted essentially of comparing values of coefficients under changing conditions. Illustrative results are shown in Fig. 16 and Fig. 17, these represent LPC coefficients (which were the most stable from the three sets) of order 20, due to space constraints, legends were removed, but individual lines represent individual coefficients.

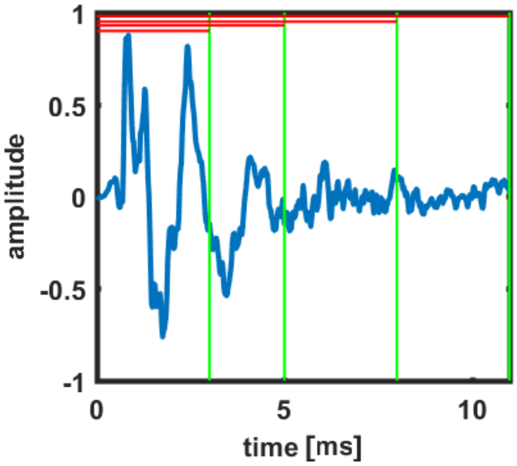

Fig. 14 Increasing frame size

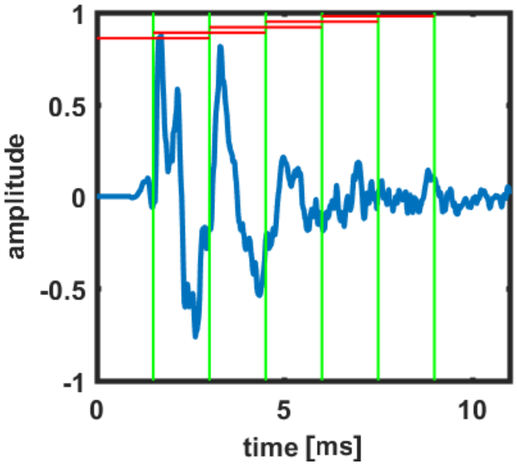

Fig. 15 Shifting event position within frame

Fig. 16 shows, with the exception of 5 ms frame, relatively stable coefficients (on small dataset), however recognition results were visibly impacted by shortening the frame (numeric results of recognition will be presented later). Fig. 17 again shows relatively stable coefficients for first three shifts, i.e. while muzzle blast of gunshot is still at least partially present in the frame, subsequent shift to a low-energy zone is reflected on the value of coefficients. The following steps deal with frame length, due to less difficult pre-processing.

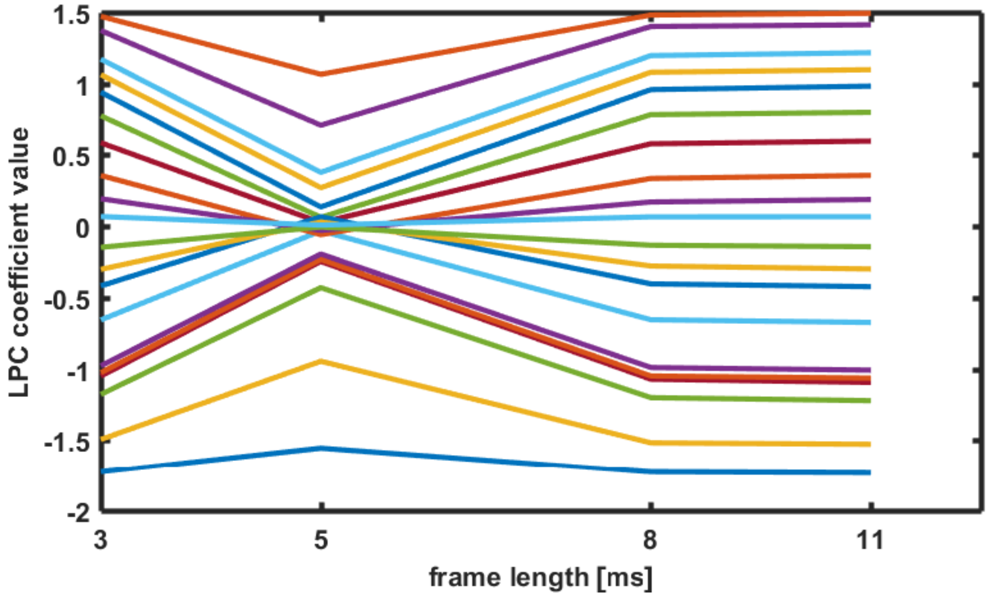

Fig. 16 LPC coefficients (order 20) - increasing frame size

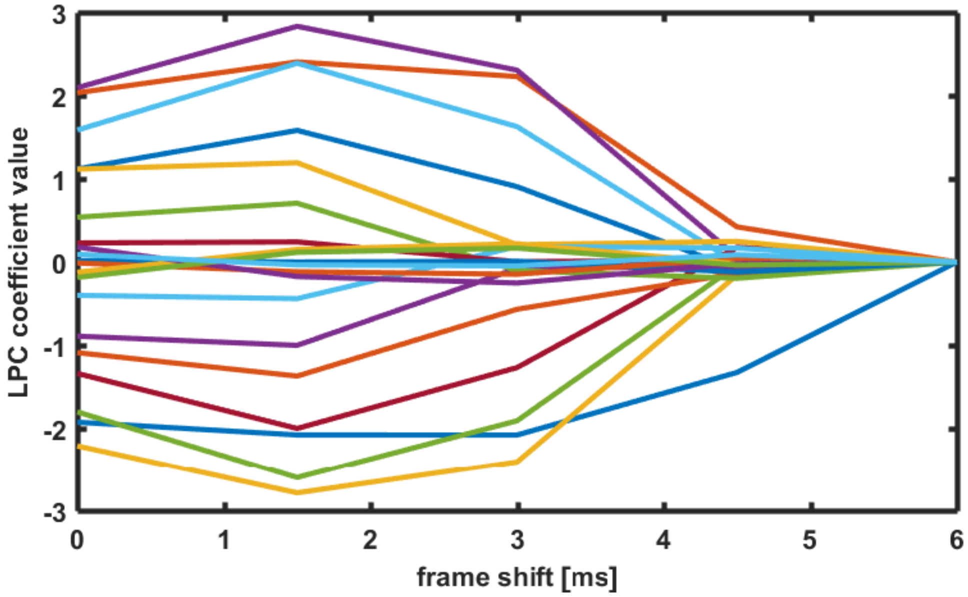

Fig. 17 LPC coefficients (order 20) - shifting frame position

In the next step, feature performance was estimated for all frame lengths and for different orders (8 to 30). MFCC was tested in two ways, MFCC20 column denotes features extracted with 20 banks and changing number of coefficients, MFCC column has number of filter banks and coefficients extracted equal.

Figures 18-20 show progressively decreasing recall for different features with decreasing frame lengths. Results for frame length of 11 ms are also summarized in Tab. 5 and Tab. 6. Results achieved for frame length 11 ms for LPC and LPCC achieved results very similar to the ones achieved with frame length 23 ms (results compared with Tab. 4

[68]) and better than other frame lengths. Thus, we will explore viability of frame length of 11 ms in the following tests, unless otherwise noted. Observation also shows, that there is no substantial improvement beyond order 12 for LPC or LPCC.

Initial improvement in performance was observed for MFCC20 with increasing number of coefficients, but subsequently random behaviour. MFCC experiences steady increase in performance with increasing order (and number of coefficients). Comparison of both can be seen in Tables 5 and 6, due to more monotonous behaviour of MFCC (as opposed to MFCC20), this approach will be used, unless otherwise mentioned.

Table 5 Recall (5) for frame length 11 ms

Order	**Feature set**			
	LPC	LPCC	MFCC20	MFCC
8	84.2	82.1	79.9	77.6
10	84.6	84.1	81.6	82.4
12	87.0	84.3	82.8	80.8
14	86.0	82.9	83.8	85.9
16	86.3	85.0	85.4	86.5
18	86.9	83.9	81.1	82.6
20	86.4	83.4	84.5	85.1
22	86.7	83.9	78.2	86.2
24	84.8	83.1	82.9	85.2
26	85.3	84.4	81.3	86.0
28	85.7	84.6	84.3	84.5
30	85.9	83.9	85.8	84.3

Table 6 Precision (6) for frame length 11 ms

Order	**Feature set**			
	LPC	LPCC	MFCC20	MFCC
8	81.7	84.0	74.3	73.8
10	85.1	85.5	77.5	76.8
12	90.3	87.3	82.0	79.2
14	89.2	88.8	80.1	77.2
16	89.4	89.3	80.9	80.4
18	90.0	89.0	79.9	81.9
20	90.3	87.2	78.9	81.8
22	89.9	89.3	81.3	82.2
24	89.8	87.8	81.3	83.0
26	89.3	88.0	81.9	82.0
28	90.9	88.1	81.7	84.8
30	89.1	88.9	77.8	84.5

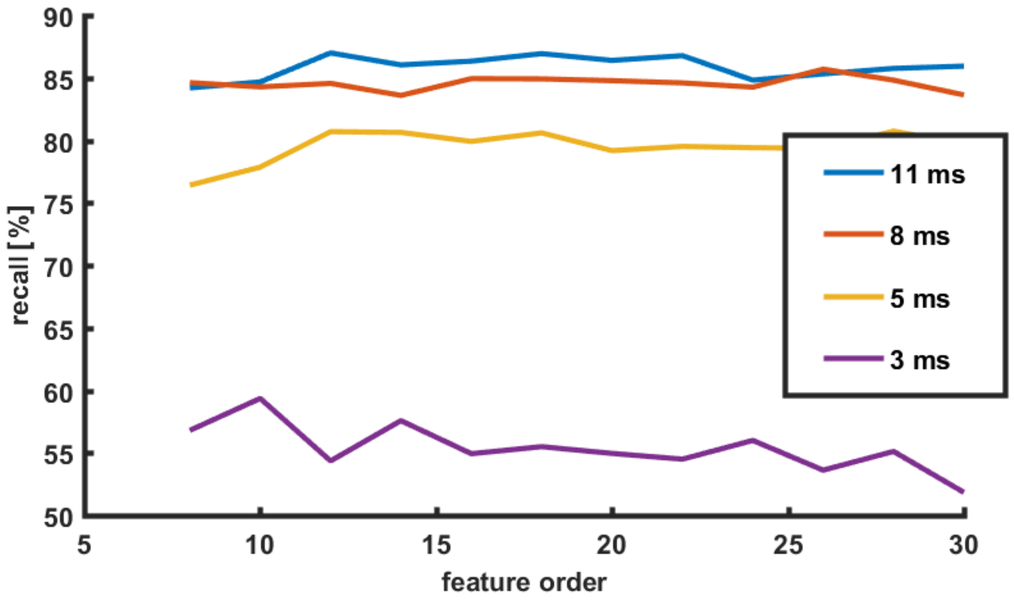

Fig. 18 Recall (5) of LPC coefficients of various orders for different frame size

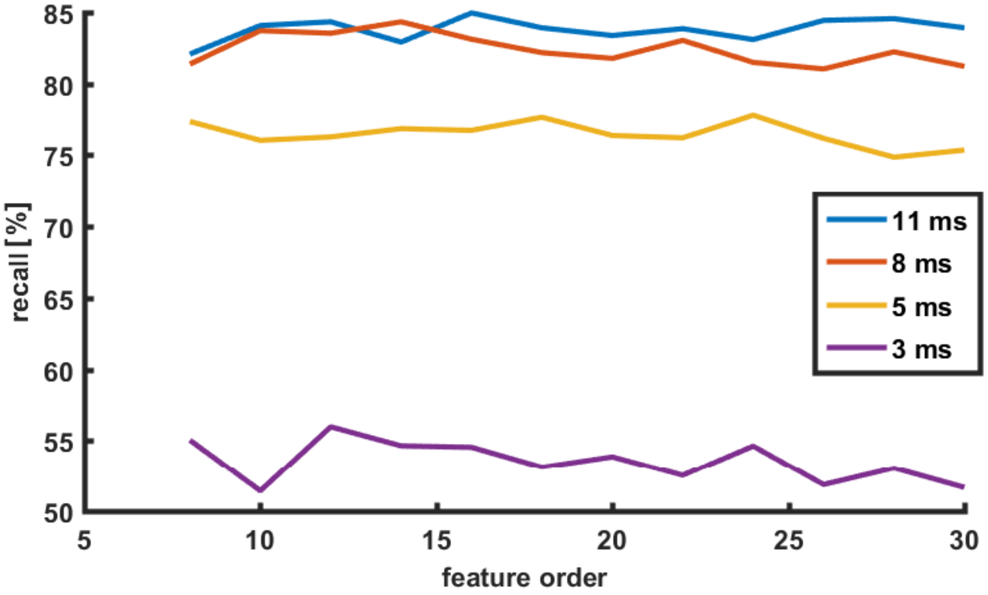

Fig. 19 Recall (5) of LPCC coefficients of various orders for different frame size

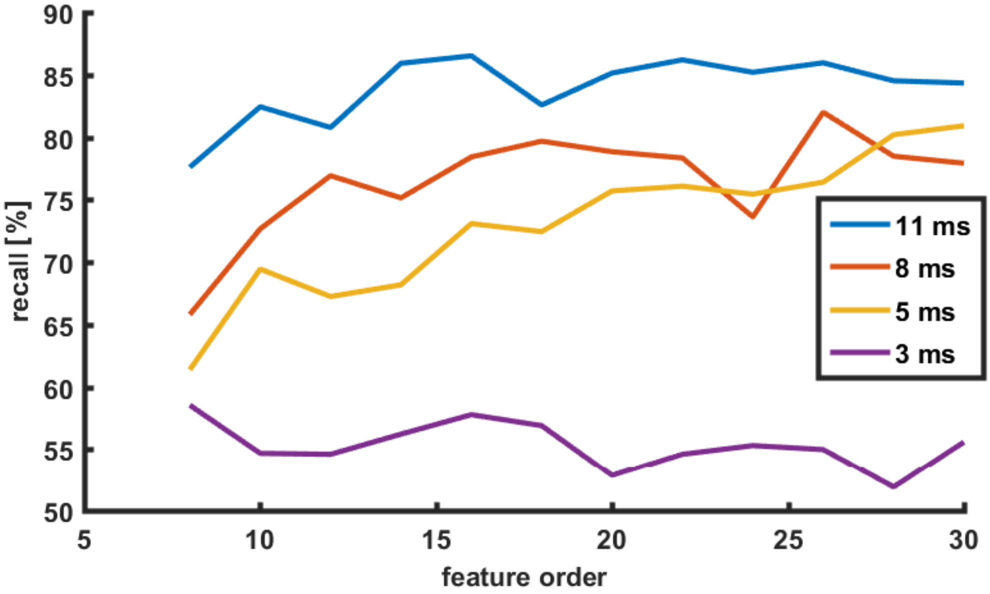

Fig. 20 Recall (5) of MFCC coefficients of various orders for different frame size

Figures above conclusively show, that 3 ms frame is insufficient. Recall difference between 11 ms and 8 ms frames is marginal, what suffers during this reduction is precision. With reducing frame size from 8 ms to 5 ms precision remains roughly the same, while recall diminishes. As noted above, these are the reasons why we choose 11 ms frame size for subsequent experiments.

In the following part, feature variability with respect to frame length was compared. In this step, only gunshot sounds were used (1532 from various weapons, distances and angles). Features were extracted from all sounds using various frame lengths, they were then compared and the most invariant was chosen.

Two methods were used, first, absolute differences between individual features were compared. Calculation of difference is shown in equation (8):

$$\overline{\Delta_m} = \frac{\sum_{k=1}^{K} \sum_{p=1}^{P-1} \left(a_{m,k,p+1} - a_{m,k,p}\right)}{K(P-1)}, \tag{8}$$

where m is series index of coefficients, $1 \leq m \leq 30$, k is gunshot index, p is index of frame position, and $a_{m,\,k,\,p}$ are corresponding coefficients. Variability of best 3 coefficients (i.e. coefficients with the lowest variability calculated with (8)) from each order were summed and compared with other orders, Tab. 7 below shows results. When changing number of best coefficients during evaluation (e.g. considering 5 coefficients instead of 3), best feature order may vary.

Table 7 Best feature orders and coefficients using absolute values

Feature	Best order	Best 3 coefficients
LPC	30	30
		29
		28
LPCC	8	3
		1
		5
MFCC	26	24
		23
		21

Apart from LPCC, according to this criterion, best coefficients were those with higher coefficient number (and consequently also feature order), this is probably due to much lower value range these coefficients acquire.

Subsequently, similar test was used, only with relative differences of features, calculated as shown in equation 9, dividing difference by average value of the two coefficients. This might offset different value ranges accross various coefficient indices and thus offer more unbiased view.

$$\overline{\Delta_m} = \frac{\sum_{k=1}^{K} \sum_{p=1}^{P-1} \frac{\left(a_{m,k,p+1} - a_{m,k,p}\right)}{\left(a_{m,k,p+1} + a_{m,k,p}\right)}}{K(P-1)}, \tag{9}$$

Tab. 8 shows achieved results of rating by relative feature importance. The table presents best feature order in 3 different feature categories and 3 best feature indices for each feature and feature order combination. In contrast with Tab. 7, we can see that most important coefficients (with one exception in MFCC) are concentrated among the first coefficients. We will see that this is in accordance with mutual information criterion and subsequent recognition performance.

Table 8 Best orders and coefficients using relative values

Feature	Best order	Best 3 coefficients
LPC	8	1
		2
		3
LPCC	10	1
		3
		2
MFCC	22	1
		2
		21

Since recognition performance is not significantly impacted beyond feature order 12, the real problem is choosing correct coefficients, instead of choosing order. To evaluate which of the two measures is better, feature performance will be tested using neural networks. Additionally, mutual information between class label and feature value will be calculated, as shown in (10). This measure reflects relevance of the feature in classification process for given classes.

$$I(x,y) = \sum_{i,j} p(x_i, y_j) \cdot \log \frac{p(x_i, y_j)}{p(x_i) \cdot p(y_j)}, \tag{10}$$

where $I(x,y)$ is mutual information, $p(x_\mathrm{i})$ if probability distribution of features, $p(y_\mathrm{j})$ is probability distribution of classes and $p(x_\mathrm{i},y_\mathrm{j})$ is joint probability. Computation was realized using Matlab function *kernelmi* (available on mathworks file exchange) with logarithm base 2, which uses kernels to estimate mutual information between continuous random variables. Due to memory constraints, this part uses 1532 gunshots and only 10000 segments of other sounds randomly picked from initial dataset. In this step, we are not dealing with mutual information between individual features, this means redundancy will affect information gain and so it will also negatively affect recognition performance, when adding features. Tab. 9 summarizes best results of mutual information tests for all feature sets. Mutual information was calculated according to (10) shown and explained above. For now, no mutual information between features was examined.

Table 9 Best orders and coefficients – Mutual information (10)

Feature	Best order	Best 3 coefficients	Mutual information [bit]
LPC	30	5	0.4789
		4	0.4741
		6	0.4665
LPCC	30	2	0.4132
		1	0.3535
		4	0.2298
MFCC	28	1	0.2883
		2	0.1378
		3	0.1274

In general, the most mutual information was concentrated in lower coefficients. In LPCC and MFCC, mutual information quickly decreased with increasing feature index, decrease in LPC was less dramatic, but still present.

Fig. 21 to 23 present performances represented by F-score of all coefficient sets chosen by different methods. X-axis represents number of coefficients, Y-axis is F-score. Feature order depends on which order was chosen as best by each method. The following methods were tested: absolute and relative stability, mutual information between features and class labels (in legend labeled as "MI") and, for reference, simple increase from 1 to 30 (in legend labeled as "increase").

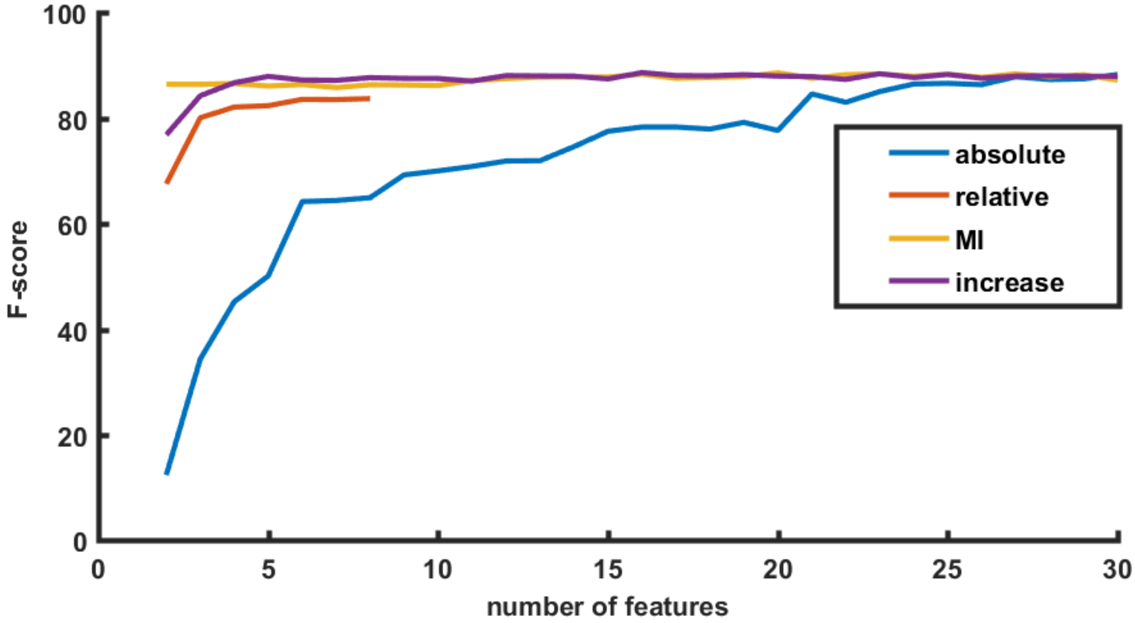

Fig. 21 F-score (7) of different number of LPC coefficients for various selection methods

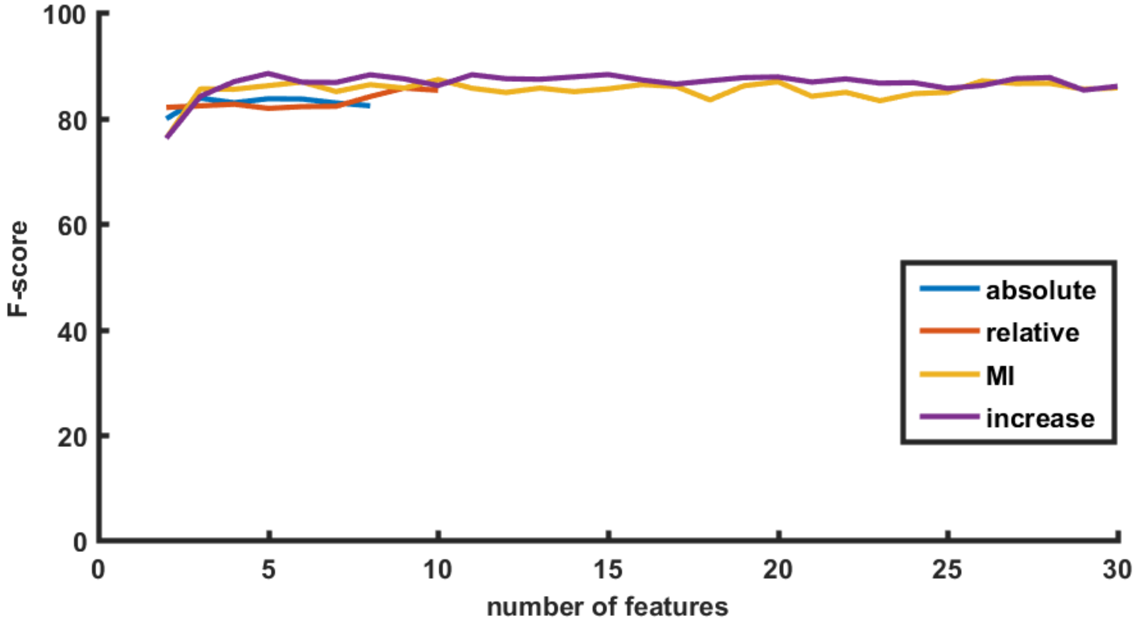

Fig. 22 F-score (7) of different number of LPCC coefficients for various selection methods

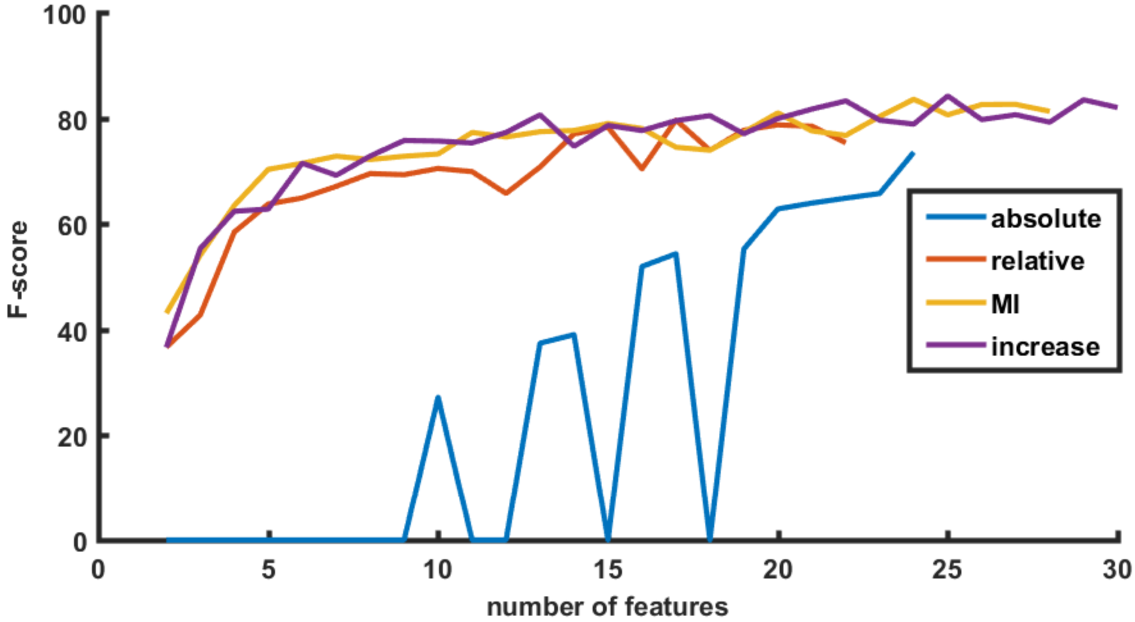

Fig. 23 F-score (7) of different number of MFCC coefficients for various selection methods

Closeness of results obtained by “MI” and “increase” might be explained by the fact, that low index coefficients have high mutual information with class labels. Relative stability exhibits slightly worse but still comparable results while absolute stability attained the worst results, probably because lowest absolute differences are concentrated in higher index coefficients, which according to (10) have comparably lower mutual information with class label than lower index coefficients.

Overall, all features achieved similar results with MFCC being slightly worse and requiring more features than LPC and LPCC.

5.2 Comparison of Various MFCC Settings

As a next step, we investigated extraction of MFCC coefficients, which has a lot of possibilities for modifications. Firstly, we describe the usual proces of MFCC extraction, then we briefly describe the dataset used for the tests. Afterwards, we describe preprocessing and postprocessing and methods of modification of extraction proces, and finally present and discuss results.

The MFCCs calculated according to the standard approach are described in [69]. First, a decadic logarithm of power spectrum is calculated (11):

$$P(k) = \log_{10}\left(\left|FT\big(s(t)\big)\right|^2\right), \tag{11}$$

where $|\mathrm{FT}(s(t))|^2$ is a square of magnitude of Fourier transform of input signal (that is, power spectrum). Afterwards, using N filter banks, usually with triangular shape, defined by (12) we filter the signal obtaining N frequency bank energies X_n (13).

$$\begin{aligned}
\Phi_n(k) &= 0 &&, \text{ for } k < k_{b_{n-1}} \\
\Phi_n(k) &= \frac{k - k_{b_{n-1}}}{k_{b_n} - k_{b_{n-1}}}, &&\text{for } k_{b_{n-1}} \le k \le k_{b_n} \\
\Phi_n(k) &= \frac{k_{b_{n+1}} - k}{k_{b_{n-1}} - k_{b_n}}, &&\text{for } k_{b_n} \le k \le k_{b_{n+1}} \\
\Phi_n(k) &= 0 &&, \text{for } k > k_{b_n}
\end{aligned} \tag{12}$$

where n is the index of filter bank, k is index of spectral coefficient, k_{b_n} is filter boundary of n-th filter and $\Phi_n(k)$ is transfer function of n-th filter. The example above is a set of equations for traingular function, however other functions shapes are described below.

$$X_n = P(k)\Phi(k). \tag{13}$$

Where X_n is energy in n-th bank and $P(k)$ is power of k-th spectral coefficient.

Filter banks are usually defined on nonlinear mel scale (this concept will be challenged in one of the first experiments in the chapter). Frequency in Hz is converted to mel-frequency by (14); this relationship also called Hertz to mel warping function is shown in Fig. 24.

$$fmel = 1127 \cdot ln(1 + \frac{f}{700}) \tag{14}$$

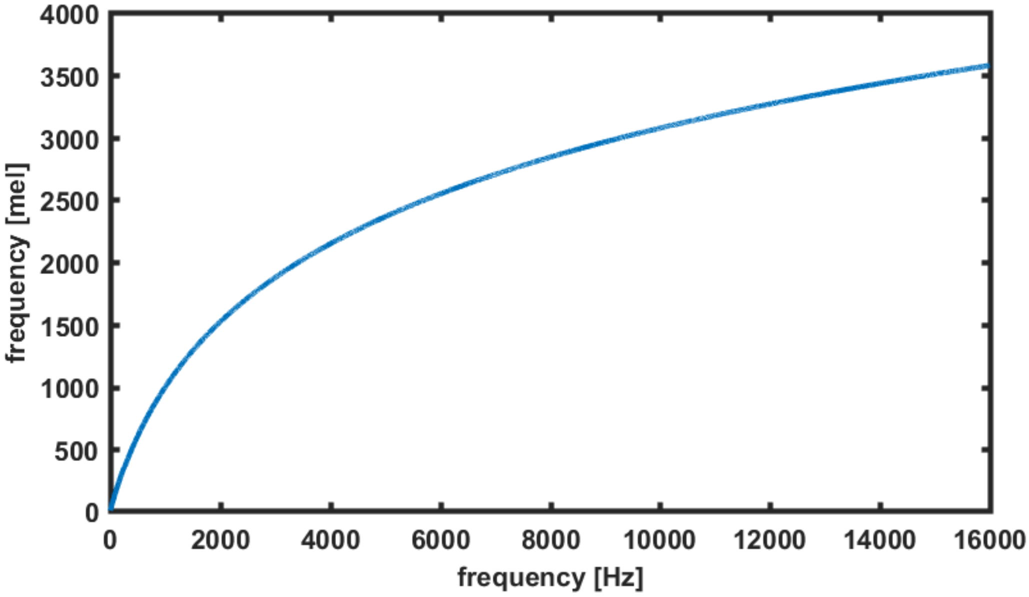

Fig. 24 Frequency scale conversion from Hz to mel

Filter boundaries are then defined by (15):

$$k_{b_n} = fmel_{lo} + n \cdot \frac{fmel_{hi} - fmel_{lo}}{N+1}, \tag{15}$$

where *fmel*$_{lo}$ and *fmel*$_{hi}$ is lowest and highest frequency in mels.

Finally, a discrete cosine transform is applied to the filter bank energies (16), resulting in a set of *M* coefficients.

$$c_m = \sum\nolimits_{n=1}^{N} X_n \cdot \cos(\frac{\pi}{N} m(n-0.5))\,\text{, for}\ \ \mathrm{m} = 1, 2, \dots, \mathrm{M}. \tag{16}$$

These steps are sometimes preceded by applying preemphasis during preprocessing stage, which can be realized by simple FIR filter. During postprocessing, cepstral liftering is sometimes applied.

We used GUDEON [70] dataset to generate audio for these experiments. We have used all 1532 gunshots and added 90% probability of added noise (consisting of various other recordings, with amplitude of at least 0.1). Non-gunshot recordings consisted of 2451 recordings of random non-gunshot sounds (with amplitude of at least 0.1). We used 60% of the data for training, 20% for validation and 20% for testing. Random data division was used respecting original ratio of ca. 40% gunshots and 60% non-gunshots for each subset. Fully connected feedforward neural networks with 1 hidden layer (consisting of 10 neurons) was used together with mean normalization, neural networks were implemented in Matlab.

Preprocessing before MFCC extraction consisted of dividing audio into non-overlapping frames of 11 ms (486 samples at sampling frequency 44.1 kHz) using rectangular window. Frames were subsequently resampled to 192 kHz and truncated to 2048 samples (from 2116 samples). After calculation of power spectrum, we have upsampled the spectrum 10x (resulting in 20480 frequency bins) in order to calculate low index coefficients using more samples than just one. Pre-emphasis as a part of preprocessing was turned off, as was cepstral liftering in postprocessing.

In this experiment, we have investigated the influence of variation of frequency bandwidth, number of filter banks, filter shapes, frequency scale (mel vs. linear) and finally MFCC order on correct gunshot recongition. Apart from this, we have also investigated the influence of audio normalization on recognition performance. F-score was used as a metric along with true positive rate (TPR) and true negative rate (TNR).

The baseline setup against which we compared the results was MFCCs of order 12 with bandwidth 1 Hz – 4 kHz, containing 24 triangular filter banks on a scale strictly linear until 1 kHz and mel afterwards (later called „linear/nonlinear"). Firstly, we compared these results to different frequency scales (linear scale, mel scale), with results presented in table 10.

Table 10 Comparison of baseline setup with different frequency scales

Frequency scale	**Metric**		
	TPR	TNR	F-score
Linear/non-linear	76.8 %	83.3%	75.4
Linear	72.2 %	86.3 %	74.4
Non-linear	75.5 %	84.1 %	75.1

In this case, results do not show any significant differences for various scales. Later results show very similar results for linear/nonlinear scale and mel scale, and, in some cases, improvement in recognition with linear scale. Table 11. below looks into how changing maximum frequency changes the results, with other options unchanged (using linear/non-linear scale). In this case, we use 24 filter banks for all frequencies, which results in filters with more bandwidth for higher frequencies. The results do not show any significant differences for various maximum frequencies, however later results with other configuration changed show that maximum frequency of 4 kHz does not achieve as good results as the rest, especially for lower MFCC order.

Table 11 Comparison of baseline setup with different bandwidths

Bandwidth	**Metric**		
	TPR	TNR	F-score
4 kHz	76.8 %	83.3 %	75.4
8 kHz	73.9 %	84.7 %	74.5
12 kHz	72.5 %	85.9 %	74.4
16 kHz	74.2 %	85.1 %	74.9

Table 12 tries setup similar to Table 11., but using different number of filter banks for each maximum frequency, so that bandwidth of all filters is the same. We can see slight improvement in performance for 8 kHz bandwidth using this setup.

Table 12 Comparison of baseline setup with different bandwidths and filter bank count

Bandwidth	**Number of filter banks**	**Metric**		
		TPR	TNR	F-score
4 kHz	24	76.8 %	83.3 %	75.4
8 kHz	32	79.4 %	84.5 %	77.8
12 kHz	37	74.5 %	85.1 %	75.1
16 kHz	41	74.2 %	84.9 %	74.8

Next experiment was similar to previous one, but here, we are using MFCC order equal to the number of filter banks. Tab. 13 summarizes the results, with 4 kHz bandwidth experiencing the most significant plunge in recognition performance, while other setups experience less pronounced, but still significant decrease.When MFCC order is too high, the performance tends to be lower than for lower orders. Possibly because too many features add noisy character to the data, another reason might be too big ratio of coefficients to filter banks (which is 1 in this case, 0.5 in Tab. 11 and even lower in Tab. 12, depending on the bandwidth chosen).

Table 13 Comparison different bandwidths and filter bank count – maximum number of coefficients

Bandwidth	**MFCC order = number of filter banks**	**Metric**		
		TPR	TNR	F-score
4 kHz	24	28.4 %	97.6 %	42.9
8 kHz	32	70.9 %	82.9 %	71.5
12 kHz	37	70.9 %	85.7 %	73.2
16 kHz	41	69.6 %	81.4 %	69.8

In another test, we used setup with maximum frequency of 8 kHz and 32 filters, which achieved the best results so far. And we investigated the influence of MFCC order on recognition performance. Due to bigger amount of data, table is not longer viable and results are depicted in Fig. 25. This tests shows that lower MFCC orders perform better and there is gradual decrease with increasing order (with significant anomalies for both nonlinear scales). Other tests support this when MFCC orders of up to 12 achieve the best results and further increase shows worse results. These results are also consistent with other features, where increasing feature order above certain value results in decreasing recognition performance.

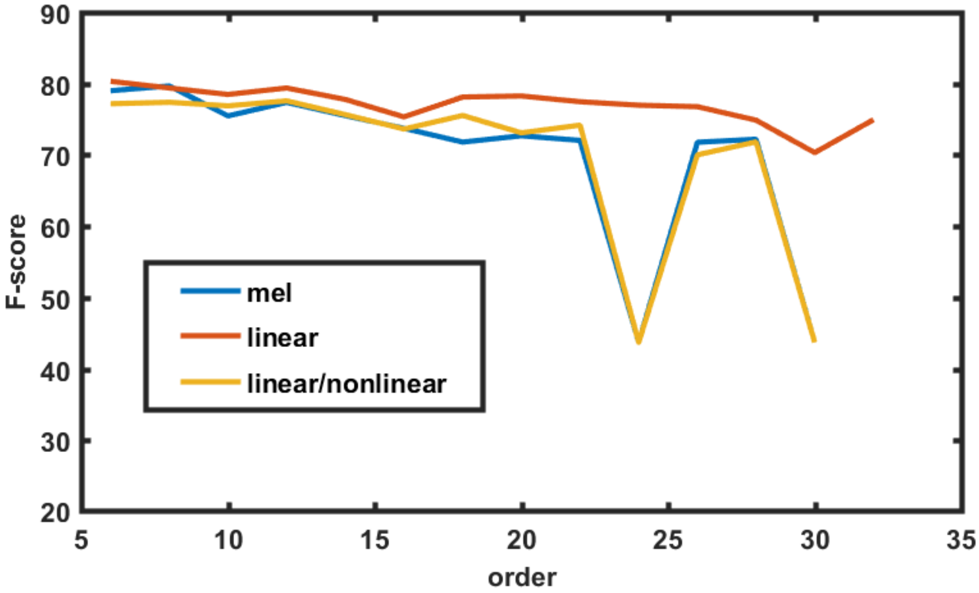

Fig. 25 Gunshot recognition with increasing MFCC order

Some publications [71] experiment with various shapes of filter banks in order to increase recognition performance. The following test compares different filter shapes. The results achieve similar F-score, however exponential filter performs slightly better in terms of True Negatives, while triangular filter performs the best for True positives. Tab. 14 shows these results.

Table 14 Comparison of various filter bank shapes

Filter shape	**Metric**		
	TPR	TNR	F-score
Triangular	76.5 %	84.5 %	76.0
Rectangular	74.5 %	83.7 %	74.2
Gaussian	75.2 %	84.7 %	75.3
Gammatone	74.8 %	84.1 %	74.7

Furthermore, Tab. 15 provides detailed comparison of various exponential filters based on [71]. All filters in tab. 15 are based on lower portion (x<0) of exponential function, s parameter describes the steepness of the function as described in [71], +1 described increased base of exponential filters.

Table 15 Comparison of exponential filter banks with different parameters

Filter shape	**Metric**		
	TPR	TNR	F-score
Exponential, s=1	71.9 %	87.1 %	74.7
Exponential, s=2	72.5 %	83.7 %	73.0
Exponential,s =1, +1	72.9 %	86.3 %	74.8

Frequency scale influence was investigated at the beginning of the section, however its effects combined with various orders were not examined. Next experiments consisted in testing performance of different MFCC orders when using different maximum frequencies, both for linear/nonlinear scale (Fig. 26) and for linear-only scale (Fig. 27). Linear-only scale performs slightly better and (except for order 22) does not contain performance anomalies such as 12 kHz performance for linear/nonlinear scale. The reason why these anomalies appear was not discovered.

We have tried to correct the anomalies by normalizing audio signal so that maximum value in every frame is 1, which resulted in no performance anomalies, but performance in general was significantly decreased (by approx. 7%).

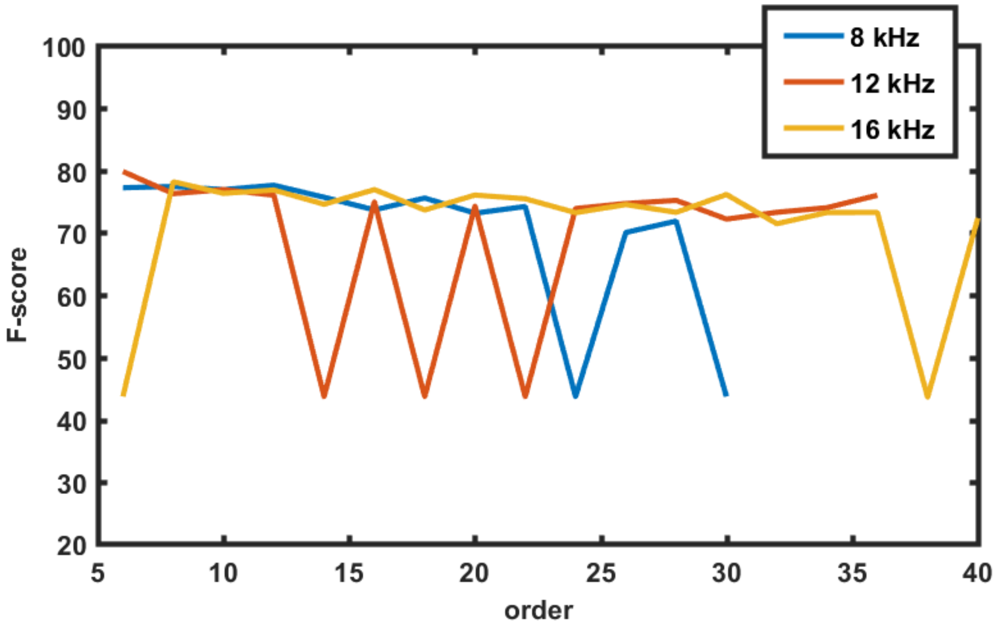

Fig. 26 Comparison of performance for different orders and bandwidths with linear/nonlinear scale

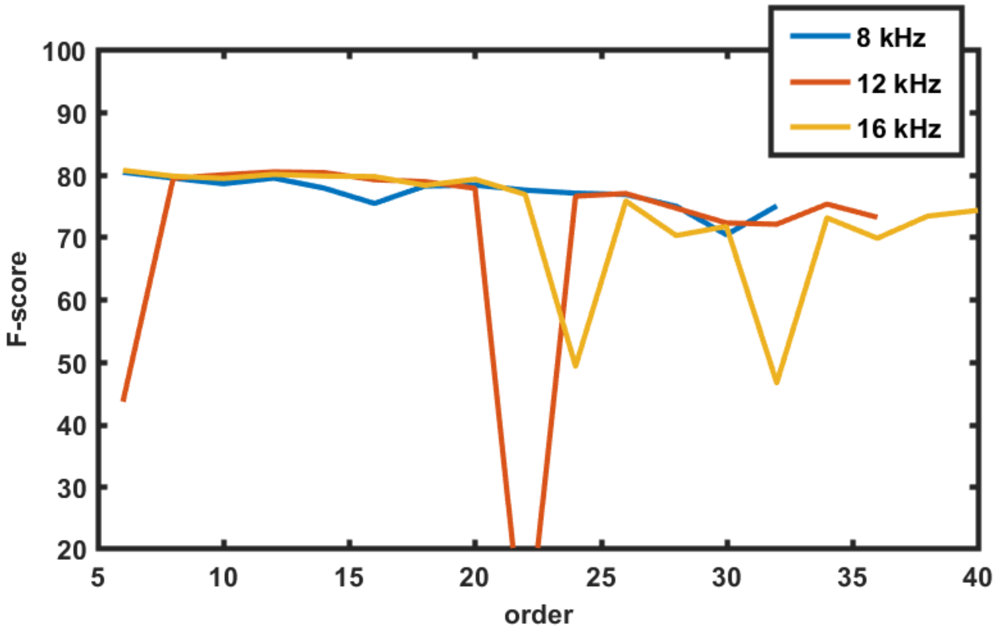

Fig. 27 Comparison of performance for different orders and bandwidths with linear scale

As could be seen throughout the experiments, the results for lower feature orders do not change dramatically with increasing bandwidth for filter banks. This conclusion is consistent with previous finding, that most of the energy in gunshots is concentrated in lower frequency bands. Another conclusion would be, that increasing feature order usually causes recognition performance to decrease, or at least the performance does not experience further increase. An important factor is also a ratio of feature order to number of filter banks, where ratio around 0.5 performed the best. Regarding filter shapes, most of them achieved comparable results, but it is possible that some filter shapes result in better true positive rate, while other achieve better true negative rates, at least in gunshot recognition task.

As a result, we conclude that it is better to use linear frequency scale. The best performing bandwidth appears to be 1 Hz – 8 kHz, with 32 triangular filter banks. We have chosen MFCC order 8 to be used during further experiments with real-time gunshot detection.

5.3 Effects of Noise Levels and Types

Until now, all sounds used in experiments contained no additional noise, apart from noise present during recording and the noise introduced by recording devices and processing. In this chapter, performance of previously used features under adverse noise conditions is investigated.

During the tests, multiple noise types and noise levels were used. White noise was chosen because of its spectral characteristics, and widespread use of white noise during testing. Additionally, sounds were combined with sound of rain and sound of idling engine, which also served as noises, under various SNR. Waveforms and spectra of those additional noises are shown in Fig. 28 to 31, it can be seen, that rain has wider spectrum while spectrum of engine is concentrated in low frequencies, with most energy concentrated below 1 kHz. Signal-to-noise ratios (SNR) were set to 0 dB, 10 dB, 20 dB and 30 dB. During tests, recordings with equal amount of noise were used both for training and for testing. Paper [42] compares results when using different SNR for training and testing sets.

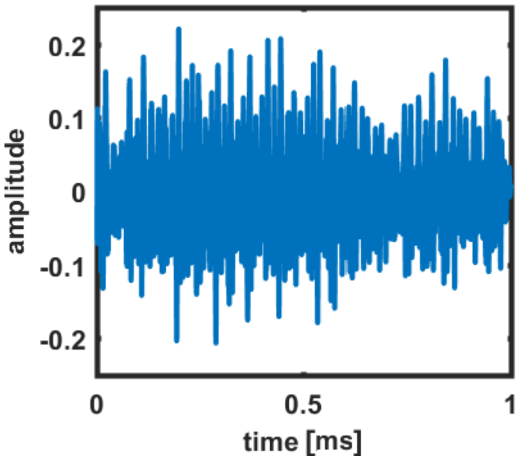

Fig. 28 Engine in time domain

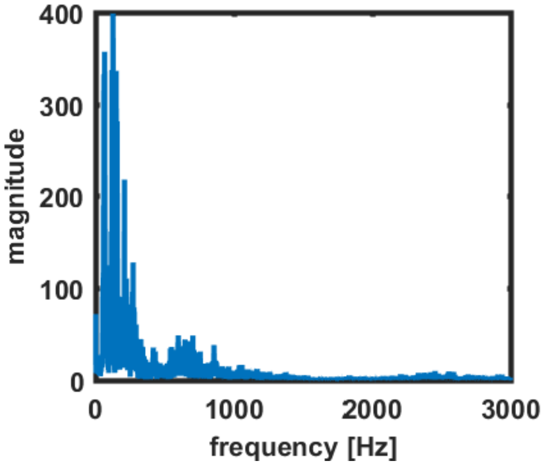

Fig. 29 Engine in spectral domain

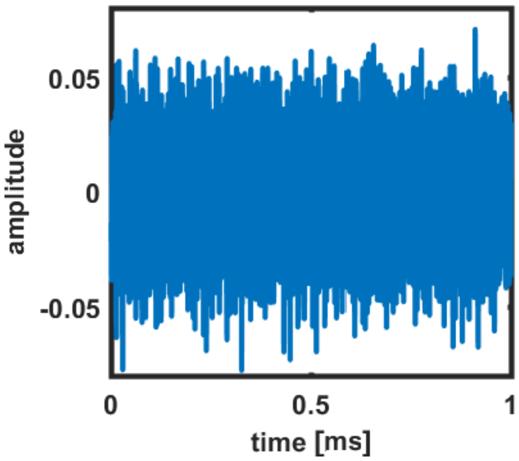

Fig. 30 Rain in time domain

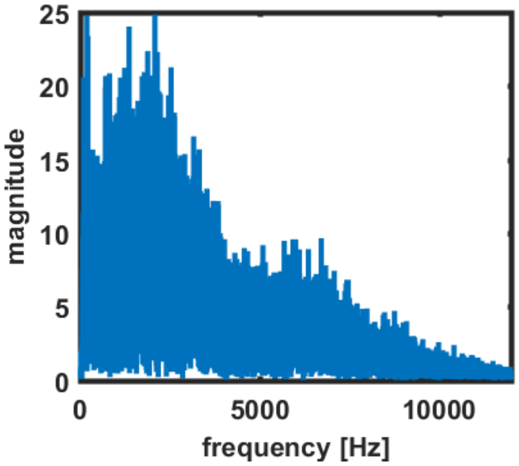

Fig. 31 Rain in spectral domain

Figures 32 to 34 show F-score for all features with different orders and signal-to-noise ratios. At 30 dB, LPC and LPCC achieve similar results, while MFCC shows inferior results. With SNR decreasing to 20 dB and 10 dB, LPC and LPCC F-score significantly decreases, at the same time, MFCC keeps roughly original values, which are even better than those of other features. At 0 dB, LPCC experiences significant decline with growing order, which results from decrease in precision (not so much recall). MFCC and LPC achieve comparable results, with LPC achieving better results at recall and MFCC at precision. Results in this chapter are based on [72].

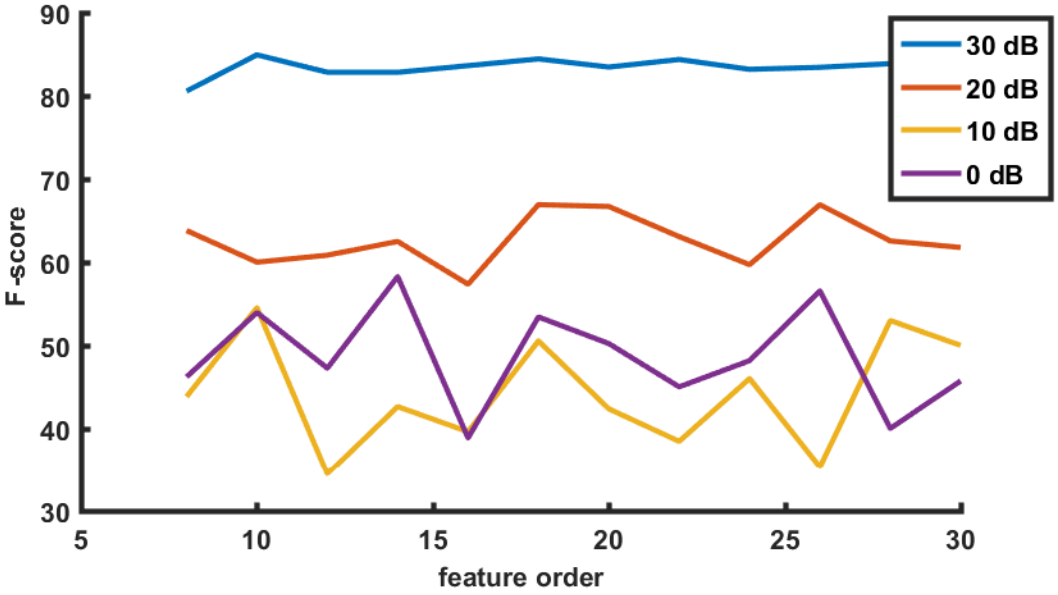

Fig. 32 F-score (7) of LPC coefficients of various orders for different SNR

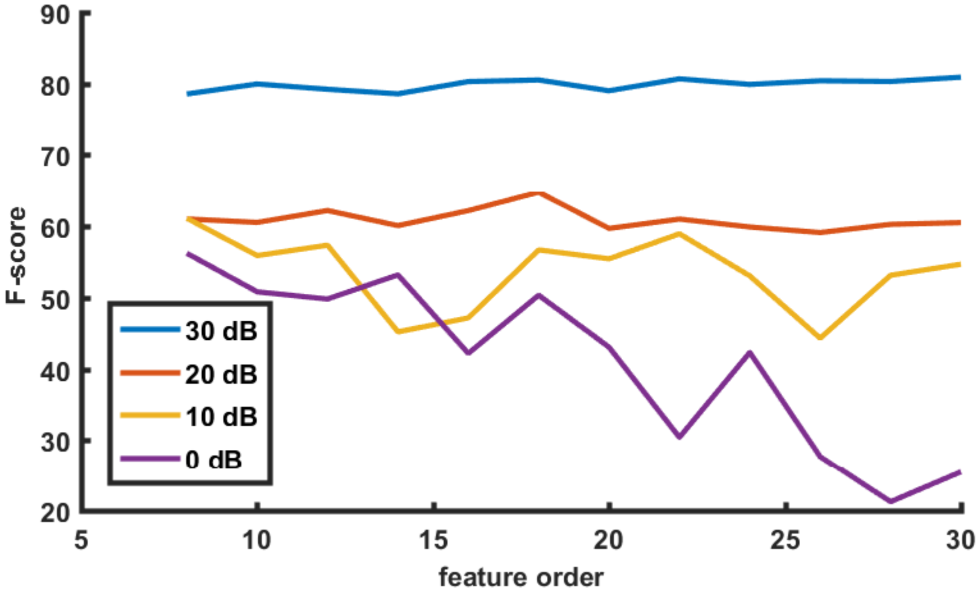

Fig. 33 F-score (7) of LPCC coefficients of various orders for different SNR

Recognition performance of LPC and LPCC for high SNR ratio sis mostly flat throughout whole feature order range. However with decreasing SNR, more random character is visible, reminiscent of MFCC, this is especially true of LPC, while LPCC keeps its monotonous character much better.

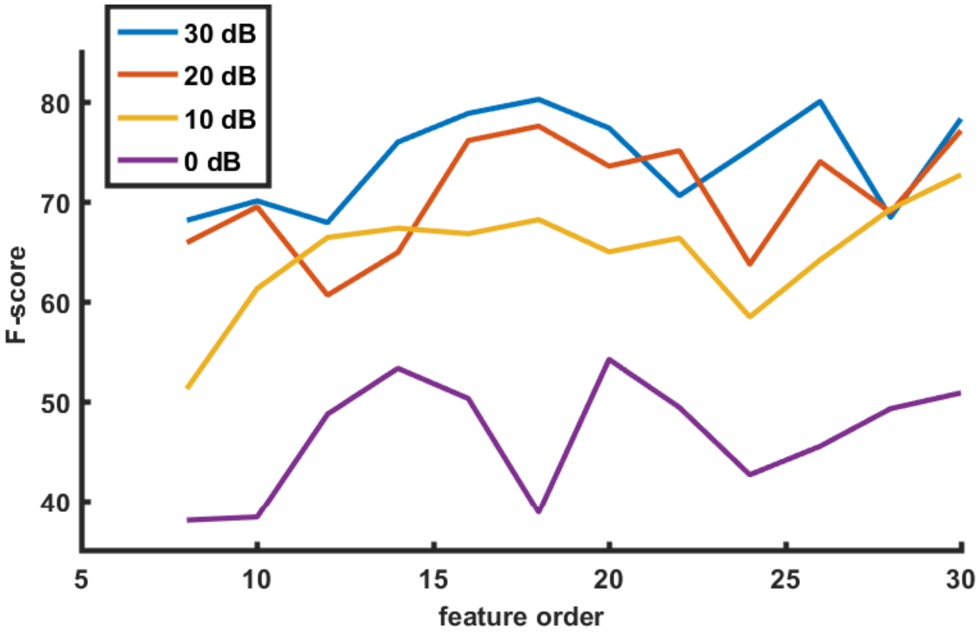

Fig. 34 F-score (7) of MFCC coefficients of various orders for different SNR

Overall, similar results as in previous steps were observed, that increasing feature order beyond certain point does not significantly influence recognition performance. The only exception being LPCC at 0 dB, where recognition performance dramatically decreases, this might be attributed to their increased noise sensisitivy in comparison with LPC or MFCC. Trends under degradation with engine and rain sounds were similar to white noise, but the degree was slightly different.

5.4 Feature Performance Comparison Based on Distributions

We propose a novel approach to feature comparison that, to our knowledge, was not presented elsewhere. The approach consists in comparing feature mean and actual values for multiple categories. The proposed output would be area under negative part of fitted distribution, which represents percentage of misclassified segments. Below, we describe the method in greater detail and provide some outputs. Please note, that these outputs were only tested for three categories (gunshots, barking dog and car horn) so they might provide more extreme values and with added categories, results might be more continuous.

In the first step, mean (and for later use also standard deviation) of examined features from all recordings is calculated – we tested this approach with LPC coefficients of even orders 8 through 30. Then, one feature and one category is selected. Feature values of individual recordings are then calculated and the distance to mean value from that category and nearest other category are compared. Histogram of such data gives us information on quantity of data closer to negative class. In order to quantify classification capacity, we proceed to fit the histogram with distribution and measure area under negative part (i. e. values closer to incorrect class).

As to the choice of proper fit, we turned to function from mathworks file-exchange webpages for „allfitdist“ function. This function takes data as an input and fits various distributions to it, then compares Bayesian Information Criterion (BIC), or other metric of choice, and establishes best fits. For now, we use different distributions for different runs, but we consider using only one distribution for all fits for better comparison. Table 16 shows information provided by our function for gunshot category (feature order 30, feature index 20). Column „Distribution“ lists 4 best distributions according to BIC (second column) – the lower the BIC, the better the fit is. Third column, „Negative Area“ shows percentage of area of distribution that is in negative values, i. e. theoretical percentage of misclassified recordings when only this feature is used.

Table 16 Distribution fit comparison

Distribution	**Bayesian information criterion**	**Negative Area [%]**
Extreme value	1035.015	36.477
Normal	1114.886	44.102
Tlocationscale	1121.743	44.100
Logistic	1239.840	42.296

Simpler variation of this approach might be just to compare histogram counts of values (difference of distance from mean of incorrect class and mean of correct class) closer to incorrect class and those closer to correct class. Fig. 35 illustrates such a histogram for MFCC values of order 20 and feature index 6 with correct class of gunshots and incorrect classes of barking dog and idling engine. It can be clearly seen that there is more instances of samples closer to the correct class, but this is not always the case.

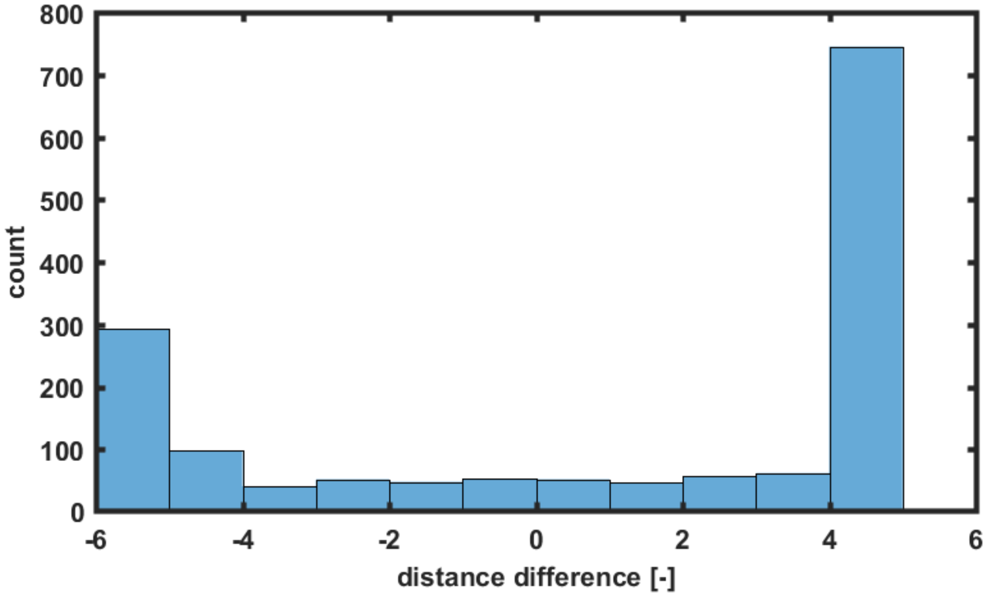

Fig. 35 Histogram of difference of samples between mean correct and mean incorrect

This way, we can rate each feature on how it will perform and then choose the optimal features for given problem. Fig. 36 illustrates this scenarion, x-axis represents number of MFCC features with scenario „increase“, where we progressively add features from index 1 to index 20 and scenario „distance“, where we used this measure to order the features. The illustrated example shows neural network performance with classes gunshots and barking dog. It can be seen that the best performance is achieved using just 3 features when „distance“ is used, whereas with simply using features sequentially, we need to use 13 features.

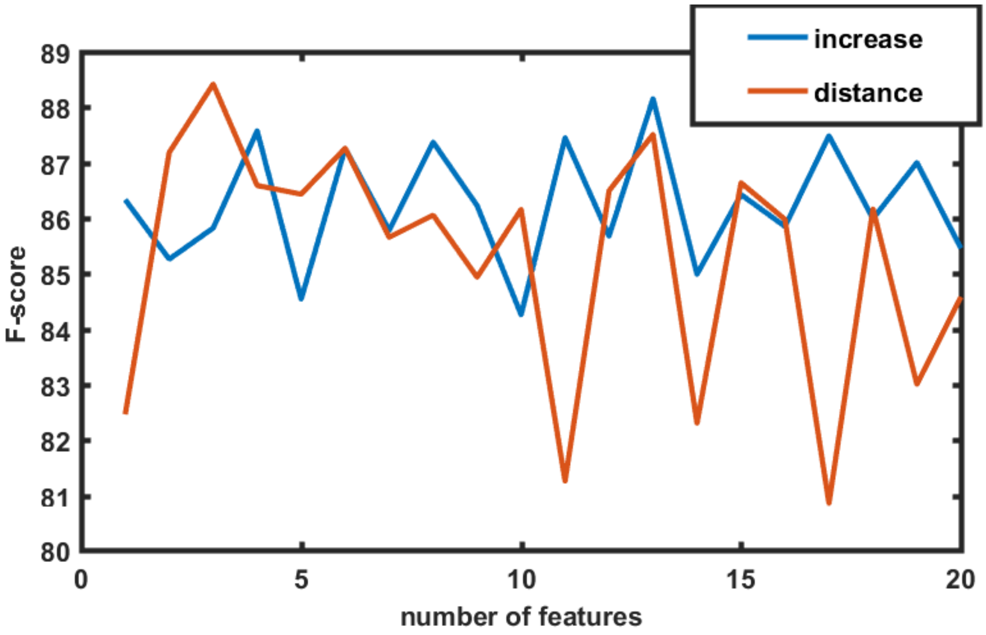

Fig. 36 F-score for two different feature selection methods

Since we are comparing only distribution of a single feature, so far this method cannot tell us about expected performance of multiple combined features. We can however compare all features one by one and choose the best performing, but there would still be an unresolved issue of mutual information between various features, so the actual improvement would be less. This can also mean, that there is not monotonous increase in recognition performance, but that global maximum is not the first maximum.

6 RECOGNITION ALGORITHMS

6.1 Overview of Recognition Algorithms

Sound event detection (SED) in general, and gunshot recognition in particular used a variety of different recognition algorithms throughout the existence of this field. Among the first approaches are gaussian mixture models (GMMs) and hidden Markov models (HMMs) [73]. Later approaches include support vector machines (SVMs), at first with linear kernel, later with non-linear kernels (such as Radial Basis Function), the latter achieving better performance [66]. State-of-the-art results are achieved by neural networks, which range from feedforward fully-connected networks, to convolutional and recurrent networks [74]. High performance of the networks comes with a downside of problematic interpretation of processes and learned representation in hidden layers of the networks. For a brief comparison of supervised machine learning algorithms, see for example [75]. For a more detailed explanation of basic machine learning algorithms, see [76].

6.2 Naïve Bayes Classifier

Naïve Bayes classifier works with a conditional probability model, using Bayes theorem (17),

$$p(Y_n|\mathbf{x}) = \frac{p(Y_n)p(\boldsymbol{x}|Y_n)}{p(\boldsymbol{x})}, \tag{17}$$

where **x** is a feature vector and Y_n is nth class label, $p(Y_n|\mathbf{x})$ is a probability of feature vector **x** calculated from observing nth class, $p(\mathbf{x}|\ Y_n)$ is a probability of obtaining feature vector **x** with nth class and $p(\mathbf{x})$ is distribution of features. In general case, we make no assumptions about features, so if we use 3 features, (17) will be of the following form (18):

$$p(Y_n|\mathbf{x}) = \frac{p(Y_n)p(x_1|x_2,x_3,Y_n)p(x_2|x_3,Y_n)p(x_3|Y_n)}{p(\boldsymbol{x})}, \tag{18}$$

i.e., with growing number of features, we would have to consider growing number of dependencies, which would increase computational demands immensely. Instead, we assume mutual independence between all features (thus the *Naïve* classifier) and simplify the equation to a form of (19).

$$p(Y_n|\mathbf{x}) = \frac{p(Y_n)p(x_1|Y_n)p(x_2|Y_n)p(x_3|Y_n)}{p(\boldsymbol{x})} \tag{19}$$

In order to classify an input with a set of features, we simply calculate probabilities for all classes, and then pick the one with maximum a posteriori probability, according to (20).

$$\hat{y} = \max_{n\in(1,\ldots,N)} p(Y_n)\prod_{i=1}^{F} p(x_i|Y_n), \tag{20}$$

where x_i is i$^{\text{th}}$ feature out of F features total.

6.3 Decision Trees and Random Forests

Decision trees consist of a series of binary decisions, working with pairs: feature and threshold. Number of decisions is equal to number of features in feature vector. The progressive decisions finally lead to a classification into one of two or more classes. Decision trees can work with unnormalized data [76], since internal structure is not influenced by feature values.

The construction of decision tree progresses from root node, the first decision node, which is chosen so that it best separates the classes to leaf nodes, which designate the final class. There are multiple ways to determine which feature best separates the dataset and we are going to briefly describe them, but first, we introduce the term impurity. Ideally, we want to achieve zero inpurity, so that one feature, or decision eliminates whole subset of classes. However this is not always the case, and often there are still both classes present after one decision (e.g. classifying gender using height, men are usually taller than women, but there is still going to be tall women and shorter men) [76].

First impurity measure is called Gini impurity index, it is calculated using (21).

$$I_{Gini}(x_i) = \sum\nolimits_n p(Y_n|x_i)(1 - p(x_i|Y_n)), \tag{21}$$

the Gini index measures probability of misclassification when decision is taken with feature x_i. This index reaches minimum – zero – value, when all samples in one node are classified into one category.

Another measure is Cross-entropy impurity index, defined by (22).

$$I_{cross-entropy}(x_i) = \sum\nolimits_n p(Y_n|x_i) \log p(Y_n|x_i), \tag{22}$$

this measure assumes maximum value in case of uniform distribution in classification into classes. That means the higher the value, the deeper the tree will have to be. While Gini index informs us of missclassification probability, cross-entropy helps us minimize uncertainity about the classification.

The decision tree is grown until we achieve either all pure nodes (i.e. no uncertainity in classification), zero information gain, or maximum tree depth (that is, we run out of features).

Lastly, there is misclassification impurity index (23).

$$I_{misclassification}(x_i) = 1 - p(Y_n|x_i), \tag{23}$$

in [76], misclassifiaction index is described as poorest choice, since i tis not as sensitive to different probability distributions.

Random trees employ ensemble learning, where we use multiple Decision trees which are trained slightly differently (either different features, different subsets of training data or both) and so achieve slightly different results. This way, we obtain a number of class labels from which we choose the most frequent, this usually leads to increase in performance and prevents overfitting.

6.4 k-Nearest Neighbors

k-Nearest Neighbors (k-NN) is a simple algorithm which, in order to classify an input, looks into its nearest neighboring data points and assings the input to the most frequent neighbor. Euclidean distance is the most common choice of metric in case of continuous input features.

This algorithm has a parameter *k*, which determines how many neighboring data points will the algorithm check. In order to choose the optimal *k*, we usually test multiple choices and choose the best performing one. In order to avoid ties, we have to use odd number for *k*.

This algorithm is particularly susceptible to curse of dimensionality, i.e. problems occuring with higher dimensional features, where it can be demanding to compute distance in high dimensional spaces, or the distance differences are not that significant anymore. In these cases, dimensionality reduction techniques are usually employed [77].

6.5 Support Vector Machines

Support vector machines (SVMs) are one of the more complex and better performing supervised learning algorithms. This can also be proven by their longtime superior performance in MNIST dataset classification [76].

SVM is an algorithm which tries to find the best separation between two classes (we have to use multiple one verus all classifiers for multiclass problems) by maximizing the margin that separates the hyperplane and marginal datapoints (the datapoints on the edges are called support vectors). In its simplest form, SVM is non-probabilistic a linear classifier, however sometimes linear classification is not possible, because classes are not linearly separable. In cases like these, mapping to a higher dimensional space, where the separation can be achieved, is performed. The mapping is done using kernel trick, it can be achieved using various kernel functions and the new dimension datapoints consist of combination of original datapoints. Among common kernel functions are polynomial kernel, hyperbolic tangent, or most frequently used gaussian radial basis function.

Among other parameters that tune SVMs is a regularization parameter (sometimes called C parameter), this expresses trade-off between margin width and number of misclassifications. That is, high C values provide high separation margin but tolerate more misclassifications, conversely lower C values lead to fewer misclassifications but also smaller margin width, this can be seen as a tradeoff between high bias and high variance respectively.

Another parameter that changes training behavior of SVMs is called gamma parameter. Gamma parameter influences how many points are considered when calculating the separating hyperplane, with lower values including also points farther from the divide and higher values considering only points close to the other class.

6.6 Neural Networks

Neural networks (NNs) are supervised machine learning algorithms with extensive use in various fields. They achieve state-of-the-art performance in polyphonic (when multiple audio sources are active at the same time) SED. Neural networks are formed by multiple individual nodes (called neurons) and connections between them in a layered architecture, every layer can contain any number of neurons. NNs are non-linear classifiers due to their usage of non-linear activation function and layered architecture.

6.6.1 Neuron

Neuron is a basic building unit of neural networks. Each neuron has *N* inputs and one output. Output value of a neuron is determined as an output of neuron's activation function and weighted sum of inputs plus bias (24):

$$y = f\left(\sum_{i=1}^{N} w_i x_i + b\right), \tag{24}$$

where y is neuron output, x_i is i^{th} input to the neuron and w_i is corresponding weight, b is bias term and f is the activation function. Although activation functions can be linear, using nonlinear functions introduces nonlinearity to the system and allows solving nonlinear problems.

Among most frequently used activation functions belong sigmoid (25), which has output range limited to interval (0, 1) and is continuously differentiable (depicted in Fig. 37). Hyperbloic tangent function (26), with properties similar to sigmoid, but range limited to (-1, 1), this function is frequently used with machine vision approaches. And lastly rectified linear unit (27), shown in Fig. 38, which is not continuously differentiable, but is used in state of the art approaches and generally achieves better results than sigmoid function. Special case of activation function is so-called softmax function, frequently used in last layer of classification neural networks which output single category. Softmax function (28) outputs probability of different output classes given certain input features. On the other hand, when we are using multiclass classification (as in polyphonic sound event detection), we can use thresholded sigmoid function, instead of softmax function.

$$y(x) = \sigma(x) = \frac{1}{1 + e^{-x}}. \tag{25}$$

$$y(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \tag{26}$$

$$y(x) = \begin{cases} 0 & \text{for } x \le 0 \\ x & \text{for } x > 0 \end{cases}. \tag{27}$$

$$y(x) = \frac{e^{x}}{\sum_{N} e^{x_n}}, \tag{28}$$

where y(x) is activation function and x is input to the function.

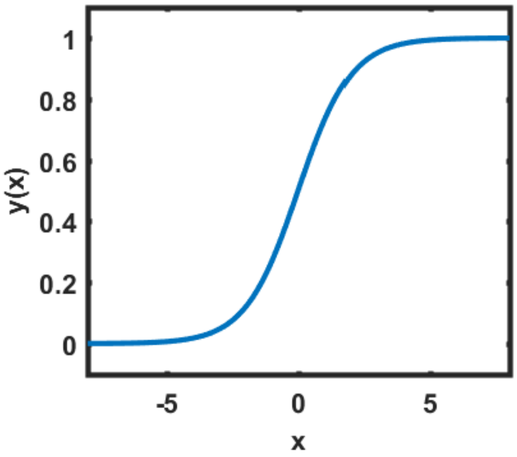

Fig. 37 Logistic (sigmoid) function

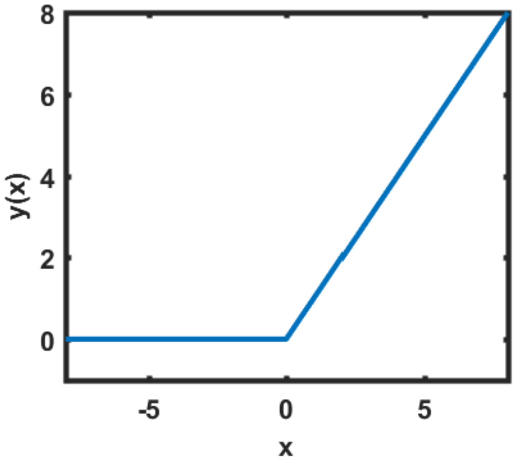

Fig. 38 Rectified linear function

6.6.2 Fully connected networks and training

Among the first types of networks developed was feedforward fully-connected network, which is depicted in Fig. 39. In this type of network, inputs are fed to the first layer of neurons, where they are linearly transformed by neuron-specific weight and bias and fed into activation function. In this fashion, information progresses through arbitrary number of hidden layers until it reaches output layer, which has number of neurons corresponding to the number of output classes. The key point is, each neuron in one layer uses output of all neurons in previous layer as input and outputs to all neurons in the following layer.

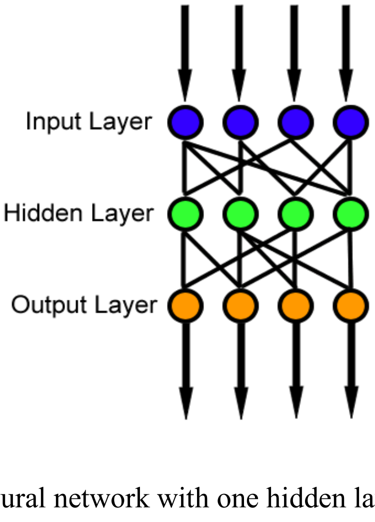

Fig. 39 Schematic depiction of neural network with one hidden layer and 4 neurons in each layer

When first initializing a neural network, weights and biases are chosen at random, to improve recognition performance, training stage takes place. Training stage is an optimization tasks where weights and biases are iteratively updated through backpropagation algorithm to minimize the error. Error is expressed through a loss function, such as (29):

$$E = \frac{1}{2n}\sum_{x} \|y(x) - y'(x)\|^{2}, \tag{29}$$

where E is error value, n is number of examples, y is actual class label and y' is predicted class label. Training is done over a training set, one pass over whole training set is called an epoch, number of epochs ranges from tens to hundreds or even thousands. Optimization is usually implemented using stochastic gradient descent and its various modifications, Adam technique is used in current state of the art approaches.

Stochastic gradient descent consists of updating the weights in the direction opposite to the gradient of loss function (such as (29)). Learning rate is a parameter that influences how big step is taken. Equation (30) shows how weight increment is calculated each step (epoch). This algorithm can be implemented so that the weights are updated after different number of training examples (n in (29)). Using more examples leads to smoother progress, but is more demanding memory-wise.

$$\Delta w_{i,j} = -\eta \frac{\partial E}{\partial w_{i,j}}, \tag{30}$$

where $\Delta w_{i,j}$ is an update to weight of i^{th} neuron in j^{th} layer, η is learning rate and $\frac{\partial E}{\partial w_{i,j}}$ is a partial derivation of loss function (e.g. (29)) with regard to weight $w_{i,j}$.

The learning process was prone to overfitting, so multiple methods to avoid it were concieved. One of the first was early stopping and validation. This approach consists of comparing classification performance on tranining data (which has a tendency to improve on each pass, also called an epoch) to performance on so called validation data. Validation dataset consists of data that is not used during training and is considered as unseen by the classifier, when performance on validation data starts to drop, training is stopped.

Another approach to counter overfitting is implementing *dropout* [78]. This idea consists of randomly dropping out neurons (together with their connections) during training. Dropout comes with a parameter, dropout rate, which can be adjusted per layer and gives a percentage chance of neuron dropping out. State-of-the-art SED systems use dropout of 25% in convolutional layers [60].

Regardless of architecture, good practice is to normalize all inputs to neural network so that they have zero mean and standard deviation equal to one. This prevents some features to exert undue influence, such as when one is in range 0-1 and another 1-1000. Lately, this approach was extended from the input layer to all hidden layers, in what is called *batch normalization* [79]. Batch normalization subtracts mean and divides by standard deviation inside each hidden layer, just before activation function. This process improves learning speed, allows use of deeper networks, higher learning rates and provides less sensitivity to random weight initializations.

6.6.3 Convolutional networks

Neural networks were soon modified and new layers invented. One such layer was *convolutional* layer, which does not use different weight for every input datapoint, but shares a set of weights (which is called *filter*, and has variable dimensions) repeatedly for all datapoints, basically applying convolution on the input. This process is depicted on Fig. 40. Filter dimensions are usually rectangular, with typical values of 3x3 or 5x5, bigger dimensions were discontinued because of high computational demands, however new hyperparameter optimization algorithms found out, there can be use also for filters 7x7 and possibly bigger [80]. Apart from filter dimensions, convolutional layers have parameters such as *stride*, which determines how much is filter moved. Stride values higher than 1 shrink the input dimensions. An optional parameter, *padding* around input data, is used to prevent dimension shrinking caused by convolution on the edges of the input. Activation functions are used in the same way as with fully-connected layers, where one element of the filter is one neuron of the network.

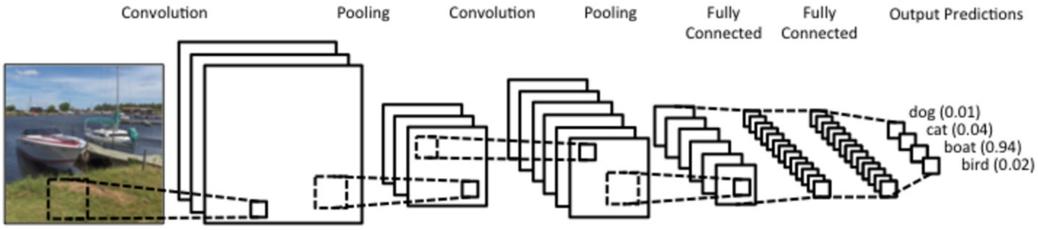

Fig. 40 Convolutional network with max-pooling layers

Convolutional layers were mostly used in image recognition at first, later they were incorporated in audio recognition (using spectrograms) with promising results. Along with convolutional layers, another frequently used layer is *pooling layer.* Pooling is a form of subsampling and is characterized by its dimensions, i.e. how much do we subsample in which dimension. Pooling is mostly realized as max-pooling, which preserves only maximum value in given area (discarding spatial information), but it can also be realized by averaging the area.

It was shown that different filters of convolutional networks learn to recognize various features, or shapes, such as edges or corners. Filters in higher values learn to recognize more complex concepts, such as cars or animals.

6.6.4 Recurrent networks

Another type of layer is *recurrent* layer, as opposed to feedforward layers, these layers incorporate a feedback, which, depending on the exact function, retains part of the value from previous timestep. Common means of implementing recurrent layers is using either long short-term memory units or gated recurrent units (illustrated in Fig. 41). They found usage especially in non-impulsive signal recognition, such as speech recognition or music classification.

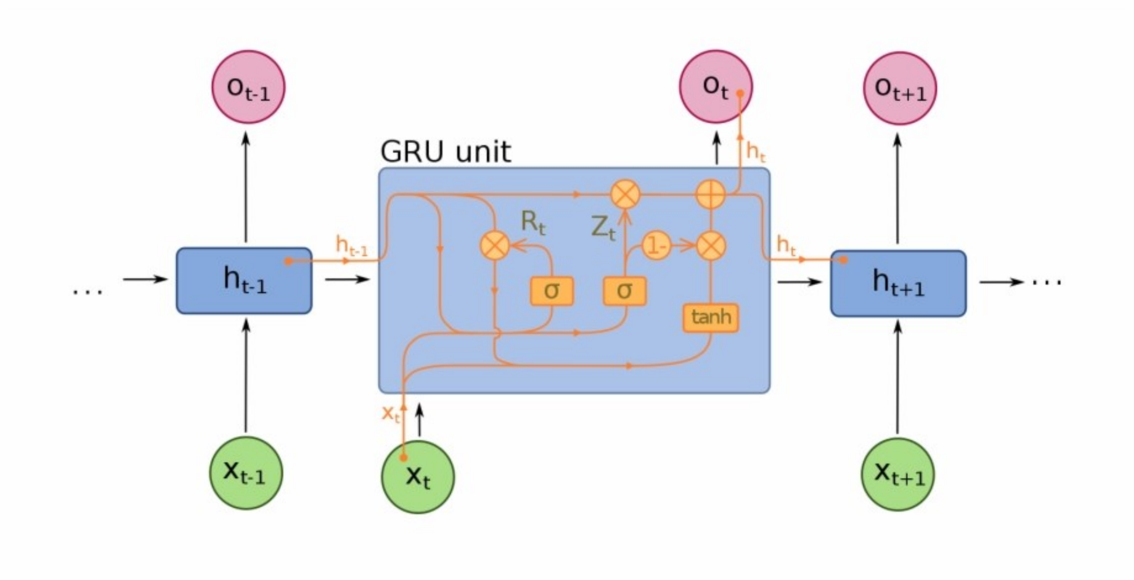

Fig. 41 Schematic depicition of gated recurrent unit

6.7 Hidden Markov Models

Hidden Markov Models (HMMs) can be characterized as a probability distribution over sequence of vectors. It is characterized by the number of states in the model, number of observation symbols per state, state transition probability distribution, symbol probability distribution in each state (GMMs are frequently used) and initial state distribution [81]. Authors in [73] use viterbi algorithm for classification (i.e. decode the optimal path) and expectation-maximization algorithm during training phase. Comparing MFCC and mel energies as features, and using several topologies (i.e. number of states), it achieves 24% performance at 0dB SNR in classification into 61 classes.

7 CONTINUOUS AUDIO EVENT DETECTION

The whole approach is based on audio signal processing in two stages, namely *Preliminary gunshot detection* (first stage) and subsequent *Advanced gunshot detection* (second stage). In the first stage, the sound scene around the sensor (microphone) is continuously captured, and shot-like sounds are sorted into group of individual shots and group of burst. Then, signals in both groups are stored for further advanced analysis in separate buffers. In the second stage, all sounds in the buffers are individually investigated in order to detect a gunshot or reject other shot-like sounds. Fig. 42 shows the concept.

The first section of this chapter introduces the idea of continuous monitoring further, along with how to deal with preliminary flagged segments. Second and third sections deal with preliminary detection of bursts and gunshots respectively.

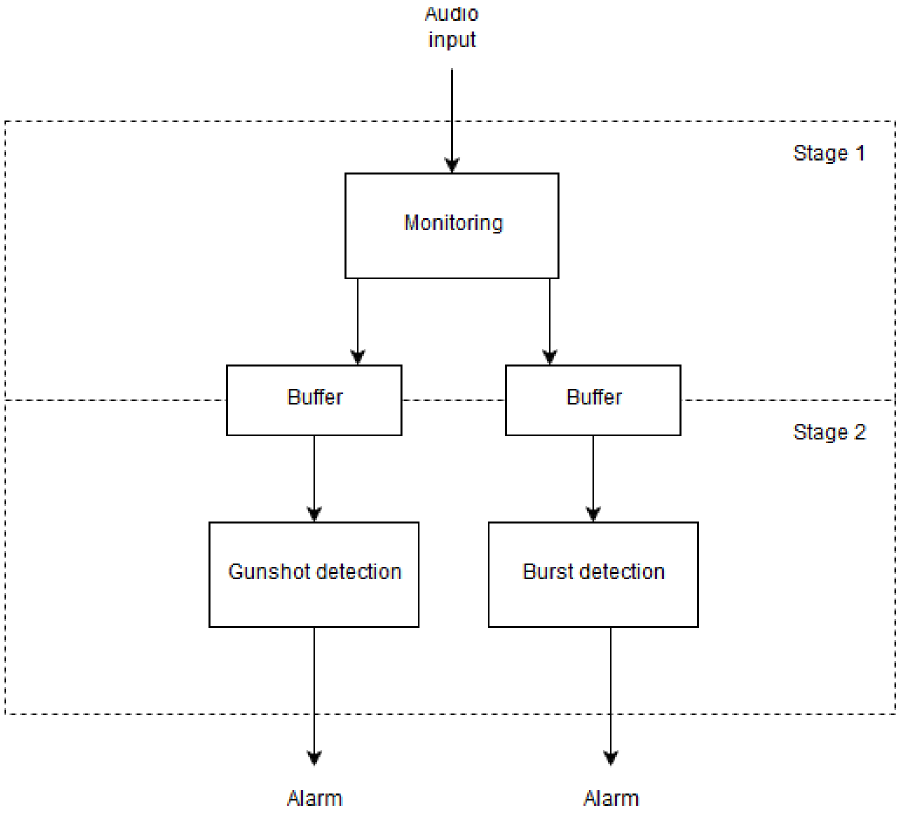

Fig. 42 Diagram of the whole two stage concept

7.1 Continuous Monitoring of Audio Events

This chapter details our work on continuous audio input monitoring in order to detect gunshots or gunshot bursts. It will introduce the concept in general, including preprocessing and basic filtering. Gunshot and burst detection will be described in dedicated subsections. Fig. 41 represents a block diagram of our approach. Bold capital letters in some blocks identify signals of interest, shown in Fig. 42 below.

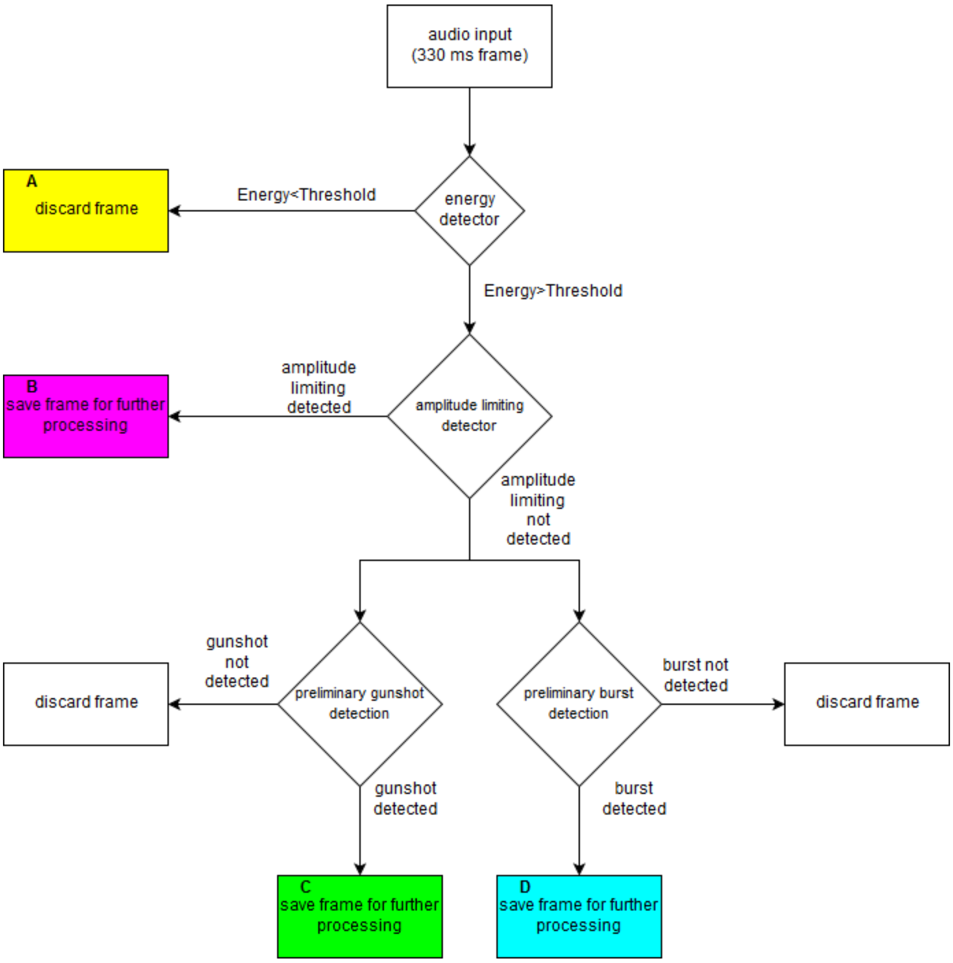
audio input
(330 ms frame)
energy detector
Energy<Threshold
A
discard frame
Energy>Threshold
amplitude limiting detector
amplitude limiting detected
B
save frame for further processing
amplitude limiting not detected
preliminary gunshot detection
gunshot not detected
discard frame
gunshot detected
C
save frame for further processing
preliminary burst detection
burst not detected
discard frame
burst detected
D
save frame for further processing

Fig. 43 Block diagram of continuous audio event monitoring system – 1st stage

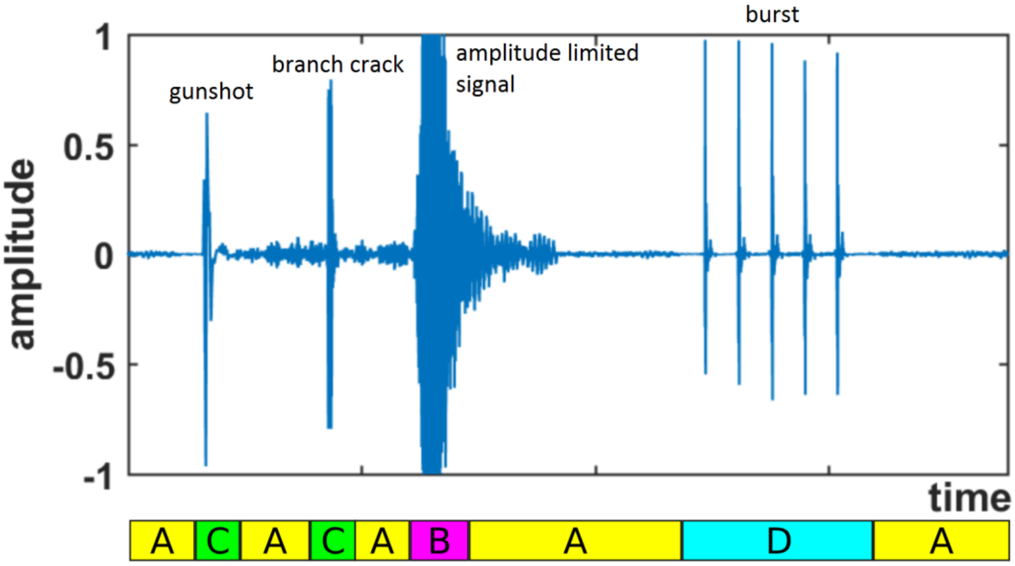

Fig. 44 Four basic sound categories as preliminarily classified by continuous audio event detection

Continous monitoring works with audio frames of length 330 ms without overlap (14553 samples with sampling frequency of 44.1 kHz). This length was chosen because methods for burst detection, mentioned in the following section, work with at least 3 periods of signal in a time-frame, which results in 300 ms for weapons with slowest rate of fire (10 rounds per second) in our audio dataset, extra 30 ms is a reserve (since, as will be presented later, period detectors report up to 10% deviation on individual periods). Overlap was not introduced because of the need to save computational power. Audio input is sampled at 44.1 kHz with 16-bit quantization, as these are frequently used values for this task and provide reasonable compromise between resolution and power demands.

Next step is to check input signal energy, and in case no (or very weak) signal is detected, we discard the frame and do not continue with other operations. Energy is calculated over whole time frame as a sum of squared samples. Energy threshold was chosen so that every single gunshot burst in used audio dataset passes the criterion, the resulting value was set to 0.3. If the signal is stronger than this value, check for amplitude limiting takes place. Amplitude limiting is checked by counting number of values close to, or at maximum absolute values (in case of normalized signal, the values are +1 and -1) and comparing this number to a threshold. The threshold was estimated observing our audio dataset, and was experimentally set to 30 samples, i.e. if more than 30 samples in the whole frame are very close to maximum values, the frame is flagged as containing amplitude limited signal. Amplitude limited frames are saved for later processing (as they will require approach different to non-limited signal) and no further action is taken. Fig. 45 – 47 below illustrate deformation of spectrum with increasing amplitude limiting. Fig. 45 shows original gunshot waveform and its spectrum normalized to maximum value 1. Fig. 46 shows lightly amplitude limited waveform of the same signal with almost no discernible change in spectrum. Fig. 47 shows heavily amplitude limited signal with noticeably different spectrum, introducing more peaks in higher frequencies. These figures demonstrate the need for separate treatment of amplitude limited signals when we are dealing with features derived from spectrum or waveform.

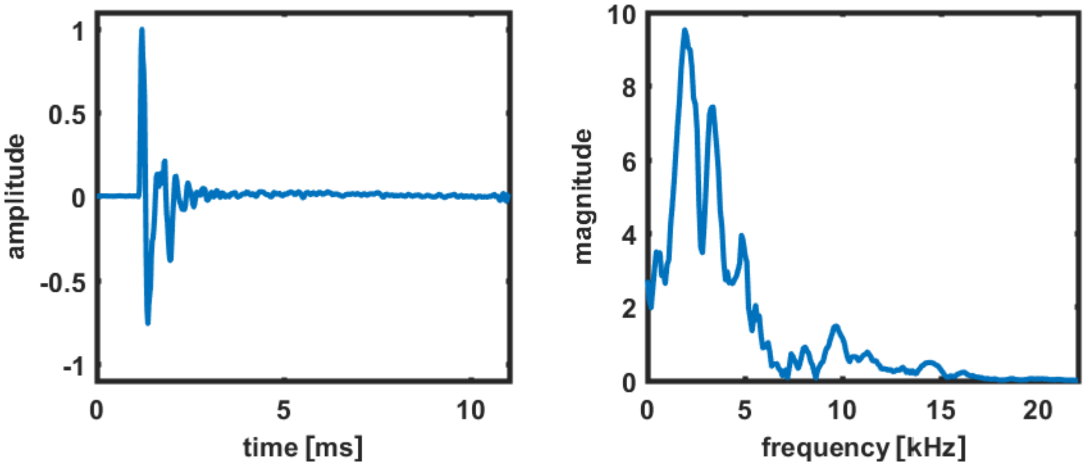

Fig. 45 Gunshot (waveform and spectrum) with no amplitude limiting

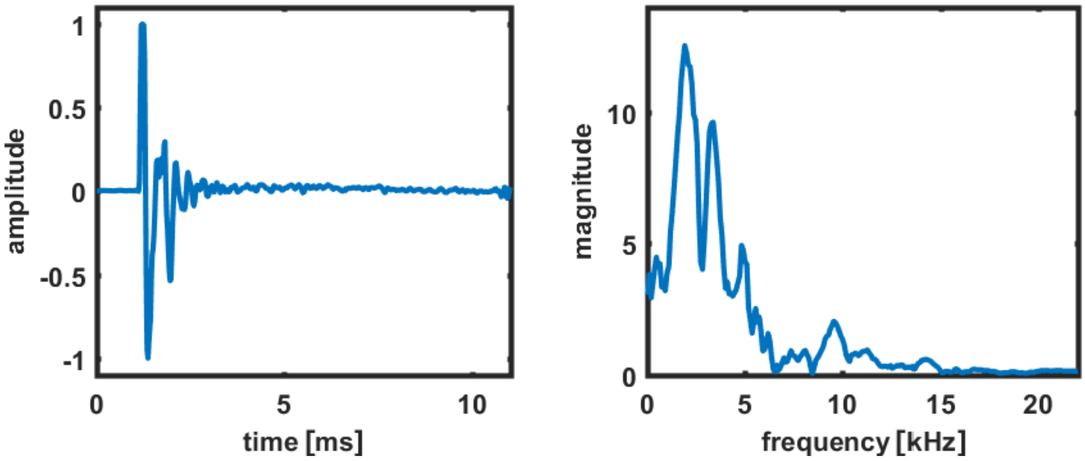

Fig. 46 Gunshot (waveform and spectrum) with light amplitude limiting

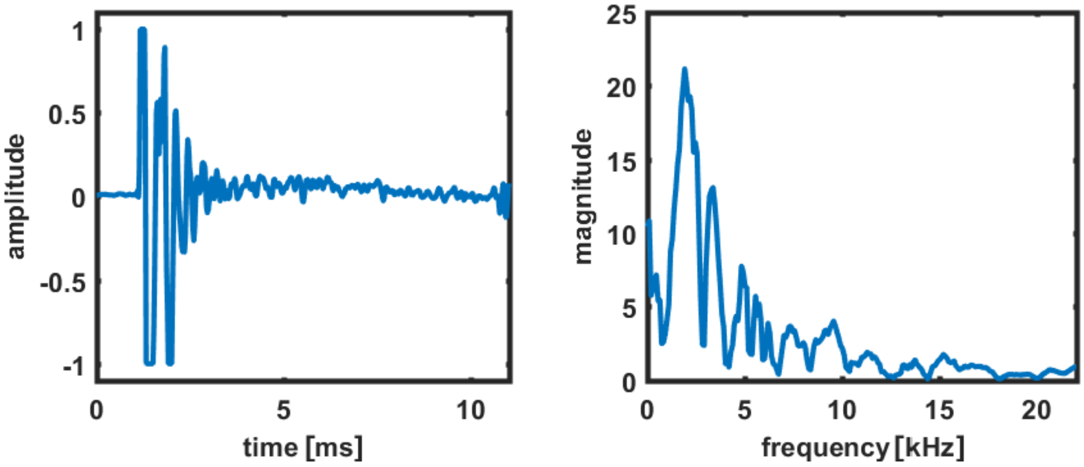

Fig. 47 Gunshot (waveform and spectrum) with heavy amplitude limiting

If no amplitude limiting is detected we proceed with the next steps. We check the frame for possible presence of single gunshots or gunshot bursts, using methods detailed below. If this preliminary test indicates presence of either, frames are saved for further processing and confirmation of true positives. Preliminary tests are also described in dedicated sections, further advanced processing is described in separate chapters. Signals B-D shown in Fig. 41. are saved in dedicated folders together with a timestamp for later processing/revision, an example of naming possible gunshots can be found below.

gunshots/22-May-2019-09-30-33-3033.wav

gunshots/22-May-2019-09-32-21-2731.wav

gunshots/23-May-2019-18-05-58-0025.wav

…

7.2 Preliminary Burst Detection

This section briefly describes processes and methods of preliminary burst detection. Advanced burst detection will be described in dedicated chapter.

After passing energy threshold check and amplitude limiting check, input frame is passed to preliminary burst detection block. Preliminary (online) burst detection uses center-clipping method (described in next subsection) to estimate whether input signal is periodic, and if so what the period is. The reason to pick center-clipping was mainly due to its low computational demands (in comparison with e.g. AMDF (Average Magnitude Difference Function), which will be described later) and sufficient performance in establishing basic frequency. This method uses center-clipping with reduction factor of 0.8 and alpha factor of 0.1, this setup ensures, periodic signal will not be lost in noise easily. In order for a frame to be flagged as a possible burst, it needs to have a period in range of +/- 5 ms from nominal weapon rate of fire (thus having range 85 ms – 105 ms for M45 and AK-47). If any frame conforms to these rather loose criteria, it is saved together with previous and the following frame for advanced (offline) processing, any adjacent frame conforming to these criteria is appended to the recording. Results for the first stage detection are presented below in Tab. 17. „Original duration“ shows duration of the whole category of sounds used in testing, „Stage 1“ column shows total duration flagged as bursts by preliminary burst detection in seconds, and also as a total percentage of original duration. All bursts in categories M45 and AK-47 pass this criterion under tested conditions.

Table 17 Preliminary burst detection results

Category	**Original duration**	**Stage 1 [seconds]**	**Stage 1 [%]**
Speech and music	11 hours	42 sec	0.11 %
Engine	1 hour 5 min	97 sec	2.49 %
Rain and thunderstorm	13 minutes	16 sec	2.05 %
Birds	35 minutes	21 sec	1.00 %
Dog	3 hours	74 sec	0.69 %

7.2.1 Center-clipping Method

The center-clipping algorithm is more suitable than AMDF (described later in chapter 8.1.1) to determine whether the given time frame is periodic, but it is not as good in determining the degree of periodicity. This algorithm works only with peaks (both positive and negative) and zeroes all values in between, zeroing threshold will be called clipping level [82]. In contrast to AMDF, this algorithm uses overlapping factor of 2/3.

Clipping level is determined by equation (25) as follows: input segments are subdivided into three frames (j-1, j, j+1) then peaks of left (MAX_{j-1}) and right frame

(*MAX*$_{j+1}$) are extracted. The clipping level CL is a product of the lower of these values and reduction factor r_f which was experimentally set to 0.8 for best performance [83].

$$C_{Lj} = r_f \cdot \min(MAX_{j-1};\ MAX_{j+1}), \tag{25}$$

after clipping, resulting samples are either rounded to +/-1 or zeroed as shown in Fig. 43 where input signal is in black, dashed pink line is clipping level and resulting clipped signal is in red. This clipped signal is used as an input for autocorrelation. Examples of autocorrelation function for periodic and non-periodic signals are shown in Fig. 49.

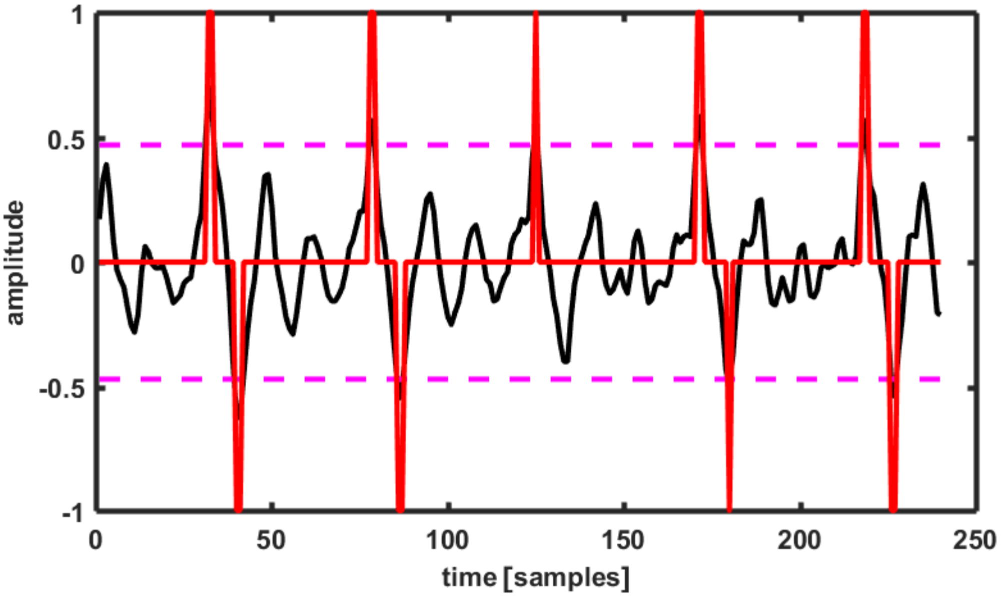

Fig. 48 Signal clipping

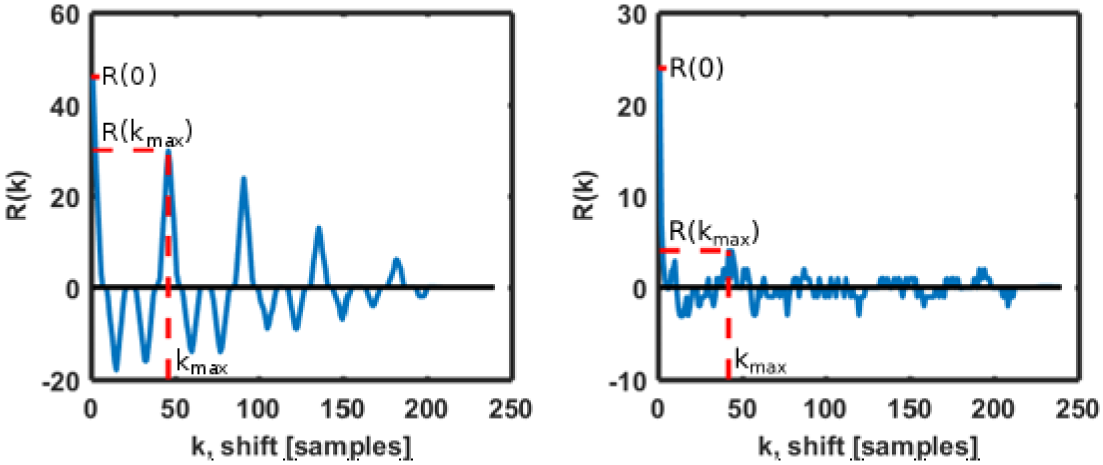

Fig. 49 Center-clipping autocorrelation output for periodic (left) and non-periodic (right) inputs

It can be seen, that periodic signal outputs distinctive peaks with decreasing amplitude at regular intervals. In contrast, non-periodic signal produces noise-like signal without any distinctive peaks or general trend.

After obtaining output from the autocorrelation function, the algorithm locates maximum of the function (apart from $R(0)$) and decides whether investigated frame is periodic according to the following criteria (26) and (27):

$$R(k_{max}) \leq \alpha \cdot R(0), \text{nonperiodic} \tag{26}$$

$$R(k_{max}) > \alpha \cdot R(0), \text{periodic} \tag{27}$$

where α is an empiric constant based on previous testing [83] and defaults to 0.3 for speech signal. In our application, we are using alpha factor equal to 0.1, since higher values resulted in too many missed detections. The period can be calculated by multiplying the position of maximum peak $k_{\max}$ by sampling period, the same way as in AMDF algorithm (30). However we cannot determine degree of periodicity (which can be done with AMDF) since energy normalization process is not as straightforward as in AMDF.

7.3 Preliminary Individual Gunshot Detection

This section describes detection of individual gunshots within bigger, 330 ms frames. Since individual gunshots (we are considering muzzle blast and disregarding acoustic shockwave, however the presence of shockwave is not a problem) without echo are very short, just several milliseconds, the whole frame will be divided into smaller subframes. Previous research [68] suggested 11 ms frame is sufficient for gunshot detection and results in performance comparable to 23 ms frames used previously.

Thus the next step is to divide 330 ms frame into 11 ms subframes, again without overlap. The reason we are not using overlap is because it introduces increased demands on computational power and our application presupposes presence of many gunshots, moreover the importance lies in high precision (i. e. low false alarm count), not on perfect recall. In the next step, energy check is performed again, in order to discard silent subframes, the threshold was set so that the most silent gunshots in our dataset will pass it. In this case, energy was calculated as a sum of squared samples and energy threshold under which no further calculation was done was set to 0.13.

Subsequently, we calculate features derived from 8th order MFCC, the concept was described more in detail in chapter 5.2, where comparison of various setups took place, but we will briefly mention the differences and any additional modifications. The calculation is basically identical, however these features are calculated on a linear frequency scale (as opposed to mel scale in MFCC), the bandwidth was 1 Hz – 8 kHz with 32 triangular filters, we will call these features LFCC (emphasizing linear frequency scale, bandwidth or filter shapes can and will vary). Additionally, before the calculation, the signal is upsampled to 192 kHz and further 10x in spectrum in order to increase the number of samples in each frequency bin. This LFCC set-up was proven to be slightly better than others (even compared to MFCC).

In the preliminary detection stage, neural networks were chosen as recognition algorithm due to its previous extensive use and good performance. Neural network training and testing was performed on a mix of data with gunshots from [20] and other sounds coming from Urban Audio dataset [4] and our recordings. We have used 7 non-gunshot classes (barking dog, drilling, jackhammer, siren, engine sound, sounds recorded near elephants – including trumpeting, and sound of rain and storm) and gunshot class.

For training, each class consisted of 900 feature vectors extracted from sound frames 11 ms long, randomly chosen from the above mentioned datasets.

Regarding architecture, in the first step, two approaches have been tested. First approach was training the network simply for gunshot detection, i. e. 2 class problem, gunshots vs. everything else. Second approach was to train the network for multiclass classification, where there was an output neuron for every non-gunshot class as well as for gunshot class. First approach yielded better results results mostly in terms of true positives for gunshot class, so we have subsequently decided to use the 2 class neural network. As for the number of hidden layers and neurons, grid search was used to determine best combination of hyperparameters. The grid search included options of 10, 20 and 30 hidden neurons in 1 or 2 layers, with both layers having the same amount of neurons. Finally, architecture with 2 hidden layers of 20 neurons each was chosen, resulting in 79% true positives and 86% true negatives over a dataset containing 1532 gunshots and 227923 non-gunshot frames from 4 different classes (barking dog, engine, raining and storm, speech and music).

Finally, if the network decides that a gunshot is present, the frame, along with previous and the following frame (and any adjacent flagged frames) is saved into dedicated folder with a timestamp for further processing as mentioned in the introductory chapter 7.1.

This preliminary approach was subsequently tested on a data consisting of classes „barking dog, engine, rain and storm, speech and music“. This monitoring yielded numerous non-gunshot sound frames labeled as gunshots. These, along with neighboring non-labeled frames (combined 31286 frames) were later used for testing in advanced gunshot detection, as described in chapter 8.2.

8 ADVANCED GUNSHOT DETECTION

This chapter explains how frames flagged as possible gunshots are processed to determine whether or not they really are gunshots. We will use neural networks for most of the testing, and in the end compare them to some other recognition algorithms to choose the best performing. Apart from describing the algorithm itself, this chapter will also describe new time-domain features we propose, which in this case exhibit great recognition performance for refining results obtained from preliminary gunshot detection.

8.1 New Features in Time Domain

This section proposes new features derived from signal waveform. Most features currently used in audio recognition are calculated in spectral domain (such as LPC or many MPEG-7 descriptors). Feature testing and reported performance in this section were performed on dataset described in [84].

This and the following paragraphs will describe calculation of 11 temporal features, with some illustrated by figures. First two features are relative positions of zero-crossings before and after the most dominant peak and third is their mutual distance (abbreviated RP-, RP+ and ZDist respectively), these are illustrated in Fig. 50 (shortening time axis for illustration purposes).

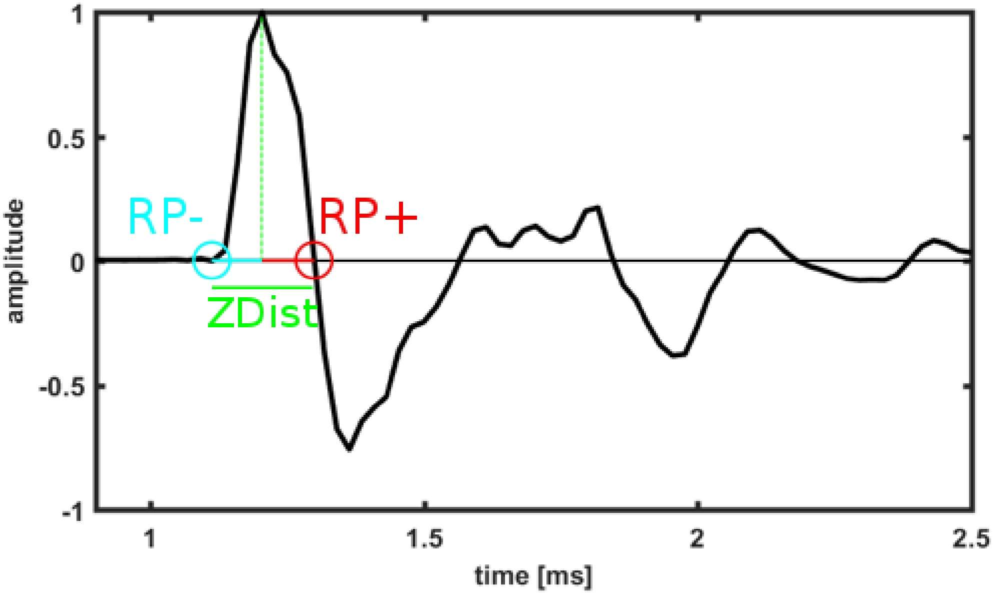

Fig. 50 Zero-crossings and their distance [84]

Other features include time distance between minimum and maximum values (PDist) and distance in two dimensions (PlDist), angle between the line connecting minimum and maximum and horizontal line (Ang) – the angle was calculated with horizontal line in seconds. Some of the features mentioned here are illustrated in Fig. 51.

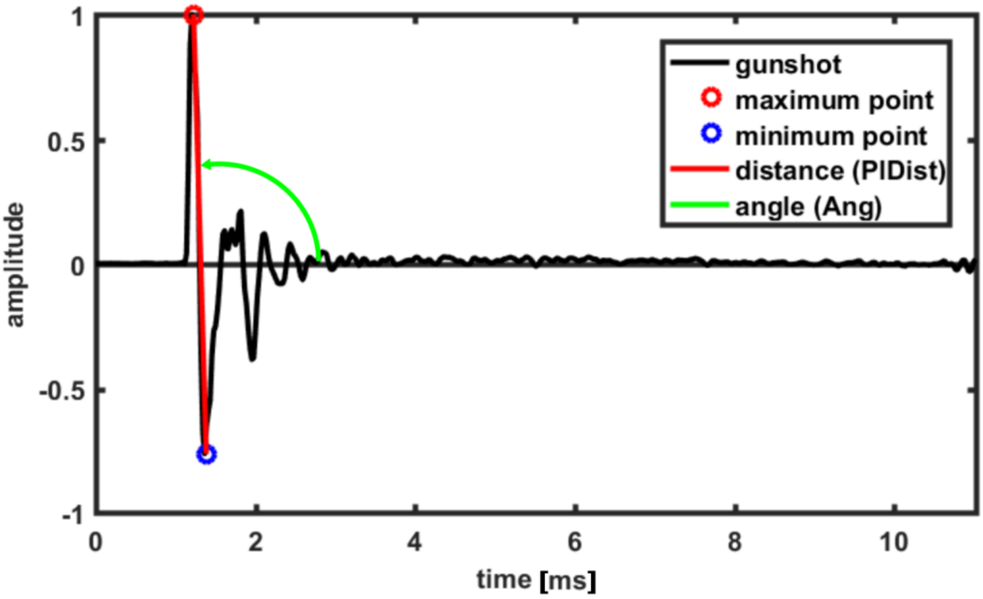

Fig. 51 Peak distance and angle [84]

The area of triangle delimited by 2 highest peaks and a minimum (referred to as "Area") is shown in Fig. 52.

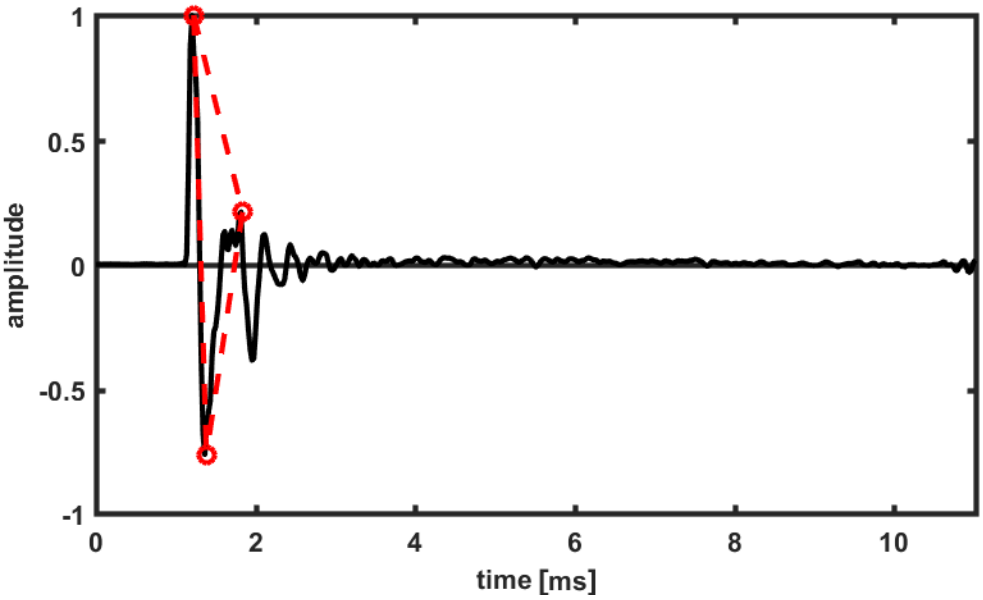

Fig. 52 Area feature illustrated [84]

Ultimately, 4 features were defined as coefficients (*A* and *B* in (28)) of exponential fit to both positive (abbreviated Ap, Bp) and negative (abbreviated An, Bn) local extremes.

$$y(t) = A \cdot \exp(B \cdot t), \tag{28}$$

where y(t) is exponential approximation, *A* and *B* are fitted coefficients, that serve also as recognition features and t is time. These features are illustrated in Fig. 53 together with approximations of positive envelope $p(t)$ and negative envelope $n(t)$ with numeric values for one sample waveform.

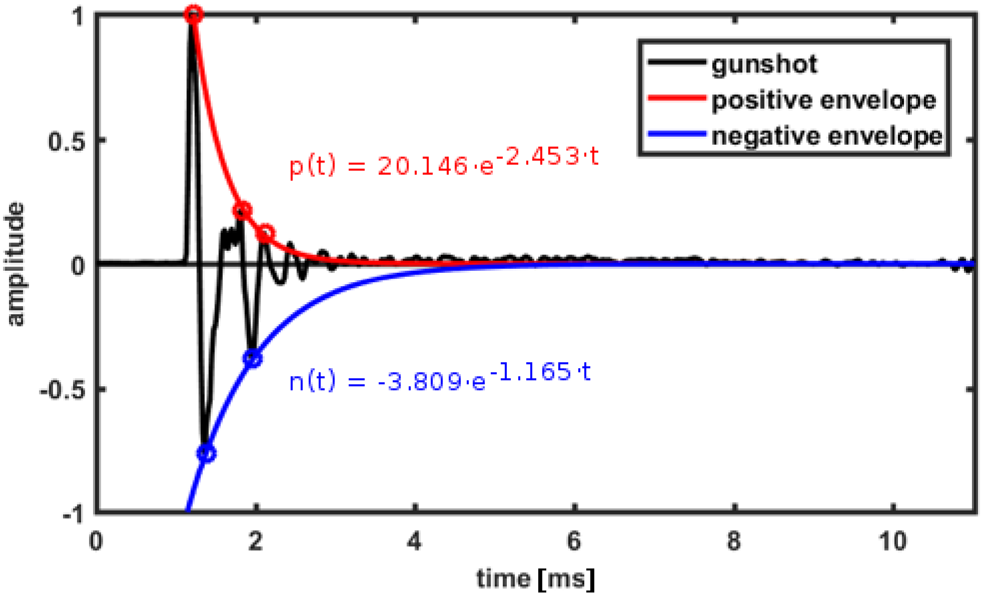

Fig. 53 Envelope approximation by exponential fit [84]

The viability of these features was tested by different means before actual usage so that we can tell which might be useful beforehand. Firstly a ratio between absolute mean value (μ) and standard deviation (σ) of each feature was calculated, with the expectation that higher value means that features will perform better. Next, we calculated mutual information between feature values and class labels using Matlab kerlenmi function (we disregarded mutual information between features themselves). Ultimately, two-sample t-test (using Matlab ttest2 function) was calculated measuring similarity of two distributions, where we compared distributions of gunshots and non-gunshots. We have used p-value of t-test (with 5% significance level, assuming unequal distribution variances), which should be lower for more dissimilar distributions, thus indicating better discrimination capability of a feature. Statistics for mean and standard deviation were calculated on all available data in all categories. Mutual information and p-values were calculated for no more than 2000 frames in each category due to memory restrictions during calculation. Ratio of mean to standard deviation indicated “Angle“ feature to be performing the best and *A*-coefficients of the fit of both negative and positive extremes the worst. Mutual information rating offers slightly different view, where “An“ feature is rated as the best, while “Area“ is the worst, with the rest of the features achieving similar scores. Lastly, p-value, ehre low values indicate dissimilar distributions indicated *B*-coefficients of the fit and RP+ with RP- are the best features. As further discussion reveals, we opted for precisely these features because of their superior performance. Tab. 18 summarizes the results for these different tests.

Table 18 Recognition rating estimate for TDF

Feature	gunshots			non-gunshots			all	
	μ	σ	|μ|/ σ	μ	σ	|μ|/ σ	Mutual Information	p-value
RP-	-135.17	131.55	1.03	-815.99	1311.42	0.62	0.47	2.817E-182
RP+	209.28	585.47	0.36	-431.13	3465.14	0.12	0.43	2.279E-43
Zdist	344.44	604.17	0.57	384.86	3278.48	0.12	0.50	3.776E-03
Pdist	327.02	1368.28	0.24	-17.65	4822.17	0.00	0.43	3.138E-09
PlDist	819.79	1143.47	0.72	3958.53	2753.80	1.44	0.41	0.000E+00
Angle	74.40	34.17	2.18	89.55	78.15	1.15	0.61	9.556E-35
Area	0.98	0.88	1.11	0.82	1.03	0.80	0.33	1.681E-01
An	-3.23	62.46	0.05	-10.38	766.72	0.01	0.68	1.933E-01
Ap	5.28	93.65	0.06	20.10	1081.58	0.02	0.77	2.265E-01
Bn	-0.35	0.42	0.82	-0.06	0.64	0.09	0.53	4.600E-136
Bp	-0.58	0.73	0.79	-0.06	0.62	0.10	0.62	1.514E-142

Actual recognition performance was tested using progressively increasing number of these features (firstly ordered by above mentioned criteria) with implementation of Matlab neural networks (10 neurons in 1 hidden layer, sigmoid activation function). This configuration did not perform very well for lower number of features, which prompted us for reordering. After few trials, we settled on six to seven features: RP+, RP-, Bn, Bp, PlDist, Angle and ZDist. Performance for increasing number of TDF is illustrated in Fig. 54, Table 19 indicates recognition performance for problems „all gunshots vs. non-gunshots“ and „AK-47 vs. non-gunshots“.Overall, we conclude that despite the fact, that some other features (such as LPC or MFCC) achieve slightly better results, our features are comparable and are an excellent addition to some more frequently used features, especially due to their temporal origin which might hint low mutual information with spectral features. Their viability will be explored and confirmed in the subsequent chapter.

Table 19 Best performance of temporal features

Subset	Recall	Precision	F-Score
AK-47	80.8 %	38.1 %	51.8
All gunshots	82.2 %	69.3 %	75.2

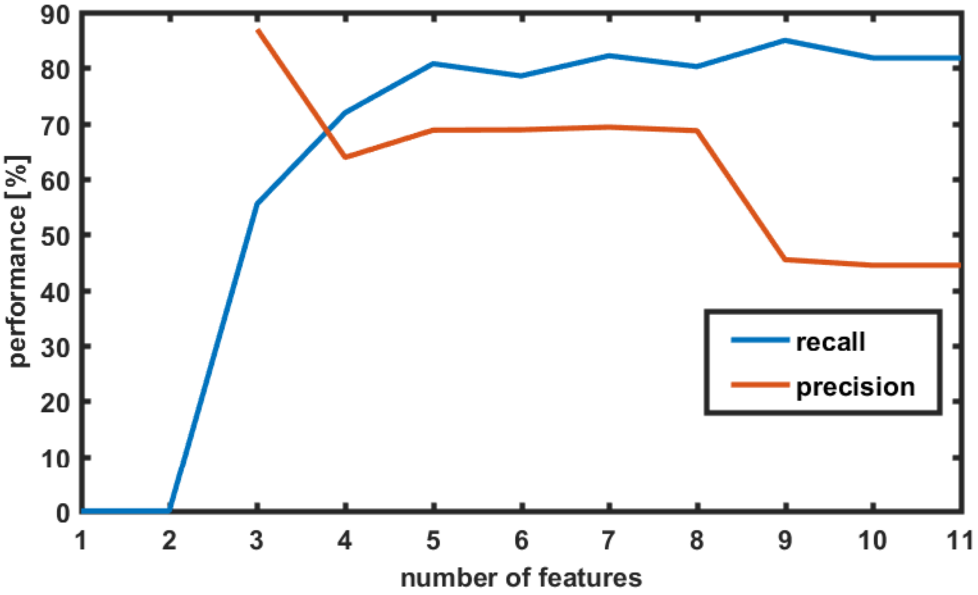

Fig. 54 Gunshot recognition performance for different number of temporal features [84]

8.2 Advanced Gunshot Detection Results

Various ideas for advanced gunshot detection were considered. One such example of state-of-the-art approach [60] uses convolutional/recurrent networks with mel spectrogram (with 40 frequency bins) over multiple time frames (1024 time frames of 40 ms each with 50% overlap) for multiclass sound event detection (including gunshots). This approach was tested, training the network using our dataset. The results on gunshot recognition could not be reproduced, and were not satisfactory. For this reason, and because of long training times, we did not consider using this architecture afterwards. Instead, we turned to MFCC once again in order to leverage its variability described in chapter 5.2. In order to limit mutual information between this stage and stage 1 recognition, many parameters were changed. These features were calculated on a mel frequency scale, using different feature order and more filter-banks and also different filter shape (gammatone) compared to preliminary detection approach, so mutual information should be limited. Individually, triangular filter banks calculated on mel frequency scale performed better, but later experiments turned out in favor of gammatone filter banks on a mel scale.

The dataset for this chapter came from 2 different testing rounds of stage 1 (preliminary) gunshot recognition. First set - set A - was obtained from testing data described in Tab. 17 and contains both, frames labeled as gunshots (approx.. 20% of set A) and frames not labeled as gunshot (approx.. 80% of dataset) by the preliminary stage. The set consisted of 59723 frames unevenly distributed in 4 classes (i.e. dogs, engine, rain, speech and music), plus 1532 gunshot sounds from [20]. Set A was further subdivided into training subset (60% of the dataset), validation subset (20%) and testing subset (20%). In algorithms that did not use validation subset, this was joined with training subset into single (80%) training subset. All subsequently mentioned training was performed exclusively using this dataset along with most of the testing (with exceptions explicitly mentioned later), which served as a basis for selecting suitable algorithms. The reported numbers thus reflect performance of described algorithms only, disregarding information from stage 1 (apart from 20%-80% distribution of flagged vs. non-flagged frames). Due to relatively low number (11944) of frames flagged as gunshots in this dataset by the first stage, and thus possibly distorted testing results, we are including an additional dataset. The dataset B is 100% testing dataset and consists exclusively of frames flagged as gunshots in stage 1 (32818 frames) in categories barking dog, engine, speech and music, public places (including distant chatting crowds, traffic soundsand construction sounds) and gunshots. Overlap in dog and engine classes should be limited to minimum, since both dataset randomly drew from more extensive pool of sounds. Sounds in gunshot category consist of dataset [20] and thus are identical with dataset A. There is no overlap in category "speech and music" in sets A (using czech speech and international music of various genres) and B (using slovak speech and international music of various genres). Public places category in dataset B has its origins in [13], since no data from this source were used in dataset A, algorithms did not see any part of the data, and so there is no overlap in this category in datasets A and B. Results from dataset B are reported separately at the end of the chapter and express overall recognition performance with information combined from stages 1 (online/preliminary) and 2 (offline/advanced), as further clarified later.

Training single feedforward network for one vs. all or multiclass recognition was an approach employed in preliminary gunshot detection and also provided unsatisfactory performance for second stage. Here, we aimed for a more complex and better performing approach, thus we have reached to explore ensembling methods. We have trained one network for each non-gunshot category separately, so that we have 4 networks, each distinguishing gunshot from different non-gunshot sound. We have been using training subset of set A for all networks, while picking only sound classes of interest. Additionally, in pursuit of other, uncorrelated features, we turned to features developed by us, and described in [84] and also in chapter 8.1. We have used 5 best features from the list (according to metrics and further tsting described in [84]). Namely exponent of approximation of negative (Bn) and positive (Bp) waveform envelope, relative positions of first zero-crossings before (RP-) and after (RP+) the dominant peak and distance between global minimum and maximum (PlDist), the rest of the features mentioned in chapter 8.1 did not provide further improvement in recognition so we did not use those. In order to see numeric values for a single gunshot, we present Fig. 55 which shows gunshot waveform used and Tab. 20 with values of every feature (MFCC+TDF).

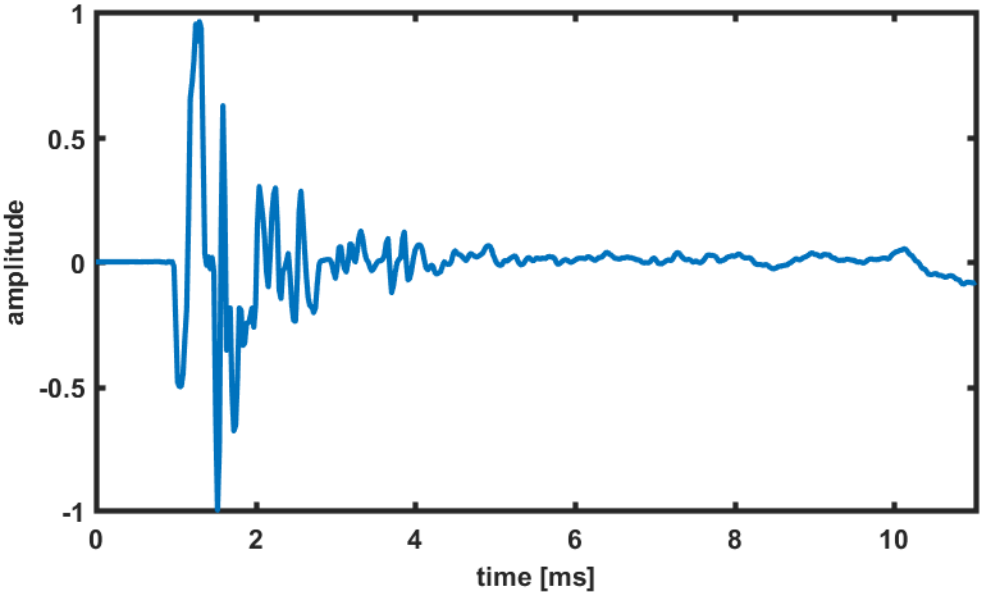

Fig. 55 AK-47 gunshot waveform - input to feature extraction

Table 20 Features extracted from gunshot in Fig. 55

MFCC 1	MFCC 2	MFCC 3	MFCC 4	MFCC 5
112.16	-10.70	-11.60	-8.32	-1.88
MFCC 6	**MFCC 7**	**MFCC 8**	**MFCC 9**	**MFCC 10**
-3.13	2.55	0.93	-10.72	-1.93
MFCC 11	**MFCC 12**	**MFCC 13**	**MFCC 14**	**MFCC 15**
3.70	-1.70	4.36	4.59	2.71
MFCC 16	**MFCC 17**	**MFCC 18**	**MFCC 19**	**MFCC 20**
5.29	-2.51	1.33	-1.57	-1.46
RP-	**RP+**	**PlDist**	**Bn**	**Bp**
-136.05	90.70	231.08	-0.79	-1.17

The results for network with single hidden layer with 30 neurons are summarized in Tab. 21, table is divided into part where only MFCC were used, combination of MFCC and 5 time domain features (TDF) named in previous paragraph, and 5 TDF only. Comparison with results from other architectures and afterwards also other algorithms is provided later. Later comparison shows both performance differences (in terms of true positives and true negatives) and execution time differences. As will be seen, some algorithms provide clearly better results, while some allow for a compromise regarding a ratio of true positives and true negatives.

Table 21 Recognition results using single neural network trained per category

Category	# frames	MFCC only	MFCC+TDF	TDF
Speech & music	2301	98.7%	99.3%	99.6%
Engine	7708	99.6%	99.8%	99.7%
Barking dog	1834	98.4%	89.4%	90.5%
Rain/storm	102	82.9%	96.2%	89.5%

As can be seen, results for combined features are generally better in comparison with other feature setups, except for the "dog" class, whose performance is much poorer. One problem with this approach (separately training networks for each category) in real-life conditions is that we cannot tell beforehand which network to use. Inspiration was drawn from ensemble learning, where mutltiple recognition algorithms are run in parallel and these vote for the result. Thus, we run all networks and then sum probabilities of resulting classes (gunshot / non-gunshot) across networks and then decide.

Tab. 22 compares results obtained over all classes using ensemble method (i.e. using each network and then sum probabilities), comparing also different network architectures. The name of the network "NN*xy*" means single layer network with *xy* neurons, the name "NN*xy*+*xy*" stands for two hidden layers, each with *xy* neurons. Apart from neural networks, we have also tried other recognition algorithms, compared to selected neural network (NN20+20) in Tab. 23. NN20+20 was chosen because it offers TNR comparable to best architecture (NN30 in this case), but vastly superior TPR. Other tested algorithms, along with brief description of their hyperparameters, are listed below. Chapter 6 on recognition algorithms provides more detailed look into each algorithm. The compared algorithms include the following, Support vector machines (SVM) with Gaussian kernel, this kernel was chosen because it achieved superior results in [66]. Another algorithm, k-nearest neighbors (kNN) is using Euclidean distance (for standardized features) and k = 5 nearest neighbors (achieving comparable or better results than using different values during optimization step). Decision tree (tree) with minimum leaf size equal to 1 (i.e. number of samples in one leaf node), maximum number of splits equal to number of samples minus one, and using "Gini diversity index" as a split criterion. This set-up for decision trees achieved the most desirable results (in terms of true negatives) during the optimization stage. And lastly Naïve Bayes classifier, where we are presupposing normal distribution for each feature. Along with these results, Ensemble result is presented, which was obtained as summing decisions (not probabilities) of all classifiers in the table and choosing the most frequent class. For example, if SVM, kNN and neural networks decide the event is gunshot and decision trees and Naïve Bayes say it is not a gunshot, overall decision is gunshot, because 3 algorithms vote for gunshot while only 2 vote for non-gunshot. True negative rate (TNR) is defined as a ratio of correctly rejected non-gunshot sounds and true positive rate (TPR) which is defined as a ratio of correctly detected gunshots. In Tab. 22 and Tab. 23, green color shows best results achieved using TPR as primary metric and highlights also corresponding TNR, orange results highlight the best TNR result plus corresponding TPR result.

Training of all algorithms below was performed with training subset of set A. Training was performed with 80% of the data (in neural networks, 60% for training and 20% for validation) and testing with 20% of the data, chosen in non-overlapping blocks.

Results reported in tables below come from testing data only. Overall non-gunshot contained 59723 frames, gunshot category contains 1532 shots from different weapons (including assault rifles, hunting weapons and handguns). Inputs to all algorithms are standardized, so that mean value is 0 and standard deviation is 1 using data from the testing set. This is done with all algorithms except for decision trees, which do not need such treatment ensuring equal scale.

Table 22 Performance for different neural network architectures and features

True Negative Rate

Features	**NN10**	**NN20**	**NN30**	**NN10+10**	**NN20+20**	**NN30+30**
MFCC	72.1%	76.3%	76.2%	79.3%	76.4%	75.8%
MFCC+TDF	92.2%	90.7%	90.0%	90.3%	89.9%	**91.7%**
TDF	97.5%	97.7%	**98.0%**	97.3%	97.4%	97.3%

True Positive Rate

Features	**NN10**	**NN20**	**NN30**	**NN10+10**	**NN20+20**	**NN30+30**
MFCC	88.6%	88.6%	90.2%	87.9%	89.5%	85.0%
MFCC+TDF	92.2%	93.5%	94.4%	92.2%	92.8%	**95.1%**
TDF	85.0%	80.7%	**73.5%**	83.7%	86.0%	86.0%

Table 23 Performance for different classification algorithms and features

True Negative Rate

Features	**SVM**	**kNN**	**NN20+20**	**Tree**	**Naïve Bayes**	**Ensemble**
MFCC	**100.0%**	82.9%	76.4%	72.9%	81.3%	85.0%
MFCC+TDF	98.2%	90.5%	89.9%	89.9%	**85.0%**	92.7%
TDF	97.2%	96.9%	97.4%	98.3%	95.7%	97.2%

True Positive Rate

Features	**SVM**	**kNN**	**NN20+20**	**Tree**	**Naïve Bayes**	**Ensemble**
MFCC	**16.3%**	88.9%	89.5%	85.3%	69.3%	88.6%
MFCC+TDF	9.5%	93.8%	92.8%	95.8%	**96.7%**	94.8%
TDF	86.3%	89.5%	86.0%	87.6%	91.2%	87.3%

As can be seen from tables above, from the point of view of least false alarms, SVM perform the best, however they have also prohibitively low true positive rate. From the point of view of best true positive rate, Naïve Bayes classifier performs the best. In order to choose a compromise, with focus on less false alarms, we have chosen decision tree algorithm with TDF only features, which provides excellent true negative rate (98.3%), while achieving very good true positive rate (87.6%). Additionally, decision tree algorithm also performs the best in terms of execution time, as will be shown later.

Apart from overall results of true negatives and true positives, we will look at true negative rates by category, i.e. what class is the most problematic overall. Tab. 24 below is similar to Tab. 21 in the fact that it offers true negatives per category. The difference is that Tab. 21 provides results for individual algorithms (in this case neural networks) and Tab. 24 provides results for whole ensemble (using decision tree algorithm), number of sound frames in each category is the same in both tables.

Table 24 True negative rates for decision tree ensemble – breakdown by category

Category	# frames	MFCC	MFCC+TDF	TDF
Speech & music	2301	88.8 %	96.2 %	98.6 %
Engine	7708	63.2 %	87.5 %	98.2 %
Barking dog	1834	87.7 %	91.1 %	98.4 %
Rain/storm	102	84.8 %	100.0 %	99.1 %

From Tab. 23 can be seen, that from the point of view of maximal true positives, combination of MFCC and TDF is the best. To further analyze contribution of individual TDFs, we took MFCC features, and added TDFs one by one and examined how true positives changed with various individual TDFs. Tab. 25 summarizes the results for decision tree algorithm.

Table 25 Contribution of individual time domain features

Metric	MFCC only	MFCC + RP-	MFCC + RP+	MFCC + PlDist	MFCC + Bn	MFCC + Bp
True positives	85.3%	92.2%	94.8%	85.0%	86.0%	89.2%
True negatives	72.9%	93.4%	90.7%	73.4%	76.7%	82.5%

Tab. 25 shows that, each TDF contributes to better recognition performance in at least one aspect (true positives / true negatives). Looking at individual contributions, we establish that from TDF from best to worst are RP- > RP+ > Bp > Bn > PlDist. Now to figure out incremental contribution of TDF to recognition performance with MFCC, we are going to add time domain features one by one to MFCC in order of their contribution from best to worst (specified above). Tab. 26 shows results of this incremental contribution.

Table 26 Incremental contribution of time domain features

Metric	MFCC only	MFCC + 1 TDF	MFCC + 2 TDF	MFCC + 3 TDF	MFCC + 4 TDF	MFCC + 5 TDF
True positives	85.30%	92.22%	89.97%	90.41%	89.91%	89.94%
True negatives	72.90%	93.40%	94.44%	95.10%	95.42%	95.75%

The biggest incremental increase in performance happens with 1 or 2 TDF, extra 3 TDF increase performance only slightly, and so, it is up to application requirements to determine whether adding the features is worth the increase in computational demands or not. However when using TDF only as features, chapter 8.1 concludes that using more than 3 features is advisable, since this significantly increases TPR.

In order to compare computational demands of different algorithms, we have run each algorithm five times and averaged the execution time. Each time, we input 59723 feature vectors (i.e. 59723 different, 11 ms long recordings converted to features). The algorithm was run on a desktop running Windows 7 with 8 GB RAM and Intel Core2 QUAD Q9650 processor without graphic card acceleration. Tab. 27 and Tab. 28 show, analogously to Tab. 22 and Tab. 23, execution times for different neural network architectures and for different algorithms respectively. Only execution time (in seconds) of algorithms is included, features were calculated separately.

Table 27 Execution times in seconds of various neural network architectures

Features	**NN10**	**NN20**	**NN30**	**NN10+10**	**NN20+20**	**NN30+30**
MFCC	0.37	0.44	0.51	0.45	0.56	0.79
MFCC+TDF	0.53	0.56	0.63	0.54	0.74	0.94
TDF	0.25	0.29	0.34	0.31	0.44	0.67

Table 28 Execution times in seconds of various recognition algorithms

Features	**NN20+20**	**SVM**	**kNN**	**Tree**	**Naïve Bayes**	**Ensemble**
MFCC	0.56	36.66	20.78	0.18	0.30	62.94
MFCC+TDF	0.74	46.16	24.68	0.19	0.40	72.25
TDF	0.44	1.27	1.48	0.11	0.14	3.48

As could be expexted in Tab. 27, more neurons meant longer execution time. This includes both input neurons (i.e. number of input features) and neurons in hidden layers. Tab. 28 is more interesting. As for input features, less features mean shorter execution time, again. However various algorithms perform very differently, with decision trees being the quickest and SVMs the slowest by a wide margin. Neural networks, have execution times only slightly worse than decision trees, and so are very good choice from execution time point of view as well.

Apart from execution time of algorithms themselves, we should also compare execution times of feature extraction algorithms, since they are slower than actual recognition algorithms, we only calculated features of 1532 recordings (each 11 ms long) and compared their execution times. Among compared features are MFCC coefficients of order 20, upsampled MFCC coefficients (used in preliminary gunshot detection) of order 20, 5 time domain features (TDF) described in previous section and LPC coefficients of order 20, Tab. 29 summarizes those results in seconds. Each extraction algorithm was calculated 5 times again and the time was averaged.

Table 29 Execution times in seconds of feature extraction algorithms

MFCC	MFCC-upsampled	LPC	TDF5
1.07	16.22	0.17	110.213

LPC coefficients, as native Matlab algorithm achieve the best performance in terms of execution time. MFCC algorithm from Matlab file exchange fares order of magnitude worse, while upsampling adds considerable amount of time to the calculation. TDF execution time is by far the longest, which also makes it unsuitable for real-time deployment in its current implementation. One explanation for such a long time of execution is, that the algorithm is in its first version and no optimization was done. However excellent recognition performance of TDF make it a great algorithm to be employed for offline, advanced analysis. Since TDF as described in previous chapter actually consists of 11 different features, it is to be expected that different features have different calculation complexity. Tab. 30 below breaks TDF down into single features or groups of features to compare their individual calculation times. Some features are in groups, because their prerequisites are calculated together with other features (such as ZDist being difference of RP+ and RP-) or are output of the same function (such as Ap and Bp). TDF5 represents 5 TDF selected in this chapter, TDF11 presents a complete set of all TDFs presented in chapter 8.1.

Table 30 Calculation times of various TDF

RP-	RP+	RP- RP+	RP- RP+ Zdist	PDist	PlDist	PDist PlDist
0.019	0.020	0.023	0.023	0.023	0.025	0.025

Ang	PDist PlDist Ang	Area	Ap, Bp	An, Bn	Ap, Bp An, Bn	TDF11
0.025	0.026	1.285	56.687	55.459	110.115	110.958

It can be seen, that there are big differences in calculation times, while most of the features take very little time (comparable to LPC in previous table) to calculate, Area feature is around two orders of magnitude more complex. Exponential fit features (i.e. Ap, Bp, An, Bn) are even much more demanding, taking up the majority of time needed to calculate all features. When we look at Tab. 26, we can see that if we are going to use combination of TDF and MFCC, we can easily do without exponential fit features without sacrificing much recognition performance. However if we are using only TDFs, as is the case because of the bigger benefits to true negative rate, we have to be using more than 3 features, since the analysis in chapter 8.1 shows there are still big benefits to adding more features and namely Bn and Bp.

Thus, the final algorithm for advanced gunshot detection based on dataset A is a decision tree with hyperparameters mentioned by the beginning of this chapter. The final performance will be now tested on dataset B to provide more unbiased results. Tab. 31 presents results in terms of true negatives. With dataset B consisting of false alarms after stage 1 (31286 frames), we also provide the proportion of original data before stage 1 in "Total frames" column. The most interesting part consists of number and percentage of true negatives in dataset B from stage 2 and the total proportion of false alarms in the original pool of recordings from which dataset B was compiled. Tab. 32 provides information on true positives in a similar manner.

Table 31 Evaluation of overal results (True Negatives) of gunshot detection on dataset B

Category	**Total frames [# frames]**	**Stage 1 - TN [# frames]**	**Stage 1 - FA [# frames]**	**Stage 1 - TN [%]**	**Stage 2 - TN [# frames]**	**Stage 2 - TN [%]**	**Overall - TN [%]**
Dog	55389	46412	8977	83.79%	7938	88.43%	98.12%
Engine	23422	8085	15337	34.52%	14982	97.69%	98.48%
Public places	69440	66570	2870	95.87%	2592	90.31%	99.60%
Speech & music	53591	49489	4102	92.35%	3884	94.69%	99.59%
Combined	201842	170556	31286	84.50%	29396	93.96%	99.06%

Table 32 Evaluation of overal results (True Positives) of gunshot detection on dataset B

Category	**Total frames [# frames]**	**Stage 1 - TP [# frames]**	**Stage 1 - TP [%]**	**Stage 2 - TP [# frames]**	**Stage 2 - TP [%]**	**Overall - TP [%]**
Gunshots	1532	1207	78.79%	1158	95.94%	75.59%

The whole system, including stage 1 and stage 2 thus achieves TNR of over 99% for 4 combined non-gunshot categories and over 75% TPR for 1532 gunshots from various types of weapons, including handguns and assault rifles.

9 ADVANCED BURST DETECTION

This chapter describes advanced processing employed on audio frames flagged and saved as possible gunshot bursts by preliminary algorithm described in chapter 7.2. The main focus of the chapter is to examine period and periodicity of input audio waveform. This approach is further improved by addition of gunshot detection on top of which we examine periodicity. First section in this chapter introduces process of feature extraction and evaluates proposed features. Remaining two sections describe two proposed versions of algorithm, compare them and propose final solution.

9.1 Burst Features

The most salient feature of gunshot bursts is its periodicity. Thus, we have focused on estimating average period of the burst, detailed period of each gunshot in a burst along with differences between adjacent periods (referred to as delta-period) and time difference between first and last period (referred to as first-delta-period). We have also compared degree of periodicity (i.e. how similar individual periods are) regarding adjacent periods (referred to as periodicity) and first and last period again (referred to as first-periodicity). Methods employed include Average Magnitude Difference Function (AMDF), center-clipping and peak-search, algorithms not yet described will be described in the following chapter.

9.1.1 AMDF Method

The Average Magnitude Difference Function (AMDF) calculates *D*(*k*) curve, which is based on modified short-term autocorrelation function, namely it uses absolute value of difference instead of multiplication, as shown in (29)

$$D(k) = \sum_{n=1}^{N-k} |s(n) - s(n+k)|, \tag{29}$$

where $s(n)$ are signal samples, *k*=(0,1,....*N*-1) is time shift, and *N* is frame length (in samples). The function is calculated for all frames. *D*(*k*) curve is afterwards normalized by division with R - regularization term corresponding to signal energy (30) so that values are in range 0-1, with zero representing perfectly periodic signal. An example of output before and after normalization can be seen in Fig. 57 with Fig. 56 being the input signal, the important thing is, that only Y-axis is scaled, without effect on X-axis. This way, different signals can be compared to each other. Typically, the envelope has decreasing tendency due to lowering number of samples in summation. However, we are using overflow to adjacent segments (if there are no adjacent segments, we just circularly shift the frame) and so function output looks more monotonous, such as the one depicted in Fig. 58, apparent period of around 50 samples. This approach is advantageous in that we are avoiding meaningless minima which usually appear for higher shift values. The first significant minimum (outside of zero-region where time shift is near 0) represents the periodicity degree and basic period of investigated frame.

$$R = \sum_{n=1}^{N} 2 \cdot |s(n)|. \tag{30}$$

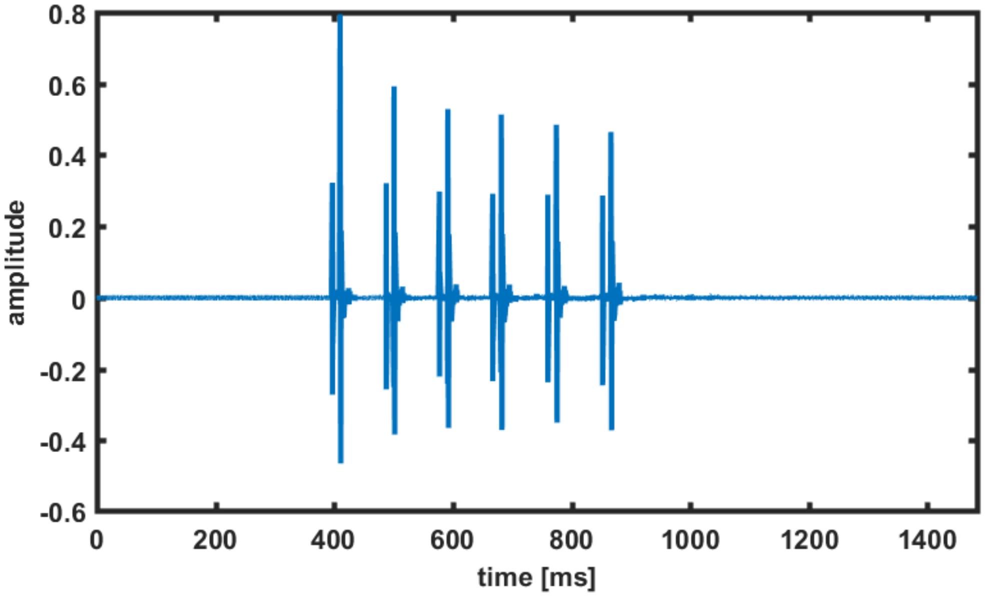

Fig. 56 AK-47 gunshot burst - input to AMDF

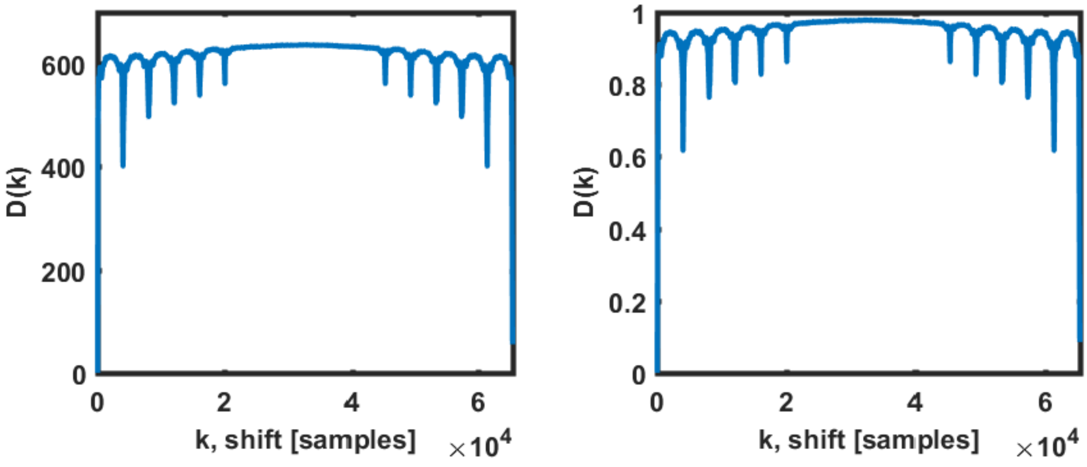

Fig. 57 AMDF output - without normalization (left) and with normalization (right)

The basic period in seconds can be calculated as follows:

$$T_0 = k_{\min} \cdot T_S, \tag{31}$$

where $k_{\min}$ stands for location of the first significant minimum in the $D(k)$-wave (horizontal coordinate) and T_s is sampling period. Moreover, the non-zero value of $D(k_{\min})$ (vertical coordinate of the first significant minimum) effectively represents degree of non-periodicity in the signal waveform. For a truly periodic (having constant period and wave shape identical in all periods) signal $s(n)$ becomes $D(k_{\min}) = 0$. The AMDF algorithm was previously applied for successful selection of multiple periodic speech segments in [85].

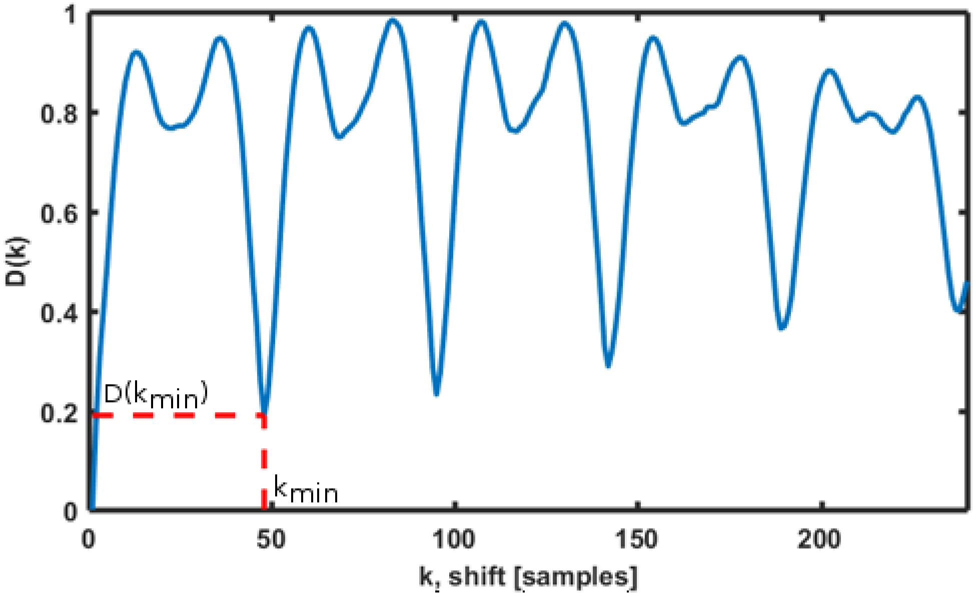

Fig. 58 Similarity function *D*(*k*)

9.1.2 Feature Statistics

In order to estimate period and delta-periods, we employed center-clipping with peak-search. Peak-search consists in finding peak positions placed approximately period-length apart, with tolerance of 10% (with the initial period estimate coming from center-clipping algorithm described in previous chapter). Periodicity was estimated using AMDF method with adjacent pairs of gunshots (from single burst) as an input. These statistics were estimated on clean gunshot bursts without added noise, they are presented below, in Tab. 33, with mean values, minima, maxima and their differences.

Table 33 Statistics of AK-47 bursts

Feature	Unit	Mean	Min	Max	Max-Min
Period - peak-search	[ms]	91.01	85.20	99.15	13.95
Delta-period	[ms]	-0.15	0.02	8.16	8.14
First-delta-period	[ms]	-1.22	0.03	9.91	9.88
Periodicity	[-]	0.46	0.21	0.78	0.57
First-periodicity	[-]	0.53	0.34	0.80	0.46

As can be seen, individual periods vary significantly (around 10%). However, with detailed look on all bursts, mean period within each burst varied only slightly (approx. 2%). On the other hand, periodicity took on a wider range of values, which were also overlapped with other, non-burst sounds, thus, we do not consider periodicity as a suitable feature later.

Next, we wanted to compare peak-search method to AMDF under clean and noisy conditions. We considered multiple noises, both stationary (awgn, engine, rain) and impulsive (barking dog, cracking branches). In order to estimate length of all periods using AMDF, we did not consider only first minimum, but all minima in similarity function (Fig. 58). Tab. 34 presents period statistics under different noise conditions.

Table 34 Period length using peak-search and AMDF with gunshot signal degraded using different levels of various non-gunshot signals

Noise type	**Peak-search**		**AMDF**	
	20 dB	0 dB	20 dB	0 dB
AWGN	13.95	13.83	15.58	12.43
Rain	12.70	12.89	9.93	13.24
Engine	13.95	13.73	15.99	139.16
Dog	13.95	13.95	9.91	298.96
Cracking branch	12.32	13.42	11.63	226.83
Without noise	13.95		9.93	

Peak-search apparently performs well even in noisy conditions, but period localization (i.e. reported start and end of periods) reports a lot of incorrect positions, thus its reliability in stationary noise conditions is misleading, solution to this problem would be to pick an algorithm according to long-term noise conditions evaluated on a different basis.

9.2 Advanced Burst Detection Results

This section compares two different approaches to burst detection using previously introduced features. The first approach being AMDF and the second peak-search algorithm (which uses center-clipping).

The first approach consists of detailed look into signal periods directly from input audio waveform. In order to establish whether frames flagged by preliminary detection really are bursts, we examine their periods in detail. In order to do this, we use both previously described methods (AMDF and peak-search), note that both methods are employed on whole recordings (with any appended frames). As stated previously, the mean period of gunshot bursts have, under tested conditions, very small deviation values. This feature was selected as a criterion to establish whether recording really is a burst, the criterion was that mean period must be nominal weapon rate of fire +/- 3 ms. In contrast with preliminary approach, this method takes into account each individual period in recording and achieves ore precise period measurements.

Results in form of false alarm counts are summarized in Tab. 35 below. The table is divided into different categories of non-gunshot sounds, it includes original input audio duration, total duration flagged as bursts in stage 1 (online recognition) and proportion of recordings flagged as bursts in stage 2 (offline duration) to those flagged as bursts in stage 1. Tab. 36 presents true positives for the two methods, separately for each weapon, showing total number of bursts per weapon and number of bursts succesfully recognized as bursts.

Table 35 Burst detection results – False alarms

Category	Original duration	Stage 1	Stage 2: AMDF [flagged recording /all]	Stage 2: peak-search [flagged recording/all]
Speech and music	11 hours	42 sec	11/126	46/126
Engine	1 hour 5 min	97 sec	54/224	43/224
Rain and thunderstorm	13 minutes	16 sec	2/16	2/16
Birds	35 minutes	21 sec	22/46	22/46
Dog	3 hours	74 sec	13/65	0/65

Table 36 Burst detection results – True positives

Weapon	Burst count	Stage 2: AMDF [true positives]	Stage 2: peak-search [true positives]
AK-47	30	30	30
M45	16	11	11
PPsh	16	12	12

In terms of false alarms, the results indicate comparable performance of AMDF and peak-search in stage 2, it can be seen, that AMDF and peak-search performed comparably well, with various non-gunshot sounds achieving less false alarms using various approaches. Overall number of false alarms is less for AMDF approach. In terms of true positives, both approaches achieved identical results.

9.3 Advanced Burst Detection Combined With Individual Gunshot Detection

Since bursts consist of individual gunshots, another approach would be applying individual gunshot detection over whole frame and use AMDF afterwards. The input to individual gunshot detection is the whole frame divided into smaller subframes (11 ms), the output is a binary signal showing presence of gunshots. This binary signal serves as an input to AMDF, which determines its period. Similarly to the first approach, if detected period falls into tolerance of +/- 3 ms of nominal weapon rate of fire, the whole recording is flagged as containing gunshot burst. This method is more computationally demanding, as apart from calculating AMDF, we also need to extract features from the signal.

In the chapter dealing with advanced individual gunshot detection, we considered mainly two algorithms, ensembles of either neural networks (with two hidden layer 20 neurons each) or decision trees. In this chapter, we will compare both of these methods using approach described in previous paragraph. Both of these algorithms provide less false alarms when using TDF only (without MFCC). Tab. 37 below compares results of this mixed method using neural network and decision tree algorithms to the results of two previously tested methods. Each cell shows number of recordings flagged as bursts out of all recordings, meaning non-gunshot categories show false alarms and gunshot categories true positives.

The recognition algorithms used in this section are exactly the same as in previous chapter. Trained on the same data, false alarms from stage 1 gunshot detection, which means some of the sounds that testing datasets in this task (burst detection) and gunshot detection overlap only minimally.

Table 37 Burst recognition performance comparison with combined approach

False positives

Category	**AMDF**	**Peak-search**	**Combined approach – neural networks**	**Combined approach – decision trees**
Speech and music	11/126	46/126	0/126	1/126
Engine	54/224	43/224	0/224	2/224
Rain and thunderstorm	2/16	2/16	0/16	0/16
Birds	22/46	22/46	2/46	14/46
Dog	13/65	0/65	5/65	24/65

True positives

Weapon	**AMDF**	**Peak-search**	**Combined approach – neural networks**	**Combined approach – decision trees**
AK-47	30/30	30/30	24/30	25/30
M45	11/16	11/16	10/16	16/16
PPSh	12/16	12/16	10/16	12/16

Tab. 37 shows that false positives, an aspect which is more important than true positives for this application, are much lower using combined methods compared to simpler methods mentioned in previous chapters. When comparing the two combined methods, neural networks achieve less true positives than decision trees, but also less false alarms. For this reason, we are chosing combined approach with neural networks and TDF as final advanced burst detection algorithm.

10 DEVELOPED SOFTWARE

This chapter briefly described main scripts and some of the auxiliary functions created as a part of this thesis. The description includes names, purpose, inputs and outputs. The listed scripts use many internal functions also created as a part of this thesis, not all are described in detail in here for the sake of brevity, but they are commented in the code itself to provide better understanding. Comment descriptions include description of inputs and outputs together with dimensions and brief description of what the function does.

The main script of the work is continuously running audio event detection.

Name: audioEventDetection.m

Purpose: main script loop, takes input from device audio input, performs real-time audio analysis to detect audio limiting, gunshots and gunshot bursts, save flagged frames into dedicated folders.

Input: input directly from device audio input

Output: writes audio to file

The script dealing with advanced gunshot recognition uses folder with wav recordings as an input and outputs list of recordings flagged as containing gunshots along with exact time at which gunshot was detected. The user can choose feature set and algorithm (default algorithm is Decision Tree and feature set TDF).

Name: gunshotAdvanced.m

Purpose: determine whether preliminarily flagged gunshots really are gunshots

Input: path to folder containing wav recordings

Output: command-line list of recordings where gunshot was detected along with detection times

For advanced bursts detection, the script accepts folder path as an input and outputs names of recordings containing gunshot bursts. User can choose the method as AMDF, Psearch or Combined. All of the methods were described in chapter 9. When choosing combined method, user can also choose feature set and classification algorithm. The default method is Combined with TDF features and decision trees.

Name: burstAdvanced.m

Purpose: determine whether preliminarily flagged bursts really are bursts

Input: path to folder containing wav recordings

Output: command-line list of recordings where burst was detected

Auxiliary function to extract 5 TDF coefficients described in chapter 8.1.

Name: TDF5.m

Purpose: Extract 5 TDF coefficients from segmented input audio vector

Inputs: - *vec*: segmented audio matrix of dimension 486x*T*, where first dimension represents consecutive segment of audio and *T* is integer equal to the number of segments

- *fs*: sampling frequency of *vec*, in kHz

Outputs: - five output variables, each representing one feature from TDF5 feature set, dimension of each output variable is *T*x1, whole output is thus *T*x5

Auxiliary function to implement Average Magnitude Difference Function (AMDF) described in chapter 9.1.1.

Name: amdfFull.m

Purpose: calculate length of all periods present in input signal (with none small variance of period allowed), used in advanced burst detection

Inputs: - *audio*: audio (mono) recording to be analyzed, of dimensions 1x*S*, where *S* is the length of input in samples

- *fs*: sampling frequency of *audio* in Hz

- *tim*: length of frame to be analyzed in milliseconds

Outputs: - *Rt*: vector of detected periods with dimensions 1x*P*, where *P* is number of detected periods and is dependent on input audio

Auxiliary function to calculate signal period using Center-clipping method described in chapter 7.2.1.

Name: cclip.m

Purpose: quick method to calculate basic period of input signal, used in preliminary burst detection

Inputs: - *audio*: audio (mono) recording to be analyzed, of dimensions 1x*S*, where *S* is the length of input in samples

- *fs*: sampling frequency of *audio* in Hz

- *tim*: length of frame to be analyzed in milliseconds

- *alpha*: alpha-factor influencing threshold over which signa lis declared periodic, value typically used in speech processing is 0.3, value used in this work is 0.1

- *rf*: reduction factor, establishes threshold under which signa lis rounded to 0 with respect to maxima detected in first/last third of *audio*, value used in this work is 0.8.

- *fmax*: maximum expected frequency in Hz, can be adjusted to limit searching interval

Outputs: *Rt*: estimate of basic period of *audio* in ms

11 CONCLUSION

This work consisted of brief introdution to basics of acoustics after which we summarized needs for succesfull gunshot recognition system. The basic requirement is sound event. dataset, which we have introduced and listed a few of them. Then we have listed several important sources dedicated to tasks of acoustic scene classification and sound event. detection, along with best performing papers dedicated to gunshot recognition itself. The work itself consists in comparing features extracted from audio data, and using it in combination with various recognition algorithms.

Firstly, general comparison of features was conducted with commonly used 1024 sample frame (approx. 23 ms with sampling rate 44.1 kHz), where LPC performed the best. In the next step, frames of various sizes were compared (11 ms, 8 ms, 5 ms and 3 ms) from which 11 ms frame size was picked, due to almost identical performance as 23 ms frame and overall better than that of shorter frames. Detailed view at recognition performance with 11 ms frame confirmed insignificance of feature order for these features, we suspect this is caused by high mutual information between lower and higher feature indices. From preliminary results on several recordings, we can see that when at least 50 % of muzzle blast is present in 3 ms frame, LPC coefficients are quite stable, which is helpful when considering using overlap. This part culminated in investigation of feature variability when changing frame size, two methods, absolute and relative, were used, subsequently compared with mutual information between class labels and features and then their recognition performance was tested with neural networks. We conclude, that relative variability was good measure of feature performance, it achieved similar results as mutual information and indicated coefficients with lower indices are generally better.

Next, features were compared under different noise conditions. Features were compared under the influence of white noise with different SNRs. Results show, that while LPC coefficients are better for nearly clean recordings, MFCC perform significantly better at medium noise levels. When high power noise is present, performance of MFCC and LPC are comparable. An interesting fact is that recognition performance of LPCC dramatically decreases when increasing feature order at 0 dB SNR.

The first part of thesis concludes with description of algorithms most commonly used in sound recognition tasks, most of which were used also in this thesis. The most extensive part is dedicated to neural networks, since they are used in majority of sound recognition papers today.

The next part of the thesis deals with developing the gunshot detection algorithm itself. The first part begins with chapter 7 which elaborates on general idea of continuous audio detection. The chapter presents two algorithms for preliminary detection of gunshots and gunshot bursts. The purpose of this stage is to make sure every minute of audio is monitored. During this stage, we mostly get rid of noise and most non-gunshot sounds while still having not insignificant false alarm ratio (around 14%). In order to increase the precision of the algorithm, we use second stage, which achieves much better results but is also computationally too expensive to handle all the real-time data.

Chapter 8 deals with individual gunshot detection. We combine all the methods examined from the first chapters, beginning with comparing multiple features. In the end, we use newly developed feature set that we have introduced in [84]. Along with feature set, we compare performance of multiple machine learning algorithms, which are later ensembled for even better performance. The final algorithm consists of an ensemble of decision trees, each specializing in eliminating different sound category. The individual detection scores of decision trees are summed up, producing a voting approach successfully used in ensemble learning. In comparison with recent works of other authors dedicated to gunshot detection, our system performs significantly better, best paper in DCASE 2017 track "Detection of rare sound events" [86] achieved error rate of 16% in gunshot class, while our system achieves performance with equivalent score of 2%.

The final chapter of this work consists of advanced burst detection. Multiple methods are compared but the main topic of the chapter is work with periodicity, how to establish precise period measurement of bursts and to compare similarity of adjacent periods. The final method in this chapter constitutes a combination of single gunshot detection and periodicity examination.

11.1 Future Work

This work was dedicated to feature analysis, selection and comparison of recognition algorithms and also general proposal of system architecture. The demands of the proposed system were estimated using algorithm execution timing, which also served as basis for confirming the need for two recognition stages.

Since the original idea inspiring this thesis was development of gunshot recognizing modules to be placed on tracking collars used in wildlife protection, there is still some development work to do. The most obvious step would be implementation of the whole system into compact, portable form which could be used along with tracking collars. Choosing proper platform would depend both on computational power needed and energy demands, which are constraints that go in opposite direction, so further testing would be needed. Additionally, there is a question of how to distribute computational demands. Since tracking collars are periodically sending their possition to authorities, there is a possibilty of sending also preliminarily flagged recordings. This way, advanced recognition, which is more computationally demanding, could be performed on a more powerful computers with steady energy supply. This would also be advantageous, since flagged recordings could be reviewed in place by an employee and possibly eliminate further false alarms. This method would however require further analysis of energy budgeting and further considerations such as how often to upload preliminarily flagged elements. More frequent uploads (such as in real time), would allow quick response, however frequent transmission is energetically demanding. On the other hand, sending the data (collected in a temporary buffer) for further analysis once every hour or two might still allow sufficiently prompt response, while saving battery time. There are a lot of possibilities that would depend on the situation in place, where local factors must be considered and the solution tailored by local needs.

REFERENCES

[1] A. I. Tarrero, „Propagación del sonido en bosques. Análisis comparativo de las medidas in situ, en laboratorio y de los valores predichos por un modelo,“ Ph.D. dissertation, Universidad de Valladolid, 2002.

[2] R. C. Maher, "Acoustical Characterization of Gunshots," in 2007 IEEE Workshop on Signal Processing Applications for Public Security and Forensics, 2007, pp. 1–5.

[3] R. Stoughton, "Measurements of small-caliber ballistic shock waves in air," *The Journal of the Acoustical Society of America*, vol. 102, no. 2, pp. 781–787, Aug. 1997. DOI: 10.1121/1.419904

[4] J. Salamon, C. Jacoby and J. P. Bello, „A Dataset and Taxonomy for Urban Sound Research,“ in *Proceedings of the ACM International Conference on Multimedia - MM '14*. New York, New York, USA: ACM Press, 2014, pp. 1041-1044. DOI: 10.1145/2647868.2655045.

[5] F. Font, G. Roma, and X. Serra, „Freesound technical demo,“ in *Proceedings of the 21st ACM international conference on Multimedia - MM '13*. New York, New York, USA: ACM Press, 2013, pp. 411-412. DOI: 10.1145/2502081.2502245.

[6] *SoundBible.com*. Accessed on: 2019-July-19. [Online]. Available: http://soundbible.com/.

[7] A. Cappelletti, B. Lepri, N. Mana, F. Pianesi, and M. Zancanaro, „Netcarity Multimodal Data Collection,“ in *Proceedings of CHI Workshop on Developing Shared Home Behaviour Datasets to Advance HCI and Ubiquitous Computing Research*. Boston, USA, 2009.

[8] M. Grassi *et al.*, "A hardware-software framework for high-reliability people fall detection," in *2008 IEEE SENSORS*, 2008, pp. 1328–1331. DOI: 10.1109/ICSENS.2008.4716690.

[9] P. Foster, S. Sigtia, S. Krstulovic, J. Barker, and M. D. Plumbley, "Chime-home: A dataset for sound source recognition in a domestic environment," in *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2015, pp. 1–5. DOI: 10.1109/WASPAA.2015.7336899.

[10] T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere, „The Million Song Dataset,“ in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011),* 2011.

[11] *British Library Sounds*. Accessed on: 2019-July-19. [Online]. Available: http://sounds.bl.uk/

[12] R. Ranft, "Natural sound archives: past, present and future," *An. Acad. Bras. Cienc.*, vol. 76, no. 2, pp. 455–465, Jun. 2004. DOI: /S0001-37652004000200041.

[13] *Free sound effects*. Accessed on: 2019-July-19. [Online]. Available: http://www.airbornesound.com/sound-effects-library/free-sound-effects/

[14] A. Mesaros, T. Heittola, T. Virtanen, E. Fagerlund, A. Hiltunen, and T. Heittola, „TUT Acoustic scenes 2016, Development dataset,“ Zenodo, 2016. DOI: 10.5281/zenodo.45739.

[15] A. Mesaros, E. Fagerlund, A. Hiltunen, T. Heittola, T. Heittola, and T. Virtanen, „TUT Sound events 2016, Development dataset,“ Zenodo, 2016. DOI: 10.5281/zenodo.45759

[16] T. Heittola, *Sound event detection in real life audio*. Accessed on: 2019-July-19. [Online]. Available: http://www.cs.tut.fi/sgn/arg/dcase2016/task-sound-event-detection-in-real-life-audio

[17] H. Ghaderi and P. Kabiri, „Automobile Independent Fault Detection based on Acoustic

Emission Using FFT,“ in *Proceedings of the International Conference & Exhibition NDT, Singapore*, 2011.

[18] H. Ghaderi and P. Kabiri, “Fourier transform and correlation-based feature selection for fault detection of automobile engines,” in *The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012)*, 2012, pp. 514–519. DOI: 10.1109/AISP.2012.6313801.

[19] M. Pleva, E. Vozáriková, L. Dobos, and A. Čižmár, „The Joint Database of Audio Events and Backgrounds for Monitoring of Urban Areas,“ *Journal of Electrical and Electronics Engineering*, vol. 4, no. 1, pp. 185–188, Jan. 2011.

[20] *The free firearm sound effects library*. Accessed on: 2019-July-19. [Online]. Available: http://www.stillnorthmedia.com/firearm-sound-library.html

[21] A. van den Oord *et al.*, “WaveNet: A Generative Model for Raw Audio,” *arXiv: 1609.03499,* Sep. 2016.

[22] *Sounds of New York City*. Accessed on: 2019-July-19. [Online]. Available: https://wp.nyu.edu/sonyc/

[23] *National Science Foundation*. Accessed on: 2019-July-19. [Online]. Available: https://www.nsf.gov/awardsearch/showAward?AWD_ID=0713334

[24] *National Science Foundation*. Accessed on: 2019-July-19. [Online]. Available: https://www.nsf.gov/awardsearch/showAward?AWD_ID=1054960

[25] *Engineering and Physical Sciences Research council*. Accessed on: 2019-July-19. [Online]. Available: http://gow.epsrc.ac.uk/NGBOViewGrant.aspx?GrantRef=EP/L020505/1

[26] *Engineering and Physical Sciences Research council*. Accessed on: 2019-July-19. [Online]. Available: http://gow.epsrc.ac.uk/NGBOViewGrant.aspx?GrantRef=EP/N014111/1

[27] *Engineering and Physical Sciences Research council*. Accessed on: 2019-July-19. [Online]. Available: http://gow.epsrc.ac.uk/NGBOViewGrant.aspx?GrantRef=EP/M507088/1

[28] *Engineering and Physical Sciences Research council*. Accessed on: 2019-July-19. [Online]. Available: http://gow.epsrc.ac.uk/NGBOViewGrant.aspx?GrantRef=EP/G007144/1

[29] A. Dufaux, L. Besacier, M. Ansorge, and F. Pellandini, “Automatic sound detection and recognition for noisy environment,” in *2000 10th European Signal Processing Conference*, 2000, pp. 1–4.

[30] A. Dufaux, „Detection and recognition of impulsive sounds signals,“ Ph.D. dissertation, Institute de Microtechnique, Neuchatel, Switzerland, 2001.

[31] A. Temko, D. Macho, and C. Nadeu, „Selection of features and combination of classifiers using a fuzzy approach for acoustic event classification,“ in *Proceedings of the INTERSPEECH conference*, 2005, pp. 2989–2992.

[32] A. Temko and C. Nadeu, “Acoustic event detection in meeting-room environments,” *Pattern Recognition Letters*, vol. 30, no. 14, pp. 1281–1288, Oct. 2009. DOI: 10.1016/j.patrec.2009.06.009. ISSN 01678655.

[33] T. Butko, A. Temko, C. Nadeu, and C. Canton-Ferrer, “Fusion of audio and video modalities for detection of acoustic events,” in *INTERSPEECH*, 2008, pp. 123–126.

[34] T. Butko *et al.*, “Acoustic Event Detection Based on Feature-Level Fusion of Audio and Video Modalities,” *EURASIP J. Adv. Sig. Proc.*, vol. 2011, 2011. DOI: 10.1155/2011/485738. ISSN 1687-6180.

[35] M. U. B. Altaf, T. Butko, and B. Juang, “Acoustic Gaits: Gait Analysis With Footstep

Sounds," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 8, pp. 2001–2011, Aug. 2015. DOI: 10.1109/TBME.2015.2410142. ISSN 0018-9294.

[36] C. Clavel, T. Ehrette, and G. Richard, "Events Detection for an Audio-Based Surveillance System," in *2005 IEEE International Conference on Multimedia and Expo*, 2005, pp. 1306–1309. DOI: 10.1109/ICME.2005.1521669. ISBN 0-7803-9331-7.

[37] T. Giannakopoulos, D. Kosmopoulos, A. Aristidou, and S. Theodoridis, "Violence Content Classification Using Audio Features," in *Advances in Artificial Intelligence*, 2006, pp. 502–507. DOI: 10.1007/11752912_55.

[38] T. Giannakopoulos, A. Makris, D. Kosmopoulos, S. Perantonis, and S. Theodoridis, "Audio-Visual Fusion for Detecting Violent Scenes in Videos," in *Artificial Intelligence: Theories, Models and Applications*, 2010, pp. 91–100. DOI: 10.1007/978-3-642-12842-4_13.

[39] R. C. Maher, "Modeling and Signal Processing of Acoustic Gunshot Recordings," in *2006 IEEE 12th Digital Signal Processing Workshop 4th IEEE Signal Processing Education Workshop*, 2006, pp. 257–261. DOI: 10.1109/DSPWS.2006.265386. ISBN 1-4244-0535-1.

[40] T. Routh and R. Maher, "Determining the Muzzle Blast Duration and Acoustical Energy of Quasi-Anechoic Gunshot Recordings," presented at the Audio Engineering Society Convention 141, 2016.

[41] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, 2007, pp. 21–26. DOI: 10.1109/AVSS.2007.4425280.

[42] L. Gerosa, G. Valenzise, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection in noisy environments," in *2007 15th European Signal Processing Conference*, 2007, pp. 1216–1220.

[43] I. Trancoso, J. Portêlo, M. Bugalho, J. Neto, and A. Serralheiro, "Training audio events detectors with a sound effects corpus," in *Proc. Interspeech 2008*, 2008.

[44] J. Portelo, M. Bugalho, I. Trancoso, J. Neto, A. Abad, and A. Serralheiro, "Non-speech audio event detection," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1973–1976. DOI: 10.1109/ICASSP.2009.4959998.

[45] T. Pellegrini, J. Portêlo, I. Trancoso, A. Abad, and M. Bugalho, „Hierarchical clustering experiments for application to audio event detection," in *Proceedings of the 13th International Conference on Speech and Computer*, 2009.

[46] X. Zhou, X. Zhuang, M. Liu, H. Tang, M. Hasegawa-Johnson, and T. Huang, "HMM-Based Acoustic Event Detection with AdaBoost Feature Selection," in *Multimodal Technologies for Perception of Humans*, 2008, pp. 345–353. DOI: 10.1007/978-3-540-68585-2_33.

[47] Zhuang, M. Hasegawa-Johnson, and X. Zhou, "Feature analysis and selection for acoustic event detection," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 17–20. DOI: 10.1109/ICASSP.2008.4517535.

[48] M. A. Hasegawa-Johnson, J.-T. Huang, S. King, and X. Zhou, "Normalized recognition of speech and audio events," *The Journal of the Acoustical Society of America*, vol. 130, no. 4, pp. 2524–2524, Oct. 2011. DOI: 10.1121/1.3655075. ISSN 0001-4966.

[49] K. Lin, X. Zhuang, C. Goudeseune, S. King, M. Hasegawa-Johnson, and T. S. Huang, "Improving faster-than-real-time human acoustic event detection by saliency-maximized audio visualization," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 2277–2280. DOI: 10.1109/ICASSP.2012.6288368.

[50] M. Pleva, E. Vozáriková, S. Ondáš, J. Juhár, and A. Čižmár, „Automatic detection of audio events indicating threats,“ in *Proceedings of the IEEE International Conference on Multimedia Communications, Services and Security, Krakow*, 2010, vol. 6.

[51] E. Vozáriková, J. Juhár, and A. Čižmár, “Acoustic Events Detection Using MFCC and MPEG-7 Descriptors,” in *Multimedia Communications, Services and Security*, 2011, pp. 191–197. DOI: 10.1007/978-3-642-21512-4_23.

[52] E. Kiktova, M. Lojka, M. Pleva, J. Juhar, and A. Cizmar, “Comparison of Different Feature Types for Acoustic Event Detection System,” in *Multimedia Communications, Services and Security*, 2013, pp. 288–297. DOI: 10.1007/978-3-642-38559-9_25.

[53] M. Lojka, M. Pleva, E. Kiktová, J. Juhár, and A. Čižmár, “Efficient acoustic detector of gunshots and glass breaking,” *Multimed Tools Appl*, vol. 75, no. 17, pp. 10441–10469, Sep. 2016. DOI: 10.1007/s11042-015-2903-z. ISSN 1380-7501.

[54] R. Malkin, D. Macho, A. Temko, and C. Nadeu, „First evaluation of acoustic event classification systems in CHIL project,“ in *Proceedings of the HSCMA'05 Workshop*, 2005.

[55] A. Kumar, P. Dighe, R. Singh, S. Chaudhuri, and B. Raj, “Audio event detection from acoustic unit occurrence patterns,” in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 489–492. DOI: 10.1109/ICASSP.2012.6287923.

[56] B. Elizalde *et al,* „Experiments on the DCASE Challenge 2016: Acoustic Scene Classification and Sound Event Detection in Real Life Recording,“ in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2016.

[57] S. Chaudhuri, M. Harvilla, and B. Raj, „Unsupervised Learning of Acoustic Unit Descriptors for Audio Content Representation and Classification,“ in *Proceedings of the Interspeech conference*, 2011, pp. 2265–2268.

[58] S. Gharib, K. Drossos, E. Cakir, D. Serdyuk and T. Virtanen. "Unsupervised Adversarial Domain Adaptation for Acoustic Scene Classification". *Detection and Classification of Acoustic Scenes and Events*. 2018.

[59] E. Cakir and T. Virtanen. "End-to-End Polyphonic Sound Event Detection Using Convolutional Recurrent Neural Networks with Learned Time-Frequency Representation Input". *2018 International Joint Conference on Neural Networks, IJCNN 2018 - Proceedings*. 2018.

[60] E. Çakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, “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, Jun. 2017.

[61] E. Kiktova, M. Lojka, M. Pleva, J. Juhar, and A. Cizmar, “Gun type recognition from gunshot audio recordings,” in *3rd International Workshop on Biometrics and Forensics (IWBF 2015)*, 2015, pp. 1–6. DOI: 10.1109/IWBF.2015.7110240.

[62] F. J. González-Castaño *et al.*, “Acoustic Sensor Planning for Gunshot Location in National Parks: A Pareto Front Approach,” *Sensors (Basel)*, vol. 9, no. 12, pp. 9493–9512, Nov. 2009. DOI: 10.3390/91209493. ISSN 1424-8220.

[63] M. Hrabina and M. Sigmund, “Acoustical detection of gunshots,” in *2015 25th International Conference Radioelektronika (RADIOELEKTRONIKA)*, 2015, pp. 150–153. DOI: 10.1109/RADIOELEK.2015.7128993.

[64] M. Slaney, “Mixtures of probability experts for audio retrieval and indexing,” in *Proceedings. IEEE International Conference on Multimedia and Expo*, 2002, vol. 1, pp.

345–348. DOI: 10.1109/ICME.2002.1035789.

[65] I. L. Freire and J. A. Apolinário Jr, „Gunshot detection in noisy environments,“ in *Proceeding of the 7th International Telecommunications Symposium, Manaus, Brazil*, 2010, pp. 1–4.

[66] T. Ahmed, M. Uppal, and A. Muhammad, "Improving efficiency and reliability of gunshot detection systems," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 513–517. DOI: 10.1109/ICASSP.2013.6637700.

[67] E. Vozarikova, J. Juhar, and A. Cizmar, „Acoustic Event Detection Based on MRMR Selected Feature Vectors,“ *Journal of Electrical & Electronics Engineering*. 2012, vol. 5, no. 1, pp. 277-282. ISSN 18446035.

[68] M. Hrabina, "Analysis of linear predictive coefficients for gunshot detection based on neural networks," in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, 2017, pp. 1961–1965.

[69] A. C. Kelly and C. Gobl, "A comparison of mel-frequency cepstral coefficient (MFCC) calculation techniques," *Journal of Computing*, vol. 3, no. 10, p. 5, 2011.

[70] M. Hrabina and M. Sigmund, "Audio Event Database Collected for Gunshot Detection in Open Nature (GUDEON)". *Journal of the Audio Engineering Society*, vol. 67, pp. 54-59. DOI: 10.17743/jaes.2018.0075.

[71] I. Jokic, V. Delic, S. Jokic, Z. Peric, „Automatic Speaker Recognition Dependency on Both the Shape of Auditory Critical Bands and Speaker Discriminative MFCCs,“ in *Advances in electrical and computer engineering*, 2015

[72] M. Hrabina and M. Sigmund, "Comparison of feature performance in gunshot detection depending on noise degradation," in *2017 27th International Conference Radioelektronika (RADIOELEKTRONIKA)*, 2017, pp. 1–4.

[73] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *2010 18th European Signal Processing Conference*, 2010, pp. 1267–1271.

[74] O. Gencoglu, T. Virtanen, and H. Huttunen, "Recognition of acoustic events using deep neural networks," in *2014 22nd European Signal Processing Conference (EUSIPCO)*, 2014, pp. 506–510.

[75] R. Caruana and A. Niculescu-Mizil, „An empirical comparison of supervised learning algorithms,“ in *Proceedings of the 23rd international conference on Machine learning* (ICML '06). ACM, New York, NY, USA, 161-168. DOI: 10.1145/1143844.1143865

[76] G. Bonaccroso, *Machine Learning Algorithms: A reference guide to popular algorithms for data science and machine learning,* PACKT, 2017. [E-book] Available: Amazon e-book.

[77] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When Is 'Nearest Neighbor' Meaningful?," in *Database Theory — ICDT'99*, 1999, pp. 217–235.

[78] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[79] S. Ioffe, Ch. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Computing Research Repository* (arxiv.org), 2015

[80] H. Cai, L. Zhu, and S. Han, "ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware," *arXiv:1812.00332*, Dec. 2018.

[81] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech

recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[82] L. Rabiner and R. Schafer, *Theory and Applications of Digital Speech Processing*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2010.

[83] M. Sigmund, "Statistical analysis of fundamental frequency based features in speech under stress," *Information Technology and Control*, vol. 42, no. 3, pp. 286-291, 2013. DOI: 10.5755/j01.itc.42.3.389

[84] M. Hrabina and M. Sigmund, "Gunshot recognition using low level features in the time domain," in *2018 28th International Conference Radioelektronika (RADIOELEKTRONIKA)*, 2018, pp. 1–5. DOI: 10.1109/RADIOELEK.2018.8376372

[85] M. Sigmund, "Influence of psychological stress on formant structure of vowels," *Elektronika ir Elektrotechnika*, vol. 18, no. 10, pp. 45-48, 2012. DOI: 10.5755/j01.eee.18.10.3059

[86] H. Lim, J. Park and Y. Han, „Rare Sound Event Detection Using {1D} Convolutional Recurrent Neural Networks, in *DCASE 2017*," 2017

[87] M. Hrabina and M. Sigmund, "Implementation of developed gunshot detection algorithm on TMS320C6713 processor," in *2016 SAI Computing Conference (SAI)*, 2016, pp. 902–905. DOI: 10.1109/SAI.2016.7556087.

[88] P. G. Weissler and M. T. Kobal, "Noise of police firearms," *The Journal of the Acoustical Society of America*, vol. 56, no. 5, pp. 1515–1522, Nov. 1974. DOI: 10.1121/1.1903473