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

SHORT VERSION OF DOCTORAL THESIS
ZKRÁCENÁ VERZE DIZERTAČNÍ PRÁCE

AUTHOR — Ing. Martin Hrabina
AUTOR PRÁCE

SUPERVISOR — prof. Ing. Milan Sigmund, CSc.
ŠKOLITEL

BRNO 2019

KĽÚČOVÉ SLOVÁ

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

KEYWORDS

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

Dizertačná práca je uložená:

Vědecké oddělení

Fakulta elektrotechniky a informačních technologií

Vysoké Učení Technické v Brně

Technická 10

612 00

Brno

CONTENTS

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 be then incorporated within 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 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. 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. 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 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 on comparing them, and a comparison of their effectiveness under clean and noisy conditions is made. The first part concludes by introducing frequently used algorithms in sound event recognition. The second part consists of the contributed work itself. It presents new proposed features and compares them to some previously used features. It also proposes a system for real-time event detection with preliminary categorization into single 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. The short version of thesis includes only the most important parts of the above mentioned content.

1 SOUND EVENTS DATABASES

The first step in sound recognition is to assemble proper audio dataset containing sounds of interest and possibly other sounds to be distinguish. The audio in dataset is usually labelled with various classes or categories (such as gunshots and sounds of barking dog), number of recordings in each class should be approximately equal, at least for training purposes. Since the conception of sound recognition field, multiple audio datasets were compiled, some tailored for specific purpose and some with the aim of being universally used to compare various algorithms. The following section lists some of those datasets and subsequently describes the dataset used in this work.

1.1 Existing sound event datasets

This section lists multiple audio datasets compiled mostly for various audio recognition tasks, but some also for other purposes (such as movie industry). For the task of sound recognition, audio without any artificial alterations is preferred, this includes not only synthethic audio (such as special effects for movies), but many times also lossy audio encoding.

For purposes of sound recognition, database focused on urban sounds [1]. This urban sound database consists of 10 sound classes (air conditioner, car horn, dog bark, drilling, engine, gunshots, playing children, jackhammer, siren and street music). This publication also offers taxonomy of urban sounds due to lack of common vocabulary. Subset of databases is dedicated to domestic and indoor sounds, for example in case of fall detection in elderly care. Netcarity project supported several specialized datasets, one such dataset focused on daily activities, such as ironing, eating or watching TV. Another database under this project is described in [2], it consists of 450 events with approx. 210 falls performed by 13 different actors. In this work, accelerometer and 3D camera data were collected as well. Datasets with indoor sounds such as appliances and gender or age classified speech was popular also with other authors. Database [3] is created for movie making purposes, there are both free and paid collections made of recordings of crowds in different places and ambience sounds. DCASE 2016 Challenge used [4] and [5], databases of indoor and outdoor sounds and events, both described in [6].

Specialized gunshot sounds databases are scarcer. First database consists of approximately 800 gunshots and other dangerous sound (e.g. explosions, car crashes …) recordings [7], it is available only to INDECT project partners. Next database is dedicated only to gunshots and mechanical sounds produced by weapons [8]. It was compiled for movie making purposes and consists of around 1100 recordings of gunshots and additional mechanical sounds, recorded in WAV format with high quality.

Apart from above mentioned datasets, there is a wide variety of various specialized datasets assembled for certain purpose, such as acoustic fault analysis of combustion engines, or collection of natural sounds in “British Library Sounds”. Then, there is different approach to assembling datasets, such as “Million song dataset”, which is a dataset of a million contemporary popular sounds. The dataset, in order to avoid copyright issues, does not contain any actual songs, but only extracted features.

Apart from datasets assembled by individuals and collectives, there are also crowdsourced datasets such as Freesound and Soundbible. Crowdsourced datasets come with weakness that they are weakly curated, and so contain mislabelled sounds or synthetic sound effects.

1.1.1 Our Dataset

Our dataset consists of selected audio data from previously mentioned datasets, as well as some recordings made by us. The dataset can be divided by various criteria, e.g. by duration into two groups. 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. The division can further be into natural sounds (such as barking dog or rain and thunderstorm) and man-made sounds (e.g. sound of idling engine). Arguably, the most important divison for this work is division into gunshots (positive class) and non-gunshots (negative class).

Since all the data comes from various sources, original number of channels, sampling frequencies, quantization and even audio format differ widely. To ensure uniformity, all sounds were converted to mono signal (averaging between channels), downsampled to 44.1 kHz and quantized with 16-bits. Recordings in lossless formats (such as .flac) were converted to .wav and recordings in lossy formats were dropped.

The compiled dataset contains a lot audio data, which would be infeasible to use in its entirety, thus most experiments operate with only parts of the dataset. During the experiments, classes of interest are picked and subsequently, required number of randomly chosen segments is used. For example, the last chapter operates with 4 non-gunshot classes whose sounds originate from different datasets, only sounds with certain minimum amplitude are chosen and only sounds flagged as gunshots by a simple algorithm. This way, we ensure testing the final algorithm with over 30000 sound frames from various sources.

Gunshots, focus of this work, come from [8] and include 1532 events from 25 different weapons of different categories (such as handguns, hunting rifles and assault rifles). In total, the largest subset of sounds of the same type is represented by 374 gunshots from the assault rifle AK-47. The distinctive sound of gunshots come from two main sources. One, the muzzle blast, is the sound of gases expanding from the weapon, it has quite a distinctive shape which is known as the N-wave [2]. The other is shockwave and it is produced by hypersonic bullets along their path, thus shockwave is not detectable behind the shooter. We have investigated the similarity of individual gunshots in this dataset comparing gunshots from the same weapon (AK-47). In all cases, gunshots were extracted from the recordings using a rectangular window with a length of 30 ms and then, each extracted gunshot signal was normalized to the size –1 to +1 to eliminate different intensity of sounds. In the next step, all gunshots were time synchronized by fixing in the maximum point, 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 $\mu(t)$ and standard deviation $\sigma(t)$ were estimated sample by sample along the whole gunshot length. Fig. 1 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. The darker shade in Fig. 1 means that the waveforms are more concentrated around the average waveform. The leftmost part is shockwave and we can see that its position varies, this is due to different geometries of gunshots. The part on the right, around the maximum point is muzzle blast, some variability can be seen, but the typical N-shape is distinguishable.

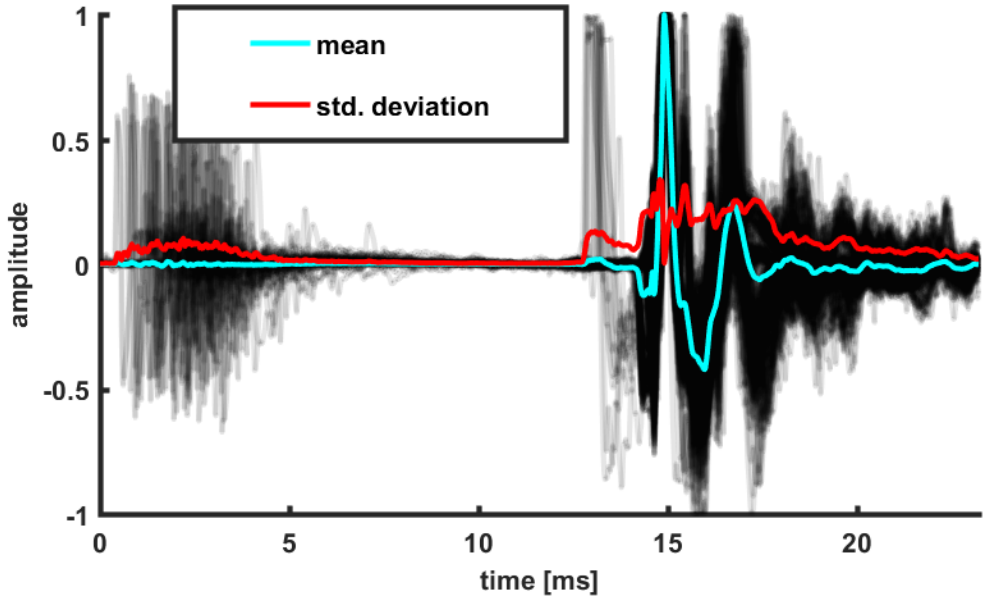

Fig. 1 Gunshot waveforms stacked on top of each other

2 COMPARISON OF FREQUENTLY USED FEATURES

This chapter compares several feature sets under different conditions and its aim is to discover the best setup that would be used later in the gunshot recognition system. Examined 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 each training.

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 (1), (2) and (3) respectively:

$$recall = \frac{TP}{P} = \frac{TP}{TP + FN}, \tag{1}$$

$$precision = \frac{TP}{TP + FP}, \tag{2}$$

$$F - score = 2 \cdot \frac{precision \cdot recall}{precision + recall}, \quad (3)$$

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. 1 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 will be used as a starting point for further [9].

Table 1 Precision (2) of various features with frame length 23 ms [9]

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 %

2.1 Effects of Frame Length and Position on Feature Variability

This chapter 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. 2. As to the event position, gunshots were segmented into frames of length 3 ms with 50% overlap as shown in Fig. 3. Variability/stability observation consisted essentially of comparing values of coefficients under changing conditions. Illustrative results are shown in Fig. 4, this represents LPC coefficients (which were the most stable from the three sets) of order 20, individual lines represent individual coefficients.

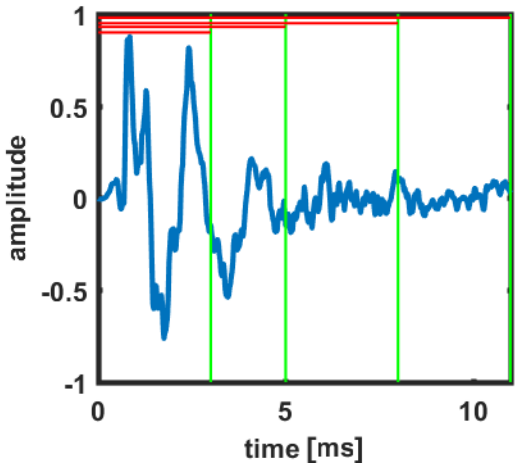

Fig. 2 Increasing frame size

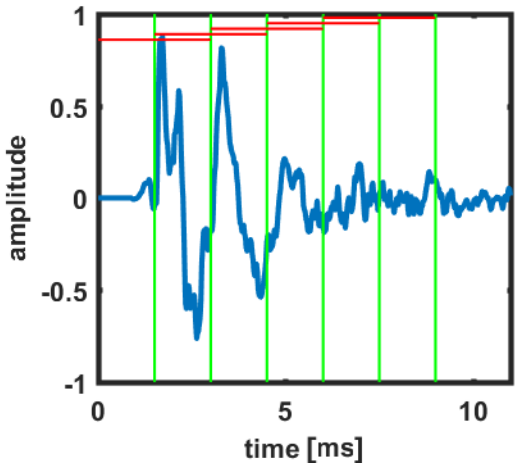

Fig. 3 Shifting event position within frame

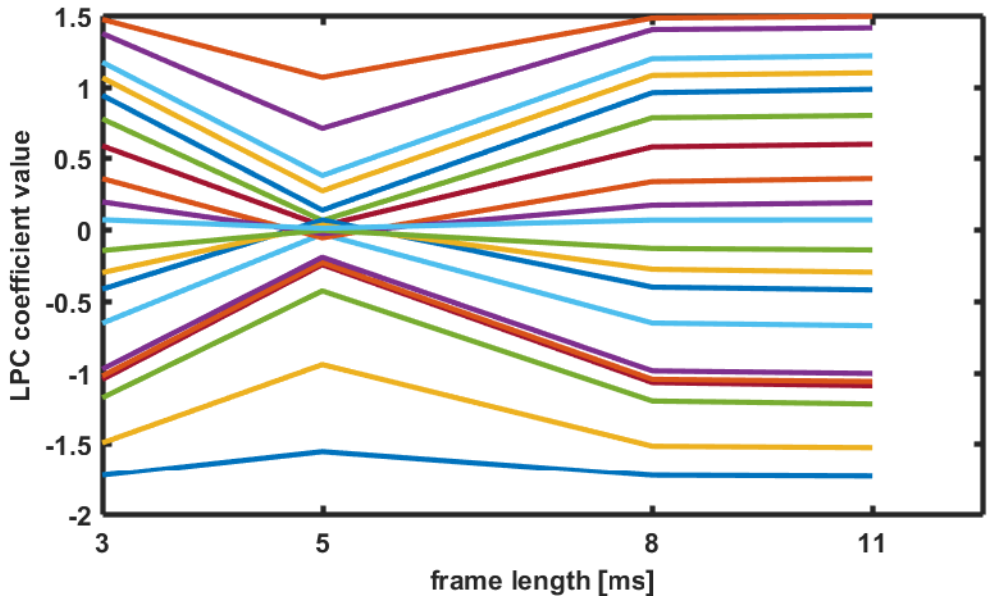

Fig. 4 LPC coefficients (order 20) - increasing frame size

During the next step, feature performance was estimated for all frame lengths and for different orders (8 to 30). Fig. 5 shows progressively decreasing recall for MFCC with decreasing frame lengths. As can be seen from Tab. 3, 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. 1 [9]). Thus, we will explore viability of frame length of 11 ms in the following tests, unless otherwise noted. Achieved results for frame length of 11 ms are presented in Tab. 2 and Tab. 3. Observation also shows, that there is no substantial improvement beyond order 12 for LPC or LPCC.

Table 2 Recall (1) for frame length 11 ms

Order	Feature set			
	LPC	LPCC	MFCC20	MFCC
8	84.2	82.1	79.9	73.8
10	84.6	84.1	81.6	76.8
12	87.0	84.3	82.8	79.2
14	86.0	82.9	83.8	77.2
16	86.3	85.0	85.4	80.4
18	86.9	83.9	81.1	81.9
20	86.4	83.4	84.5	81.8
22	86.7	83.9	78.2	82.2
24	84.8	83.1	82.9	83.0
26	85.3	84.4	81.3	82.0
28	85.7	84.6	84.3	84.8
30	85.9	83.9	85.8	84.5

Table 3 Precision (2) 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

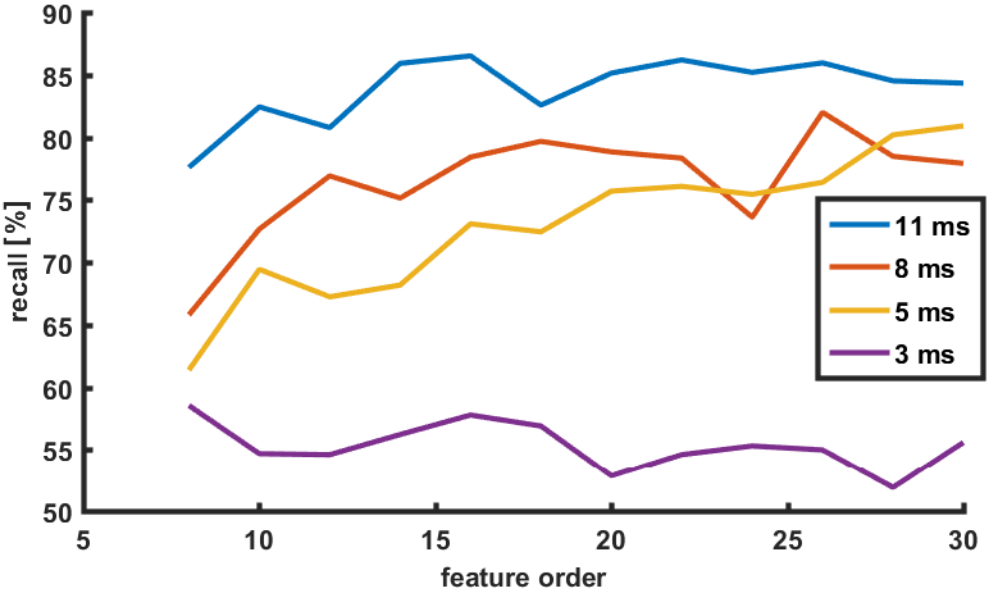

Fig. 5 Recall (1) of MFCC coefficients of various orders for different frame size

Figur 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.

In order to assess vaibility of features, we propose comparing their differences in two ways. We can compare absolute differences between features, which might be problematic due to different feature scales. And we can compare relative differences, defined by (4):

$$\overline{\Delta_m} = \frac{\sum_{k=1}^{K}\sum_{p=1}^{P-1}\frac{\left(a_{m,k,p+1} - a_{m,k,p}\right)}{\left(a_{m,k,p+1} + a_{m,k,p}\right)}}{K(P-1)}, \tag{4}$$

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. Best 3 coefficients (i.e. coefficients with the lowest variability calculated with (4)) from each order were summed and compared with other orders, Tab. 4 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 4 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 confirm viability of our metrics, feature performance will be tested using neural networks. Additionally, mutual information between class label and feature value will be calculated, as shown in (5). This measure reflects relevance of the feature in classification process for given classes.

$$I(x,y) = \sum_{i,j} p\left(x_i, y_j\right) \cdot \log \frac{p\left(x_i, y_j\right)}{p(x_i) \cdot p\left(y_j\right)}, \tag{5}$$

where $I(x,y)$ is mutual information, $p(x_i)$ if probability distribution of features, $p(y_j)$ is probability distribution of classes and $p(x_i,y_j)$ is joint probability. In this step, we are not dealing with mutual information between individual features.

Tab. 5 summarizes best results of mutual information tests for all feature sets. Mutual information was calculated according to (5) shown and explained above. For now, no mutual information between features was examined. In general, the most mutual information was concentrated in lower coefficients.

Table 5 Best orders and coefficients – Mutual information (5)

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

Fig. 6 presents performance represented by F-score of LPC chosen by different methods. 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").

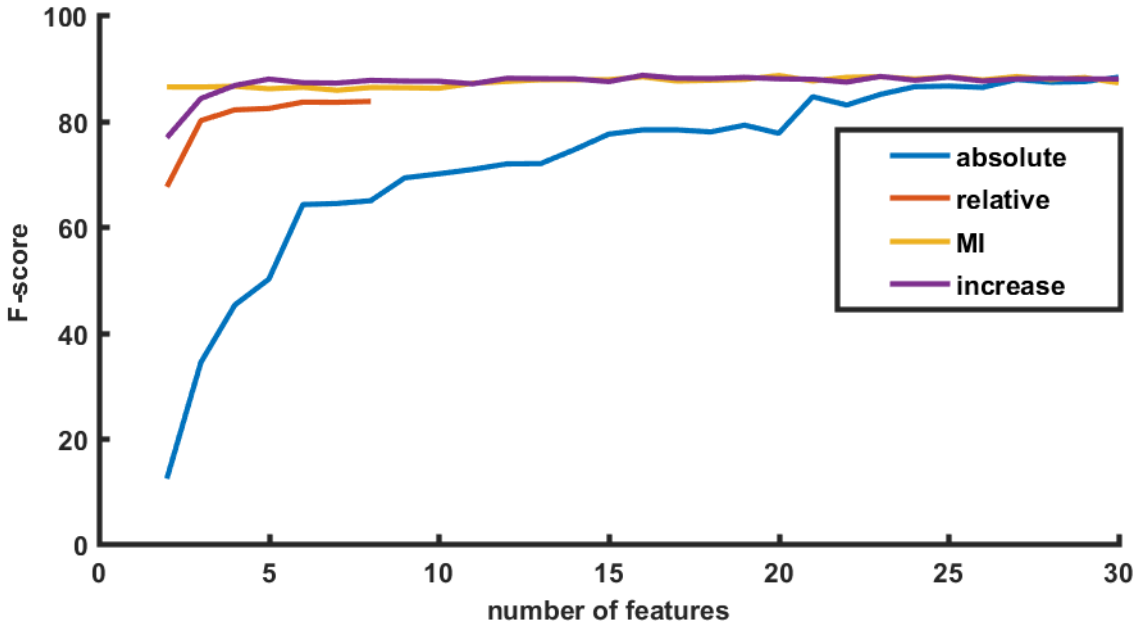

Fig. 6 F-score (3) of different number of LPC coefficients for various selection methods

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 slowest ascend, probably because lowest absolute differences are concentrated in higher index coefficients, which according to (5) have comparably lower mutual information with class label in contrast with lower index coefficients. Overall, all features achieved similar results with MFCC requiring more features than LPC and LPCC.

2.2 Comparison of Various MFCC Settings

As a next step, we have investigated extraction of MFCC coefficients, which has a lot of possibilities for modifications. We base the usual proces of MFCC extraction on [10]. The section briefly describes dataset used and evaluation method, then proceeds to explore various ways in which we can modify MFCC extraction.

We used GUDEON [11] dataset to generate audio for this experiment. 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 evaluation 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 positives ratio (TPR) and true negatives ratio (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“). The next step was to vary different parameters of extraction, compare the results and possibly adjust parameters for optimal performance. A series of tests compare effects of increased bandwidth, increased bandwidth and number of filter banks, various frequency scales, and ultimately filter shape. Other tests included varying feature order to bank ratio and input normalized so that maximum absolute value is equal to one.

Tab. 6 shows results of increasing bandwidth, other things unchanged. Tab. 7 increases bandwidth and number of filter banks. Subsequently, we have adjusted number of filter banks to 32 and bandwidth to 8 kHz to reflect the best attained results so far and varied frequency scale, the results are presented in Fig. 7. Tab. 8 compares different filter shapes with baseline setup (except with linear scale).

Table 6 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 7 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

Table 8 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
Exponential	71.9 %	87.1 %	74.7

The results indicate that feature order to bank ratio at, or below 0.5 is performing better in comparison with higher ratios (such as 1). Normalizing input to maximum absolute value equal to 1 had detrimental effects on recognition, decreasing it by around 7%.

As a result, we conclude 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 conclude further experiments with real-time gunshot detection.

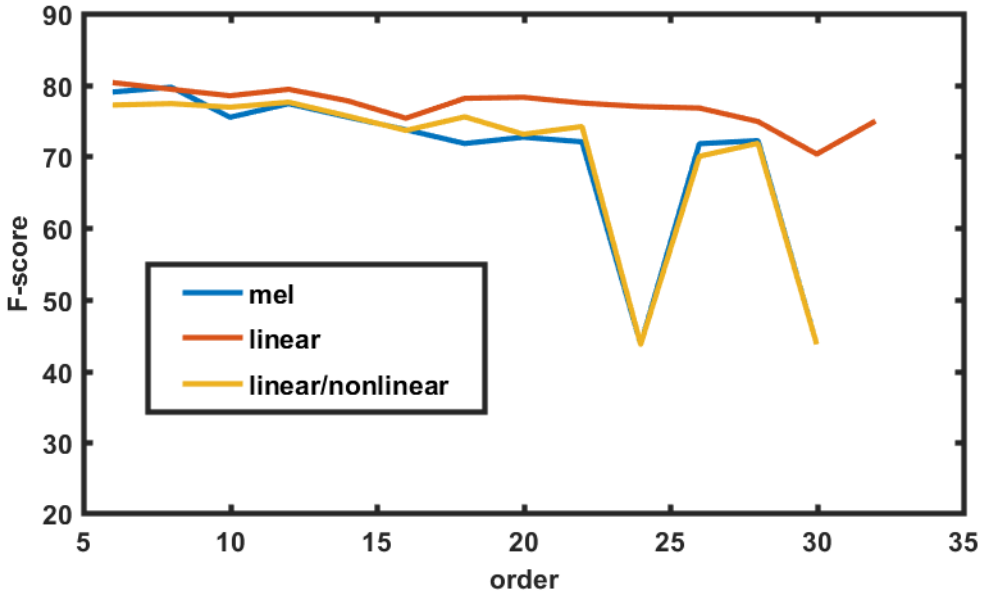

Fig. 7 Gunshot recognition with increasing MFCC order

2.3 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 briefly 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. Spectral characteristics of the noises differ in occupied bandwidth, with progression $BW_{AWGN}>BW_{Rain}>BW_{Engine}$, with engine concentrating most of its power below 1 kHz. Signal-to-noise ratios (SNR) were set to 0 dB, 10 dB, 20 dB and 30 dB, for reference, also clean signal was used. During tests, recordings with equal amount of noise were used both for training and for testing. Fig. 8 and 9 show F-score of MFCC and LPC features under different SNR conditions using white noise, performance of LPCC was similar to LPC with F-score decreasing with increasing order at SNR = 0dB. The results indicate, that LPC perform better in conditions with little noise. However when SNR drops to 20dB or 10 dB, performance of MFCC degrades much slower than that of LPC, even to the point where MFCC perform significantly better. At noise levels 0dB, performance of LPC and MFCC is comparable again. Trends under degradation with engine and rain sounds were similar to white noise, but the degree was slightly different.

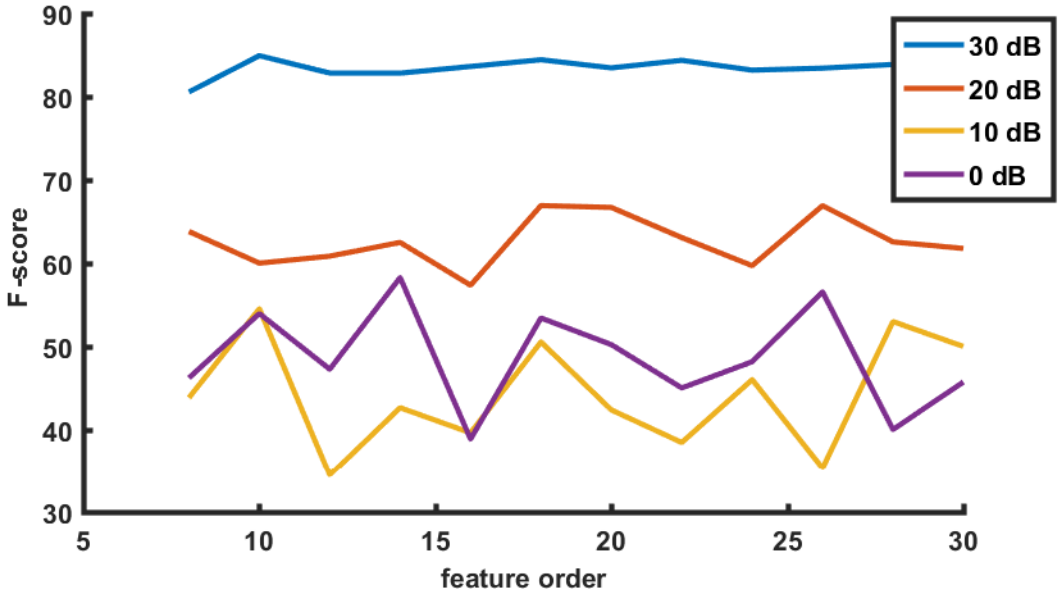

Fig. 8 F-score (3) of LPC coefficients of various orders for different SNR

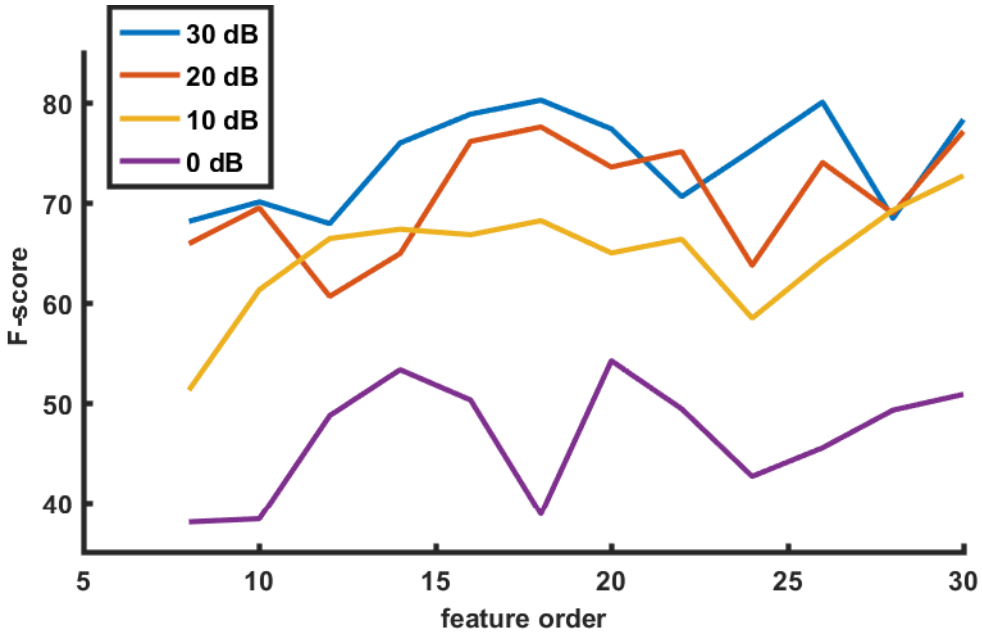

Fig. 9 F-score (3) of MFCC coefficients of various orders for different SNR

3 CONTINUOUS AUDIO EVENT DETECTION

The whole approach is based on audio signal processing in two stages, namely *Monitoring of audio events* (first stage) and subsequent *Detection of gunshots* (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.

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.

3.1 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.

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.

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 test are also described in dedicated sections, further advanced processing is described in separate chapters. Signals that are preliminarily flagged, 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

3.2 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, 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. 9. „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 9 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 %

3.2.1 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 [12]. In contrast to AMDF, this algorithm uses overlapping factor of 2/3.

Clipping level is determined by equation (7) 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 [13].

$$C_{Lj} = r_f \cdot \min\left(MAX_{j-1};\ MAX_{j+1}\right), \tag{7}$$

after clipping, resulting samples are either rounded to +/-1 or zeroed. This clipped signal is used as an input for autocorrelation. Examples of autocorrelation function for periodic and non-periodic signals are shown in Fig. 10.

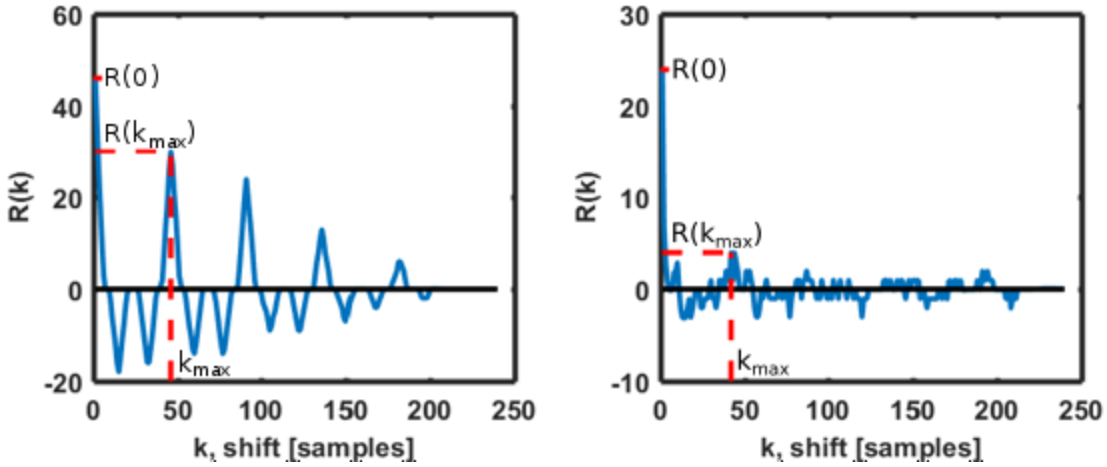

Fig. 10 Center-clipping autocorrelation output for periodic (left) and non-periodic (right) inputs

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 (8) and (9):

$$R(k_{max}) \leq \alpha \cdot R(0), \text{nonperiodic} \tag{8}$$

$$R(k_{max}) > \alpha \cdot R(0), \text{periodic} \tag{9}$$

where α is an empiric constant based on previous testing [13] 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 (13).

3.3 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 [9] 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 2.2, 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 [8] and other sounds coming from Urban Audio dataset [1] 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 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 3.1.

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 4.2.

4 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.

4.1 New Features in Time Domain

This section proposes new features derived from signal waveform. Feature testing and reported performance in this section were performed on dataset described in [14]. 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. 11 (shortening time axis for illustration purposes).

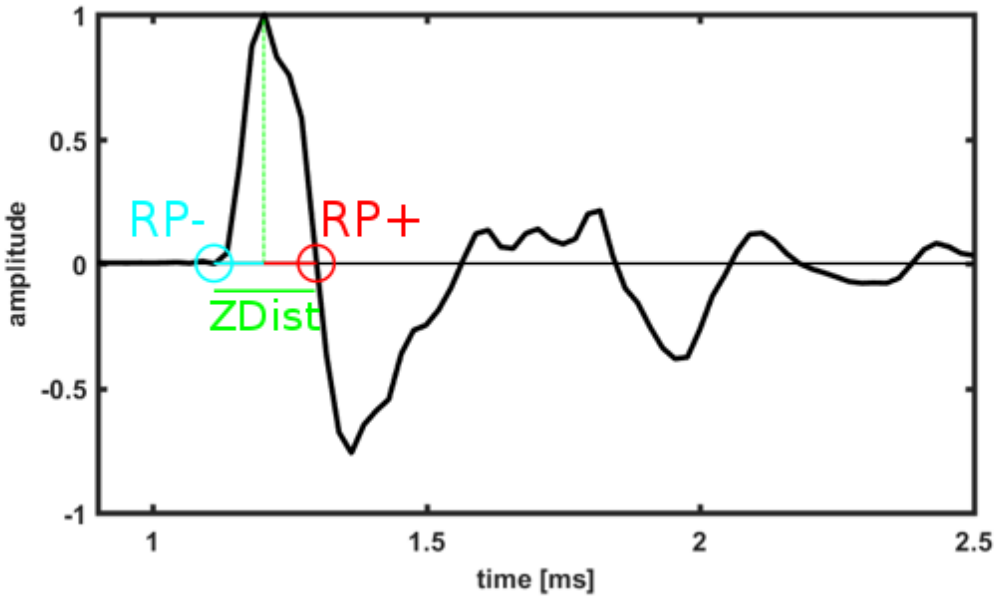

Fig. 11 Zero-crossings and their distance [14]

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. 12.

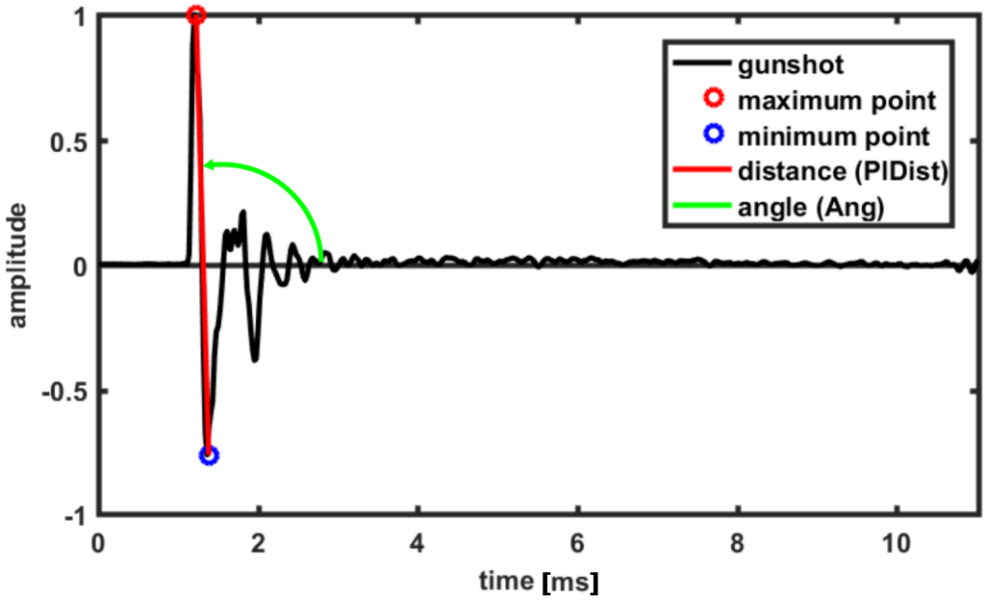

Fig. 12 Peak distance and angle [14]

The area of triangle delimited by 2 highest peaks and a minimum (referred to as "Area"). Ultimately, 4 features were defined as coefficients (A and B in (10)) of exponential fit to both positive and negative local extremes.

$$y(t) = A \cdot \exp(B \cdot t), \tag{10}$$

where y(t) is exponential approximation, A and B are coefficients and t is time. These features are illustrated in Fig. 13 together with approximations of positive envelope $p(t)$ and negative envelope $n(t)$ with numeric values for one sample waveform.

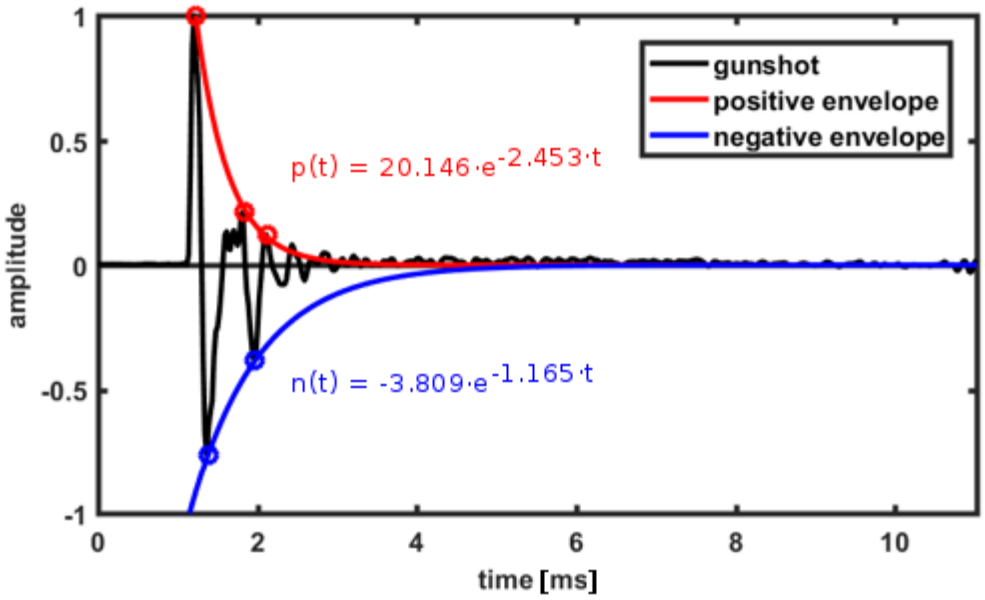

Fig. 13 Envelope approximation by exponential fit [14]

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

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. 14, Table 10 indicates recognition performance for problems „all gunshots vs. non-gunshots" and „AK-47 vs. non-gunshots".Overall, we conclude that despite 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.

Table 10 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

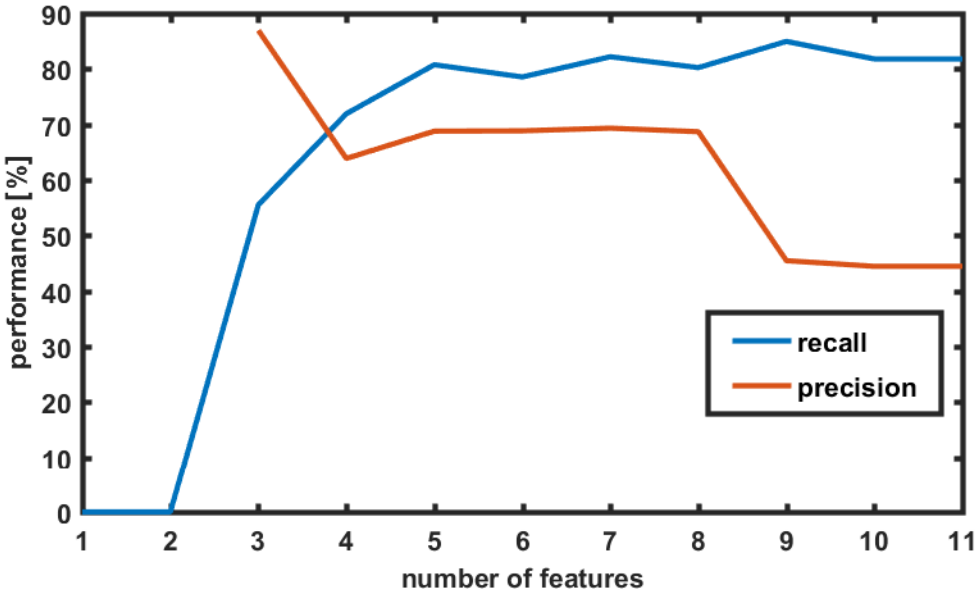

Fig. 14 Gunshot recognition performance for different number of temporal features [14]

4.2 Advanced Gunshot Detection Results

Various ideas for advanced gunshot detection were considered. One such example of state-of-the-art approach [15] 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 2.2 as well as our proposed features described in previous section. 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 (order 20) and more filter-banks (40 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. We have also used 5 best features described in chapter 4.1. 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 4.1 did not provide further improvement.

This section uses 2 different datasets coming from stage 1 tests. Dataset A contains 59723 non-gunshot frames in various classes and 1532 gunshots. The division is approximately 20% to 80% of frames flagged and not-flagged as gunshots by first stage respectively. This dataset is used for training (60% training, 20% validation) and evaluating algorithms (20% testing). Dataset B consists of 31286 non-gunshot frames (all

flagged as gunshots by stage 1) and 1532 gunshots and is used exclusively to test the final algorithm. There is minimum to none overlap in non-gunshot sounds for sets A and B, but gunshots are shared in both.

In search for algorithm suitable to handle multitude of non-gunshot classes, we have decided to take advantage of ensembling. Training single feedforward network for each non-gunshot category separately, using training subset of set A, subsequently summing resulting scores over each category and deciding based on final score. After comparison of multiple neural network architectures, we have concluded that two layers with 20 neurons each perform the best (this architecture will be referred to as NN20+20). Apart from neural networks, we have also tried other recognition algorithms, compared to selected neural network (NN20+20) in Tab. 11. Other tested algorithms, along with brief description of their hyperparameters, are listed below. The compared algorithms include the following, Support vector machines (SVM) with Gaussian kernel. 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. 11, 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. Inputs to all algorithms are standardized, so that mean value is 0 and standard deviation is 1 using data from testing data. This is done with all algorithms except for decision trees, which do not need such treatment ensuring equal scale.

Table 11 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%). Contribution of true negatives in this setup is approximately the same from each category of non-gunshot sounds.

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. 12 compares execution times of all algorithm and feature combinations. Only execution time (in seconds) of algorithms is included, features were calculated separately.

Table 12 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

In neural networks, more neurons meant longer execution time. This includes both input neurons (i.e. number of input features) and neurons in hidden layers. Regarding other algorithms, as for input features, less features mean shorter execution time. 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. 13 summarizes those results in seconds. Each extraction algorithm was calculated 5 times again and the time was averaged.

Table 13 Execution times in seconds of feature extraction algorithms

MFCC	MFCC-upsampled	LPC	TDF5
1.07	16.22	0.17	110.213

LPC coefficients, as a 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. From TDF breakdown, exponential fit features take the longest time to calculate, with the rest of the features comparable to MFCC. The idea of leaving those out would work in combination with MFCC, however using only TDF requires Bn and Bp to provide results mentioned above.

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. 14 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. 15 provides information on true positives in a similar manner.

Table 14 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 15 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.

5 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 3.2. 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.

5.1 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.

5.1.1 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 (11)

$$D(k) = \sum_{n=1}^{N-k} |s(n) - s(n+k)|, \tag{11}$$

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 (12) so that values are in range 0-1, with zero representing perfectly periodic signal. An example of similarity function is depicted in Fig. 15, with apparent period of around 50 samples. The first significant minimum (outside of zero-region where time shift is near 0) represents the periodicity degree and basic period of investigated frame.

$$R = \sum_{n=1}^{N} 2 \cdot |s(n)|. \tag{12}$$

The basic period in seconds can be calculated as follows:

$$T_0 = k_{\min} \cdot T_S, \tag{13}$$

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 $D(k_{min}) = 0$.

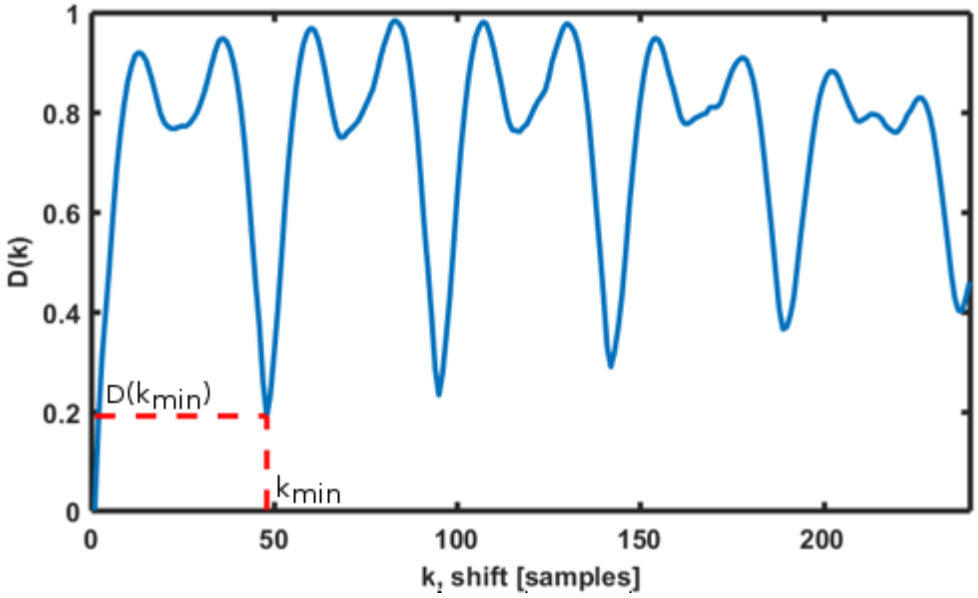

Fig. 15 Similarity function $D(k)$

5.1.2 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. 16, with mean values, minima, maxima and their differences.

Table 16 Statistics of AK-47 bursts

	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.

This was followed by comparison of the two methods (peak-search method to

AMDF) with added noise or other overlapping sounds (AWGN, rain, idling engine, barking dog, cracking branches). AMDF performed slightly worse with 20 dB SNR and completely broke at 0dB SNR with some of the sounds present. Peak-search performed 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.

5.2 Advanced Burst Detection Results

This section compares two different approaches to burst detection using previously introduced features. The first approach being based on deriving period from waveform, using AMDF and peak-search algorithm (which uses center-clipping). The second is applying individual gunshot recognition on frames and calculating signal period from Binary mask of detected gunshots.

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 the whole recordings (with any appended frames). As stated previously, the mean period of gunshot bursts have, under tested conditions, very small deviation of 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.

In terms of false alarms, the results indicate comparable performance of AMDF and peak-search in stage 2. 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. The results are summarized in Tab. 17.

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 other 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. 17 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 17 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. 17 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.

6 CONCLUSION

This work briefly summarizes all steps needed for development of succesful gunshot detection system and subsequently introduces our contribution. The first chapter introduces a number of existing audio datasets and concludes with compiling a audio dataset used throughout this work. The next step is feature extraction and feature comparison. The chapter works with various sets of frequently used features and compares their performance under different conditions. At first, compares the effects of preprocessing (such as frame length), then compares different modifications of MFCC calculation and concludes with comparison of effects of noise on feature performance. In the third chapter, the development of real-time gunshot recognition system begins. The chapter describes the whole process and reveals assorting of input into two main

categories, individual gunshots and gunshot bursts. Preliminary algorithms capable of working in real-time are, used together with features based on previous chapter. Chapters 4 and 5 dive into advanced recognition of gunshots and bursts respectively. Novel features are introduced and used in conjunction with sofisticated algorithms, achieving state of the art performance.

Firstly, we have reduced frequently used frame length of 23 ms to 11 ms based on comparison of performance with various frame lengths. 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. Comparison of feature performance under various noise conditions shows, that LPC performs better under no-noise conditions, but noise at only 20 dB SNR causes MFCC to perform significantly better.

The next part of the thesis deals with developing the gunshot detection algorithm itself. It begins with chapter 3 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 4 deals with individual gunshot detection. We combine all the methods examined from the first chapters, beginning with comparing multiple features, also incorporating newly developed feature set. 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. In comparison with other recent works on gunshot detection, our system performs significantly better, best paper in DCASE 2017 track “Detection of rare sound events” [16] 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.

SELECTED REFERENCES

[1] 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.

[2] 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.

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

[4] 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.

[5] 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

[6] 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

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

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

[13] 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

[14] 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

[15] 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.

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

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.

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.