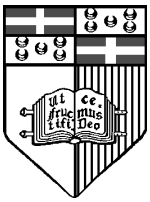
Aggression Detection in Urban Environments based on Audio Analysis

Edward Fleri Soler

Supervisor(s): Dr. George Azzopardi



Faculty of ICT

University of Malta

May 2017

Submitted in partial fulfillment of the requirements for the degree of B.Sc. (Hons.) Information Technology (Artificial Intelligence)

Abstract:

Amid escalating security concerns, improvements in technology and reductions in costs, the Electronic Security Systems market is projected to exceed US$80 billion by 2020[1]. Surveillance systems have become common-place within urban settings such as public transport systems, parks and motorways. Current equipment is restricted by the man power required to manually monitor and report activity, and is susceptible to human error. In this project, computer vision techniques are applied for the development of an automated aggression detection system for the detection of gunshots, breaking glass and screams in audio surveillance files.

Audio surveillance sensors are a cheaper, less-intrusive alternative to conventional Closed-Circuit Television (CCTV) cameras and are able to capture events with no visual counterpart. A novel approach for the detection of aggressive events in noisy, suboptimal conditions typical of urban environments is proposed. As opposed to traditional pipelines founded on the basis of low-level audio feature analysis, this proposal has been adapted to employ computer vision techniques upon time-frequency audio representations, in order to attain a high robusticity to noise.

Image features are extracted and utilized for the generation of models and the training of classifiers in order to distinguish between salient events and background noise. A stage dedicated to the localisation of events is implemented prior to classification, overcoming the need to segment and survey complete audio files.

A comprehensive dataset composed of event recordings, superimposed with complex, realistic background noise at varying Signal-to-Noise Ratios (SNR) has been acquired for the thorough analysis of the system. The robusticity of the system to high levels of varying background noise has been confirmed, cementing the applicability of such an architecture within a forensic audio analysis context.

[1]According to Global Industry Analysts, Inc.

Acknowledgements.

I would like to thank Dr George Azzopardi for his supervision and aid throughout the design and development of this system. Furthermore, I would like to thank my parents Clarissa and Jean-Paul, as well as my sister Ella, for their constant support throughout this project and my academic life.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Problem Definition

With advancements in technology, reductions in production cost and a greater demand than ever, surveillance systems are being installed in parks, squares, stations and public spaces around the world. These systems provide peace of mind to communities, aid law enforcement in the prevention of crime and promote public safety [1]. Visual surveillance, in the form of closed-circuit television (CCTV) cameras, is by far the leading means of security surveillance. However, this approach is not flawless. Visual surveillance methods may fail to identify threats in crowded environments due to the partial or total obscurity of areas of interest. Current technologies also require manual detection of threats, restricting the accuracy and cover of threat detection by the man power available to analyse footage and other sensory information. Within hectic environments, automated audio analysis is a far more effective approach, overcoming the above mentioned shortcomings.

In this project we shall explore the idea of audio surveillance in urban environments. Namely, we shall focus on aggression detection through the identification of audio signals relating to the breaking of glass, gunshots and screams. Audio systems may then replace or work together with traditional visual systems to analyse, identify and deal with threats in the most effective way possible.

The main challenge involved in the designing of this system is the identification of significant events within a noisy background. Urban environments are sporadic, crowded and hectic, with background noise from cars, buses, trains, crowds and ordinary day to day events. This system must be capable of analysing key features from the audio input to distinguish audio events from noise, correctly classifying them. Given the application of this system, a great degree of accuracy and reliability must be achieved to make it sustainable and applicable within the real world.

1.2 Motivation

The motivation behind this system is the improvement of automatic analysis systems in the fight against crime. The creation and deployment of automatic aggression detection systems would pave the way for better policing and law enforcement, taking the strain off of personnel tasked with patrolling and monitoring public areas.

The cost effectiveness and simplicity of audio sensors makes them an appealing con-

tender for visual surveillance systems. Microphones are generally cheaper to produce and require less processing power than CCTV cameras, while overcoming illumination issues, deeming them equally suitable during day and night time [2]. Furthermore, the omni-directionality of modern audio sensors trumps the limited field of view associated with visual sensors. This not only allows for the detection of events over a larger area, but also facilitates the identification of important events, such as gunshots and screams, which hold no visual signature [3].

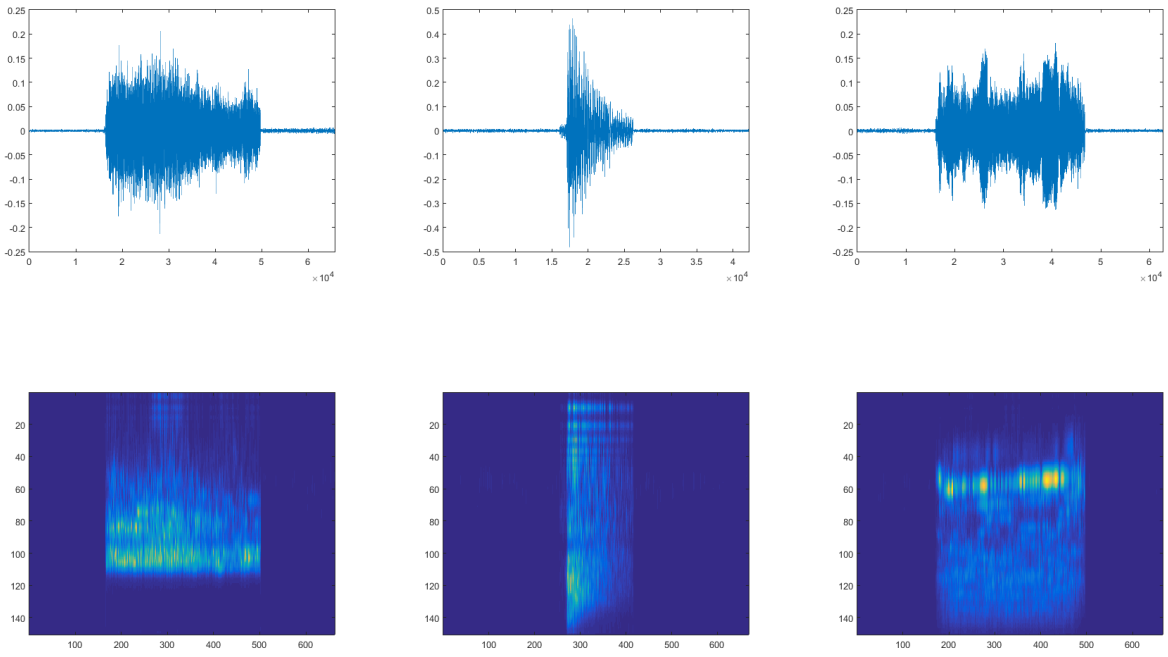
The majority of modern IP cameras, commonly used for surveillance purposes, are manufactured with built in microphones, readily forming the foundation required for the deployment of such a system [4]. The formation of a sensory surveillance network would allow for the merging and analysis of audio and video feeds, providing comprehensive coverage and superior detection of events [5].

The implications of such a system run deeper than monitoring and policing. Practitioners within the field of audio forensics are faced with the analysis, verification and interpretation of hours of audio and video tapes to be submitted as evidence on different cases to the courts. Digital signal processing techniques, including those proposed within this system, could easily be adapted to carry out such tedious, repetitive tasks, saving valuable time and resources for the prosecution during submission of evidence [6].

1.3 Approach

This system shall detect events of aggression through the off-line analysis of audio signal input from microphones at a given scene. As opposed to the traditional classification based on low-level audio characteristics, the proposed system shall follow a visual processing approach. Audio input shall first be transformed into the visual domain by means of spectrogram-like representations, as may be observed in figure 1 on the following page. A classifier shall then be trained on the image features to distinguish between events of the three different classes (breaking glass, gunshots and screams) and background noise.

The proposed system deviates from the framing or windowing approach typical in literature, applying an alternative localisation step to identify and classify temporal regions of suspected activity. By sifting through input data prior to classification, a substantial reduction of computational expenses is foreseen. This robust, intuitive pipeline could be adapted for the detection of a variety of events in an interpretative manner.



(a) (b) (c)

Figure 1: Waveform and corresponding gammatonegrams of (a) Breaking Glass, (b) Gunshot and (c) Scream

1.4 Scope

The scope of this system shall be restricted to an off-line application, suitable for the analysis and automated detection of both impulsive and sustained aggressive events within audio streams. The development of a real-time surveillance oriented system shall not be considered within this project. However, the pipeline of the proposed system could easily be adapted to take on a real-time application.

This system shall be developed and tested with the Mivia Audio Events Dataset developed by Foggia et al. in [4]. Consisting of over 28 hours of audio tapes, this dataset holds 18,000 events of aggression superimposed with background noise typical to urban areas. An assumption made within this dataset is that no two events may co-occur, as events are spread out with a rough 3 second interval. This dataset was found to be the most appealing, due not only to its nature and size, but also to its repeated use within similar applications in literature. This shall allow us to develop a comprehensive understanding and evaluation of our system in comparison with previous attempts.

1.5 Aims and Objectives

The aim of this project is to devise a system which automatically detects aggressive events in urban environments through the analysis of audio signals. This goal shall be reached through the following intermediate objectives:

1. Conversion of audio files to visual representations
2. Extraction of image features and generation of vocabulary
3. Training of classifiers on event image features
4. Localisation and classification of suspected events within test data
5. Comprehensive system evaluation

1.6 Report Layout

In the next chapter we shall investigate different approaches followed in literature to similar tasks and shall lay down the foundations for the remainder of this paper. In chapter 3 we propose the design of the system pipeline, outlining the general work flow, before detailing the implementation steps in chapter 4. Chapter 5 consists of a comprehensive evaluation of the system, including experiments, their outcomes and the motivation behind them. Finally, in chapter 6 we shall discuss the success of the project and the possibility of future alterations and improvements, prior to concluding in chapter 7.

2 Background and Literature Review

2.1 Background

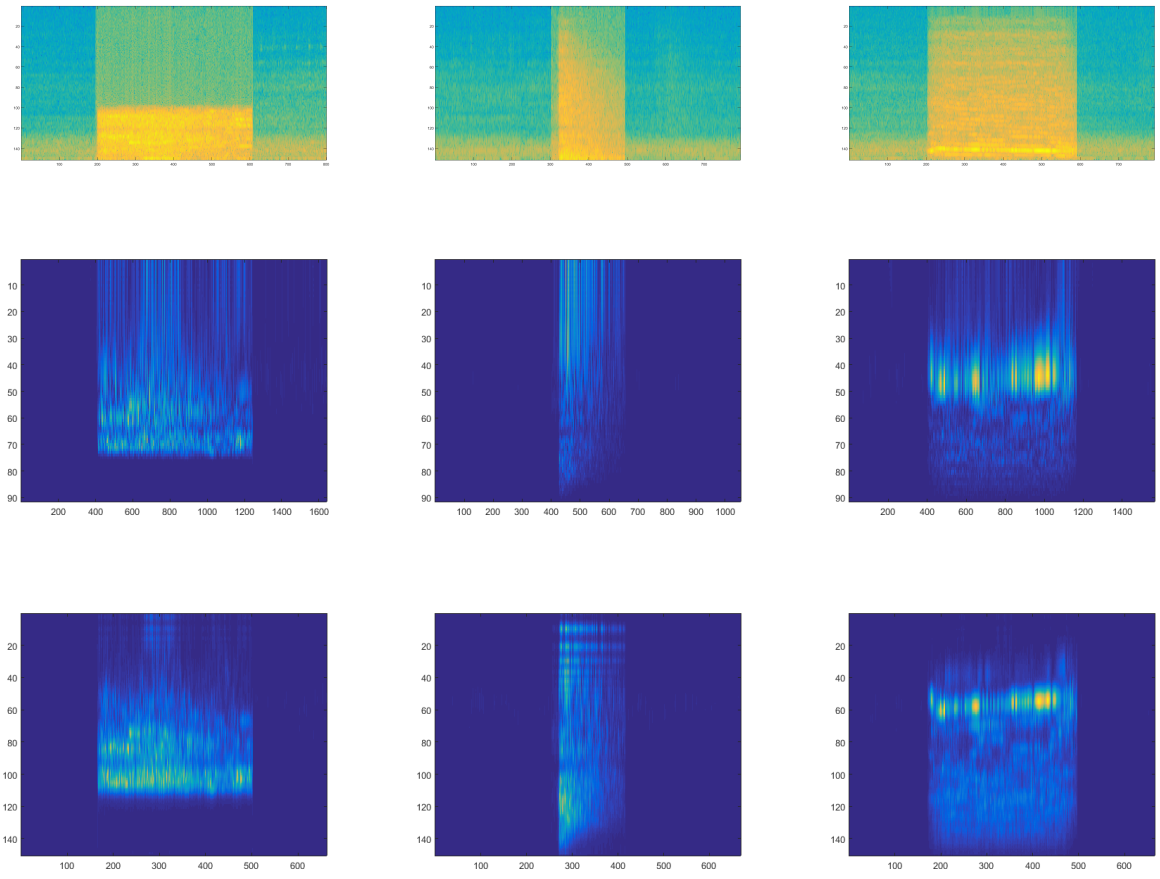
A number of references to techniques common within digital signal processing and computer vision shall be made throughout the course of this project. We shall therefore begin by shedding light onto these procedures before delving into further technicalities.

2.1.1 Audio Representation

As previously mentioned, the proposed system follows a novel approach to feature extraction: as opposed to the direct characterisation of events based on low-level audio features, audio input is first transformed into its equivalent visual representation. Spectrograms are one such visual representation, breaking down a sound at any temporal point into its constituent frequencies across the spectrum. Spectrograms are represented by means of a spectral image with time in seconds (x-axis), frequency in Hz (y-axis) and colour depicting the intensity of any given frequency band at any given time. Within the digital domain, spectrograms are generated through the application of the Short-Time Fourier Transform to audio streams, so as to compute the constituent frequencies and their intensity at any point in time.

Typically, spectrograms follow a linear-frequency scale, with equal granularity assigned to each frequency band. However, research has shown that over the course of evolution, the human ear has adapted a greater sensitivity to low-frequency sounds [7]. These findings motivated the adaptation of logarithmic-frequency scales to better mimic the sensitivity of the human ear by assigning higher granularity to lower frequency bands.

Further motivated to mimic the processing and perception of sounds by the human ear and brain, in 1987 Patterson et al. [8] developed the Gammatone-Filterbank; a log-frequency spectrogram representation which linearly approximates the auditory filtering carried out by the human ear. Observing figure 2 on the following page, the logarithmic stretching of the frequency axis is clearly visible for both the logarithmic-frequency spectrogram and the gammatonegram assigning greater focus to lower frequencies. In chapter 5, a comparison between each of these image representations and their effectiveness at defining audio features shall be made.



(a) (b) (c)

Figure 2: Respectively, linear-frequency spectrogram, logarithmic-frequency spectrogram and gammatonegram of (a) Breaking Glass, (b) Gunshot and (c) Scream

2.1.2 Feature Extraction

Spectrograms, like all other images, are comprised of a matrix of pixels, each assigned a colour value. The vast number of pixels together with the set of all possible pixel values deems images to be somewhat continuous in nature. This is problematic in a mathematical and digital context, as discrete data is favoured for the automation of processes. Feature extraction is the process of generating image descriptors based on the image contents and characteristics. These standard, reproducible descriptors allow for the comparison of image characteristics in a quantifiable manner, taking us a step closer to the direct comparison and relation of images as a whole.

Salient points of description, known as keypoints, are detected and extracted from images through the application of an automated algorithm such as the Scale Invariant

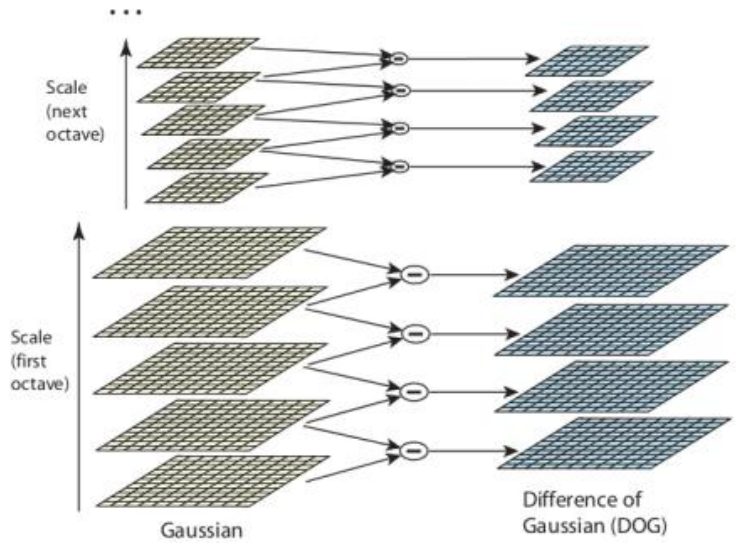


Figure 3: Difference of Gaussians procedure

Feature Transform (SIFT) [9]. This algorithm searches for local extrema, suitable to act as keypoints, through a process known as the Difference of Gaussians. This process involves the repeated application of Gaussian blurs of different intensities to images across a scale space. The difference between the neighbouring Gaussian blurs of same scale is taken, resulting in the final set of Difference of Gaussians, as shown in figure 3 above. These 3-dimensional data structures are then sifted for local maxima, which will serve as feature descriptors. The actual descriptor is computed as a histogram of pixel gradient orientations, and typically exists near edges, corners and areas of texture.

2.1.3 Bag-of-Words

Spectrogram features extracted till this point are known as low-level features as they are of very fine and specific detail. These features do not make much sense in their raw form, and require further abstraction to a higher level of representation. This is where the Bag-of-Words model is adopted.

The term 'Bag-of-Words' was first keyed in a linguistic context by Harris in [10]. This model was used as a simplified descriptor of documents based on the construction of a bag (or dictionary) holding all words present within the document. Interesting analytics, such as term frequency, may then be computed with respect to this bag, generating a description of the document in question.

This model found its application in the representation of images for classification in

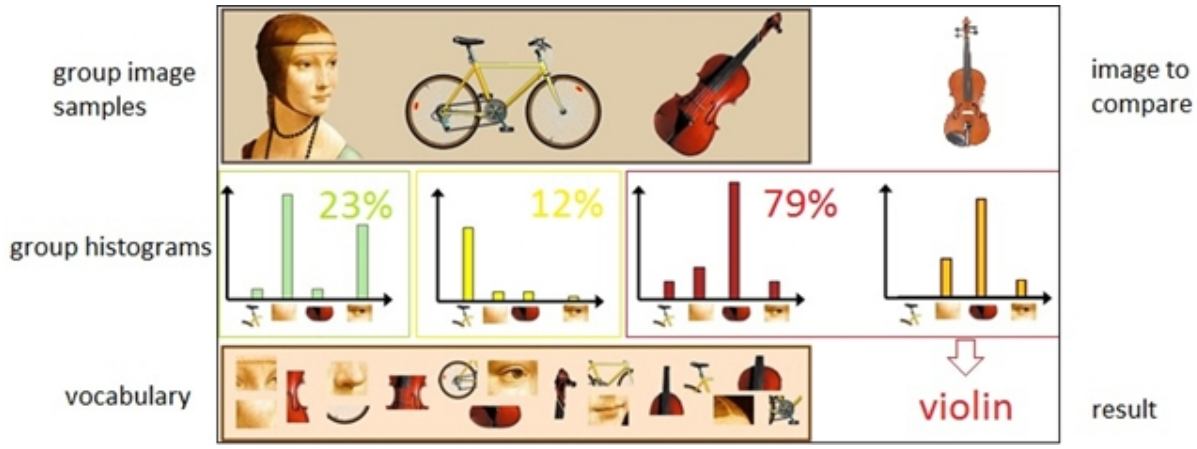


Figure 4: Example of the Bag-of-Words model applied to image classification

[11]. Figure 4 illustrates the proposition of Yang et al., beginning with the creation of a dictionary known as a vocabulary, to hold K visual words generated through the clustering of low-level image features. Following the creation of the vocabulary, each test image feature is associated with its closest visual word within the vocabulary. The result is a histogram descriptor of size K storing vocabulary word frequencies for each image. A classifier may then be trained upon these histogram-of-words, benefiting from the fact that each image holds a fixed-length descriptor [12], with the descriptor length K being a parameter adjustable to suit different applications. Similar representation models, such as the Vector of Locally Aggregated Descriptors (VLAD), produced by Jegou et al. [13], have found application within literature. However, the Bag-of-Words model is regularly selected as the model of choice within computer vision oriented applications, given its simplicity and effectiveness.

2.1.4 Classification

Within the context of computer science, and specifically machine learning, a classifier is a learning model trained to assign one out of a finite set of labels to an unseen example, based on readily available training data. All classification techniques require training data to act as a metric of comparison with unseen cases. Different mathematical models are then employed to compute and assign the most reliable classification from the set of labels.

One of the earliest adopted classification algorithms is the K-Nearest Neighbour. This supervised algorithm maps all training instances onto a search space together with their pre-assigned labels. Any unseen cases are mapped onto the same search space based on

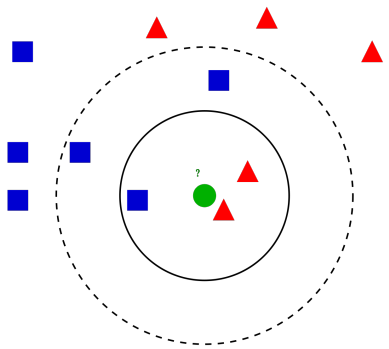


Figure 5: Application of the K-Nearest Neighbour algorithm

the data descriptor. A search for the closest K neighbours within the search space is then conducted, with the majority label being assigned to the unseen case. Figure 5 illustrates the application of the algorithm in the classification of a green circle. With a neighbourhood size of $K = 3$, the circle will be classified as a red triangle, however with neighbourhood of $K = 5$, it would be classified as a blue square. An appropriate neighbourhood size K must thus be tuned so as to avoid over-fitting and over generalisation. This intuitive algorithm has been widely adopted for a number of classification tasks, but has since been overshadowed by more effective classifiers.

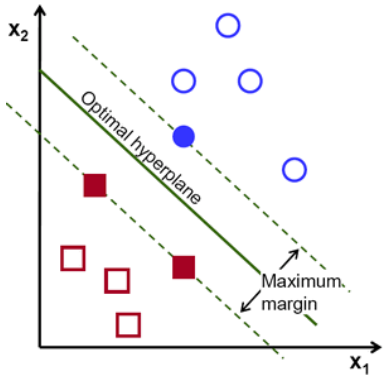


Figure 6: The goal of the SVM classifier is to find the optimum hyperplane within the search space holding the greatest possible margin

Support Vector Machines (SVM) are a two-class classification model widely adopted in literature [2, 3, 4, 11, 14, 15, 16, 17] due to their higher level of accuracy and ability to model non-linear decision boundaries [12]. SVMs are fed labelled training data from two different classes, mapping them onto a search space based on their descriptors. The most optimum hyperplane dividing the two classes in a mutually exclusive fashion is then computed so as to produce the maximum margin bound possible on either side of the

hyperplane, as may be observed in figure 6. Unseen test cases are then mapped onto this same search space, with a label assigned respective to the side of the hyperplane on which they lie. A transformation known as a kernel may be applied to the search space of the SVM, mapping all points onto a new hyperspace, allowing for the modelling of non-linear decision boundaries. The motivation behind this procedure is the procuration of a hyperplane within a hyperspace with a maximal margin, therefore achieving the most reliable classification results. The complex, multi-dimensional hyperspace generated by the proposed system would be more suitably classified by such an approach. The classification structure adopted together with the most suitable kernel choice shall be detailed in chapter 4.

2.2 Literature Review

The vast field of audio analysis covers a multitude of applications, and has cemented its standing as a mature branch of research within computer science. Over the last half century, ground breaking feats within this discipline of study have been achieved, with its relevance greater now than ever before. In this section, an overview of audio analysis applications and techniques adopted in literature shall be presented, providing motivation and insight into this domain.

Initially, audio analysis secured its roots as a useful field of research through inspiration from the entertainment and multimedia industries. With the explosion of audio and visual media, the benefits of applying digital signal processing techniques to such a popular realm were realised, with mounting interest leading to rigorous research and testing. Cai et al. provide a setting in which these techniques may be utilised in [18]. They propose the application of audio analysis in the form of event detection for video summarisation. Within different settings, the occurrence of certain audio events signify a certain state, which may not be detectable through visual analysis. In this paper, a baseball game is used as an example scenario to demonstrate the superiority of audio over video analysis, whereby a sudden burst of cheers from the crowd may signify the hitting of a home run; a challenging event to detect visually.

One branch of audio analysis is automatic content analysis and retrieval [19]. This subdivision is concerned with the generation of a standardised abstraction for the description and relation of audio files. Audio fingerprinting [20] involves the summarisation of audio recordings, supporting the comparison, indexing and retrieval of audio files. In [21], Foote proposes a system capable of retrieving audio documents, including music, based

on a similarity measure derived from these audio characteristics. A heavily researched application within automatic content analysis and retrieval is automatic music genre classification. Scaringella et al. [22] break down the various approaches taken to achieve this goal, providing an overview of the state-of-the-art at the time.

Speech recognition is most likely the single most explored branch of audio analysis, with a comprehensive research background dating back to the mid-1950s. The majority of modern approaches to this problem are now inspired by deep learning [23]. However, a number of signal processing techniques devised for, and applied to speech recognition have spilled over into other domains of audio analysis.

Other than for commercial and entertainment purposes, audio analysis has found application in a variety of fields, such as surveillance and forensics. In [24], Maher et al. explore the application of digital signal processing techniques to audio recordings, such as 911 calls, for the deciphering of gun shots. Advanced processing techniques would allow for the identification of firearms [25] as well as the rough localisation of the shooter and target through the detection of muzzle sounds and bullet ricochets respectively.

With the improvement of digital editing software, the manipulation and falsification of audio evidence poses a great threat to law enforcement and prosecutors in the fight against crime [6]. In [26], the proposition of audio record authentication through the analysis of acoustic environments is made. Malik et al. develop a new approach for the scrutiny and validation of audio evidence by considering both vocal and non-vocal characteristic acoustic features. This system is capable of distinguishing between natural and synthetic sounds, identifying regions of suspected manipulation.

With heightened terrorist threats and improved efforts in the fight against crime, a great deal of focus has been assigned to public safety through advanced security. Surveillance equipment to the likes of CCTV cameras and audio sensors are in abundance in developed countries, and require a transition towards automatic monitoring. Areas densely frequented by the general public, such as transport networks, shopping centres and parks are the focal point of new automated monitoring schemes. The greater majority of audio analysis systems follow a pipeline based on the direct extraction of audio features and characteristics. Although this traditional procedure deviates from the proposed system, it is still wise to evaluate and consider these techniques in preparation for the implementation of the proposed pipeline.

One of the earlier efforts for the automated detection and handling of abnormal situations was made by Clavel et al. in [27]. In this paper, the authors describe their vision of

a multimedia event detection system, capable of detecting a wide range of abnormal situations including natural disasters, aggression and intimidation through the employment of multiple modalities. The authors then proceed to focus on one class of events: gunshots. The procedure begins by segmenting the audio input into partially overlapping frames of 20ms each. Of the set of audio features commonly employed within event detection systems [3], this system exploits characteristic energy, spectral and Mel-Frequency Cepstral Coefficient (MFCC) features, generating a feature vector for each frame. Gaussian Mixture Models are then applied to form a representative model for each class of gunshots. Finally, the Maximum A Posteriori decision rule is applied to 0.5s segment windows, classifying each segment as holding a specific class of gunshot, or background noise.

[16] follows a similar approach to tackle shout detection within public transport vehicles, with a slight twist in the pipeline akin to that of our proposed system. Prior to the extraction of audio features, a segmentation algorithm is applied to the audio input so as to locate areas of suspected interest. Focus is then restricted to said temporal segments, greatly reducing the computational expense. Alternative to the typical application of GMMs, a hierarchical SVM model is adopted to classify a segment as background noise, vocal or shout in a stratified manner.

In [28], Valenzise et al. propose the fusion of an audio detection system with visual surveillance equipment for the automated steering of CCTV cameras. The detection module follows a pipeline analogous to the above mentioned systems, with the added functionality of source localisation. An array of microphones (usually 4 or more) are installed in a predefined formation. Depending on the source location of the detected event, a quantitative delay within the audio feed will be experienced by each microphone. In a process known as time delay estimation, the angular source of the event may be computed to an accuracy of around one degree, allowing for the automated steering of nearby CCTV cameras.

More recently, the suitability of the Bag-of-Words model for the high-level representation of event sound characteristics has been realised. Carletti et al. generate a dictionary and Bag-of-Features model based on low-level audio features for gunshots, breaking glass and screams in [29], while Foggia et al. similarly employ the Bag-of-Words model in their road surveillance system [2]. Both papers acknowledge the obstacle of handling impulsive events, such as gunshots or car crashes, and longer, sustained events, such as screams or tires skidding. Foggia et al. reiterates the need to deal with both event types in [4]. Here, a system dedicated to the detection of audio events in highly noisy environments is devel-

oped and evaluated. This system follows a windowing approach based on low-level audio features, with a great deal of stress placed on the selection of an ideal window size. The focus of this system is to achieve a high level of robusticity, as would be required by a surveillance system within an urban context. A custom dataset superimposing gunshots, breaking glass and screams with background noises of different styles and varying levels was created for the purpose of evaluation, and has been selected as the ideal dataset for the proposed system. A direct comparison with this system shall be made during the evaluation stage in chapter 5.

The differences between audio events, such as gunshots, and phonetic structures, such as speech, are well illustrated by Dennis et al. in [14]. The authors proceed to mention how audio events often have short durations but hold a more distinctive time-frequency representation than speech. This provides motivation for the adaptation of spectrograms and other visual representations to audio event detection systems, as opposed to the use of techniques originally employed within speech analysis. One clear advantage of introducing time-frequency representations is that the whole frequency spectrum of an audio file is considered, as opposed to the typical filtering and band-focusing involved in conventional techniques.

While visual approaches to audio event detection have not been heavily pursued, the results attained to this point seem promising. Dennis et al. propose a system capable of identifying overlapping sound events in [15]. The novelty of this system lies in the use of spectrogram images for the characterisation of events. The Local Spectrogram Features (LSF) algorithm searches for maxima within the spectrogram by applying a cross-filter to search for points of maximum intensity within the current frequency band and temporal point. These maxima are extracted as keypoints, storing both log-frequency power as well as geometric data for the training and detection of events. This procedure achieved an average accuracy rating of 97% on a dataset of overlapping sounds consisting of horns, phones, bells and other typical environmental sounds.

In [17], Foggia et al. take inspiration from the human perceptive system while developing an audio event detection system based on the gammatonegram. Here, motivation for the suitability of the gammatonegram over other visual representations is justified by the processing undertaken by the cochlea membrane within the inner ear, together with the brain, in converting a sound into an auditory image. Haar basis functions are applied to extract features from the gammatonegram for classification by means of the AdaBoost learning algorithm. Considering the final precision rating of 96.2%, this system yet again

confirms the potential of a visual approach to an audio event detection system, motivating and influencing the design of the proposed system.

3 Design

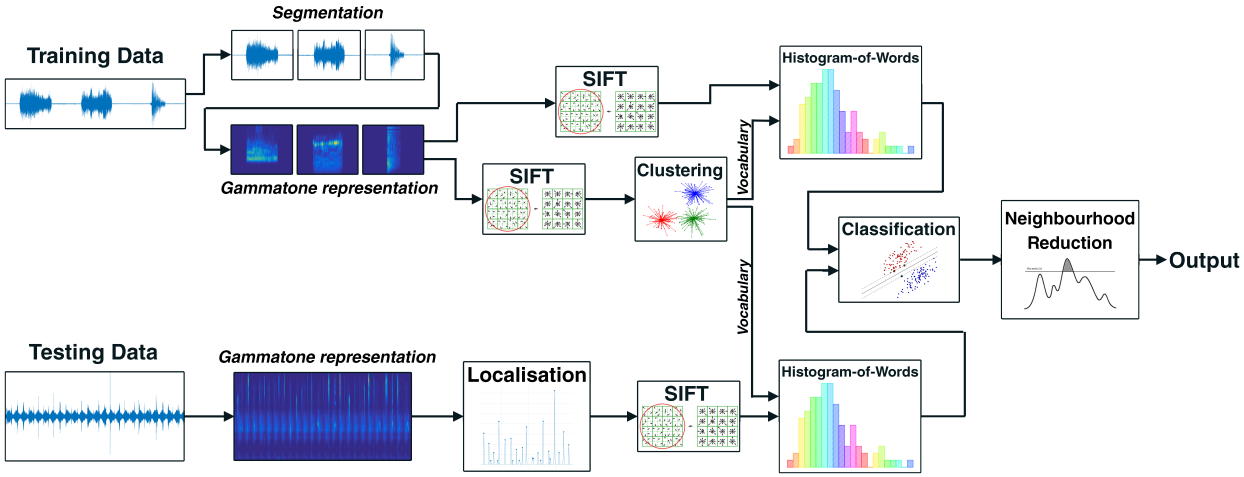


Figure 7: Pipeline of the proposed system

3.1 Data Handling and Audio Representation

Figure 7, above, summarises the main procedures involved in the proposed pipeline. Considering event detection to be a learning problem, the first concern of this pipeline is the preparation of training data. The dataset considered consists of a number of events of interest superimposed with background noise, together with meta data for event start and end times, event class, background class and SNR level. Firstly, the training data is segmented, generating an organised set of event and background noise snippets tied to their respective meta data. Next, each snippet is transformed into its respective time-frequency representation. As previously discussed, different representations yield different results, deeming the image representation parameter to be an important one. The effects of the different representation schemes will be discussed during the system evaluation.

3.2 Feature Extraction and High-Level Representation

Once the training data has been organised and translated to the time-frequency domain, feature extraction commences. The SIFT algorithm [9], routinely used within literature, is implemented for the extraction of image features, known as keypoints. A set of keypoints are extracted from each snippet, including background noise, before undergoing clustering

through the K-Means algorithm to form a dictionary of size K. The resulting clusters form the code book on which all image descriptors shall be based from this point on.

The final stage in the preparation of the training data is the generation of image descriptors for each snippet. The incorporation of spatial data is of key importance at this point, as each class of events holds a characteristic energy signature at specific temporal and spectral regions. Spatial pyramids are therefore employed so as to relate each keypoint to a spatial zone within the image. The SIFT algorithm is once again implemented to extract keypoints from each of the spatial zones, before abstracting these features into histogram-of-words, as dictated by the Bag-of-Words model. The result is a set of histogram-of-frequencies of size K, equal to the defined dictionary size, for each zone within the spatial pyramid, acting as a descriptor for each of the snippets.

3.3 Localisation

As opposed to the typical windowing method, the proposed pipeline follows a hierarchical detection approach, locating suspected events prior to classification. The localisation stage was inspired by the notable energy signature across the spectrum left by all events, even in unideal, noisy circumstances. Figure 8, on the following page, depicts a typical test audio file, which the proposed system would expect, readily mapped onto the time-frequency domain. At three minutes long, it consists of 33 events across the three classes, superimposed with background noise of cars at a SNR level of 10dB. One may clearly observe the distinct signatures left by each of the events in the form of a vertical energy band across a wide range of frequencies.

Localisation is performed by applying a line detection algorithm tuned to respond to vertical lines of a parameter-defined width. This algorithm returns an energy response in regions holding vertical lines, while ignoring non-linear textures or lines at other orientations. As may be observed in figure 9, the result is a filtered rendition of the gammatonegram, allowing for a cleaner, more accurate detection of suspected events.

Finally, the response is summed across the frequency domain, producing a plot of total response at each temporal unit. Peaks within this plot are taken to be the center point of suspected events, while the troughs on either side are respectively taken to be the start and end times. Now that localisation has been completed, all that remains is the assignment of labels to each of these temporal bands.

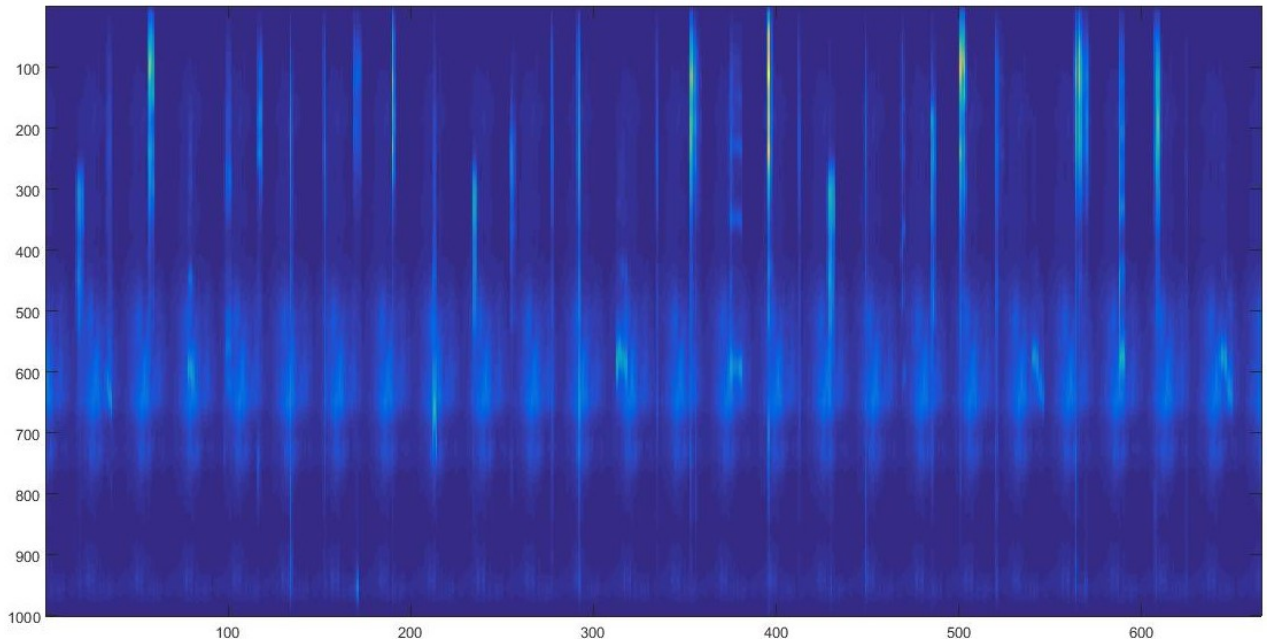


Figure 8: Gammatonegram of audio file consisting of 33 events at a 10dB SNR level

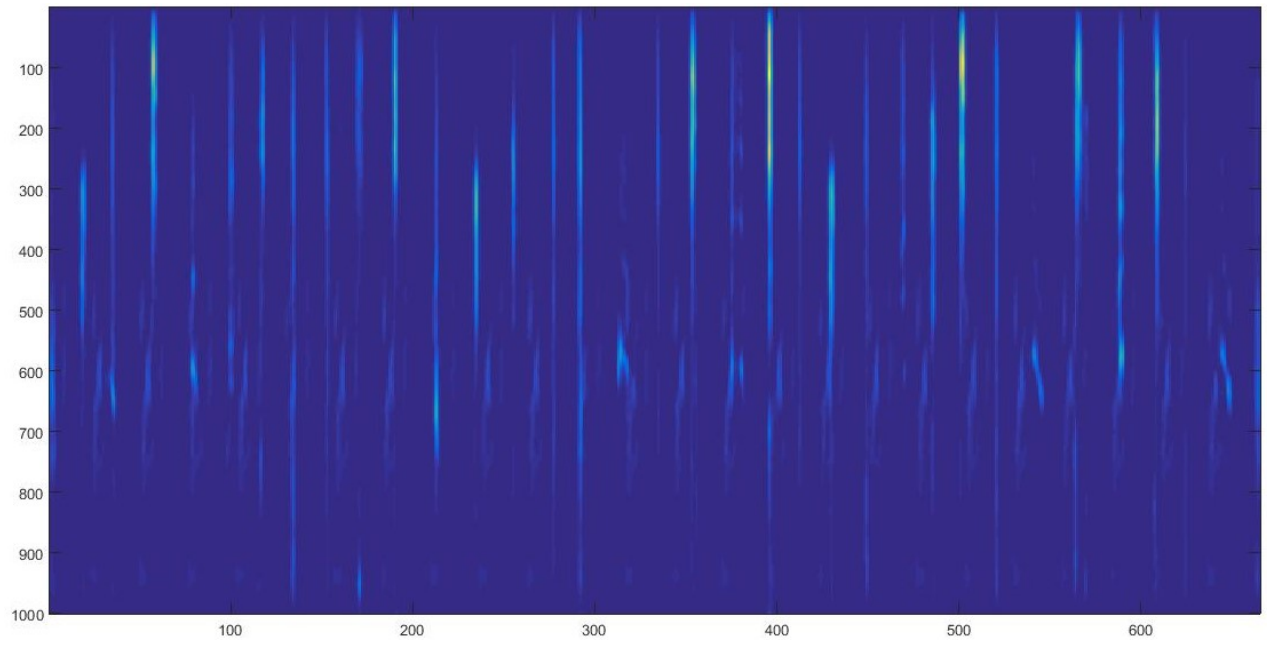


Figure 9: Response of the vertical line detection algorithm applied to the audio file gammatonegram

3.4 Classification

The final stage of the pipeline concerns the classification of the suspected events extracted during the localisation phase. Similarly to the training data, the test audio files undergo segmentation respective to the suspected events' start and end times. A histogram-of-words is computed for each of these segments in the exact same manner as described for the training data. Now that both the training and testing data are represented in the same mode, SVM classification takes place, assigning each suspected event to one of the three event classes, or alternatively labelling it as noise.

A process known as neighbourhood reduction is applied to analyse the localisation and classification scores of each of the suspected events, marking any suspected events below a threshold as noise. The final predictions are compared to the ground truth during an evaluation stage, computing statistics relating to the accuracy and success of the system.

4 Implementation

This section deals with the implementation of the above mentioned designs. Fine details relating to the various procedures, techniques and optimisations adopted, as well as the logic behind implementation decisions, shall be discussed. The MATLAB programming language and IDE was found to be the most suitable environment on which to develop this system, given the primary role which computer vision techniques play within the pipeline.

4.1 Dataset Acquisition

The central goal of the proposed system is the detection of aggressive events in urban environments; specifically within noisy, sub-optimal conditions. The system must therefore be tested with a range of levels and types of background noise, so as to obtain a clear understanding of its strengths, weaknesses and capabilities. The MIVIA Audio Events Dataset [4] developed by Foggia et al. was thus found to be the best adapted for such a role. Developed as a custom dataset for the testing of a similar automatic event detection system [4], this dataset focuses on the organisation of audio files with respect to background noise levels, and will insightfully allow for the direct comparison of the proposed pipeline with another system.

Each audio file, consisting of a mixture of events superimposed with different background noises, exists at SNR levels ranging from 5dB to 30dB, at 5dB intervals. The higher SNR files depict ideal conditions in which such a system may operate, while the lower SNR files test the robusticity and applicability of such a system within an urban context. The comprehensive meta data provided together with this dataset allows for a thorough evaluation of the system, considering the effects of different background noises and the systems efficacy at detecting different aggressive events. Prior to the segmentation of the training audio, the meta data was extracted from xml files[2] and stored within structs together with the segmented audio.

4.2 Audio Representation

As previously discussed, three time-frequency representations shall be considered during the evaluation of this system: linear-frequency spectrograms, logarithmic-frequency spectrograms and gammatonegrams[3]. All three representations required parameter tuning with

[2]XML Toolbox script used for meta data extraction by Marc Molinari [30]

[3]Scripts used for logarithmic-frequency spectrogram and gammatonegram by Daniel Ellis [31, 32]

respect to the frequency and temporal range, windowing methods and overlap percentages. A trade-off between temporal and frequency granularity had to be reached, to allow for accurate spectral representations, while maintaining reasonable temporal accuracy.

Initially, larger, higher quality representations were considered to be suitable, due to the high level of detail involved. However, following further development of the system, the computational load brought about by the large data variables was apparent. The prominent features being extracted to describe each snippet did not relate to fine edges or textures. Instead they relied on the general detection of energy intensities within specific spatial domains. A reduction in the quality of the visual representations therefore brought about significant improvements in efficiency, with negligible reduction in accuracy.

4.3 Vocabulary

The vocabulary, or dictionary, used when implementing the Bag-of-Words model, has a direct effect upon the accuracy of the system, as it defines the distinct set of descriptors which are used to express each image. The SIFT algorithm[4] is employed to extract features from the training data set for clustering into a vocabulary. Initially, traditional SIFT was adopted for the extraction of features, applying the Difference-of-Gaussians method to identify suitable keypoints. Dense SIFT skips the identification stage by instead applying a step function, to iteratively traverse the image in question, extracting keypoints at set intervals. This procedure suffers from the selection of slightly sub-optimal keypoints, but is far more efficient than the typical approach, and was therefore considered for testing. Following validation, dense SIFT was found to be suitable, resulting in no noticeable reduction in accuracy. Following extraction, keypoints were normalised, converting all values to lie within a range from 0 to 1. This procedure is often practised within literature, and was found to improve validation accuracy slightly.

Another parameter with direct effects on the accuracy of the system is the dictionary size K. When considering a suitable value for this parameter, one must take into account multiple variables, including the size of the dataset, the low-level features extracted and the image content. Low values of K typically result in high generality, due to the inability to generate complex expressions based off a limited wordset. Increased dictionary sizes allow for a greater level of specificity, but can often result in over-fitting. The dictionary size has a direct correlation with the descriptor sizes for each image, as a vocabulary of size K results in a histogram of K word frequencies, thus leading to excessive computation with

[4]SIFT, K-Means and K-D Trees, among other functions, were imported from the VLFeat Library [33]

large vocabularies. Research has also shown that, when spatial data is incorporated into image descriptors, a smaller vocabulary suffices, maintaining performance while reducing computation [11]. For this reason, a vocabulary size of 50 was adopted throughout the development and debugging of the system, while larger vocabularies were considered for the final evaluation.

Considering the vast size of the dataset, a voluminous amount of features were extracted and stored for clustering of the dictionary. This resulted in an execution time of over 70 hours to perform clustering on the full training feature set. However, as pointed out by Foggia et al. in [4], down-sampling of the feature set by a factor of 2, results in a considerable improvement ($\sim 90\%$) in computation time with no effects on the accuracy. Therefore, a simple 'coin-flip' statement was included in the code to reduce the feature set to half its initial size, reducing the K-Means execution time to roughly 6 hours without affecting accuracy.

4.4 High-Level Representation

Now that a suitable vocabulary has been generated, a high-level description of each image may be constructed. As previously mentioned during the design phase, the extraction of spatial data is of paramount importance when performing classification. Yang et al. demonstrate the notable effects brought on by incorporating spatial information during the classification of scenes in [11]. Taking the example of an image of a beach, one would expect to find features relating to sand in the bottom half of the picture, while features describing the sea or sky would exist in the top half of the picture. The manner in which this extra data would aid in classification is thus immediately apparent. For this reason, a great deal of attention was assigned to this stage of the system, so as to adopt the most effective encoding of spatial data.

Spatial tiling is a widely adopted technique, and involves the splitting of an image into a grid (usually 2x2, 3x3 or 4x4). Features are then extracted from each of the cells and are abstracted to a histogram-of-features before being appropriately encoded. A more complex procedure is that of spatial pyramids. This strategy incorporates multiple spatial tiles of varying granularity into one encoding. On the following page, figure 10 demonstrates the multi-level structure of the spatial pyramid. At the top of the pyramid, we consider a simple 1x1 grid, whereby a set of features from the image in question are extracted with no encoding of spatial information. Moving to the next level, we now have a 2x2 grid, resulting in a histogram-of-features being computed for each cell, as is repeated in the final level.

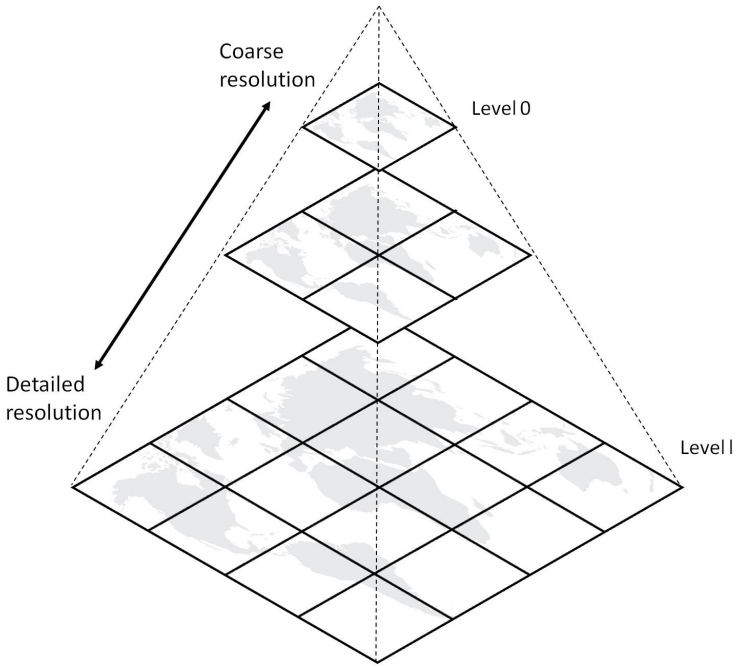


Figure 10: Spatial pyramids incorporate multiple spatial tiles of different granularity into one encoding

With each descent, the number of cells increases, thus reducing the area covered by each cell, in-turn increasing the spatial specificity. The result is a vector of size K representing the top-most level, a vector of size 4K representing the intermediate level, and a vector of size 16K representing the bottom level; where K is the dictionary size.

The final stage involves the assignment of weights to each of the levels to reflect the spatial precision of each grid. Equation 1 below outlines the approach to weight assignment for each level. Referring to figure 10 again, the resulting outcome would be a weight of 1 being assigned to the bottom most level, 0.5 to the middle level and 0.25 to the top level, assigning a great deal of importance to the spatial region in which a feature exists.

$$Weight = \frac{1}{2^{current\,level - total\,levels}} \tag{1}$$

$$Distance(a, b) = \sqrt{\sum_{i=1}^{n} (b_i - a_i)^2} \tag{2}$$

The process of mapping keypoints onto words to produce a histogram-of-features involves the calculation of distances between the current keypoint and each word in the dictionary. This is typically computed through Euclidean distance and was initially im-

plemented in this manner. Observing equation 2, the Euclidean distance is computed by taking the root of the sum of square differences of all dimensions n between any two points. However, a multi-dimensional partitioning data structure known as a K-D Tree was found to optimise the search process for the closest word, thus replacing the Euclidean distance procedure. Following the relation of each keypoint with a word, the histogram-of-features for each cell were populated with frequency scores. Ultimately, each audio snippet was represented by a vector of length respective to the number of pyramid levels and vocabulary size K. At this point, Yang's et al. arguments put forward in the previous subsection regarding vocabulary size with spatial data is realised, as a large dictionary size K would result in an oversized descriptor vector for each audio segment. A comparison of different spatial encodings, together with different vocabulary sizes K, is made in chapter 5.

4.5 Localisation

The localisation stage of the pipeline is what differentiates this approach from all other known approaches in literature. This stage of the implementation is designed to work in an off-line non-real-time manner, by accepting test audio files for detection. Considering this to be the first in a two part hierarchical approach to event detection and classification, a considerable accuracy must be achieved, as any events which go undetected at this stage will add to the false rejection rate.

Gammatonegram representation was found to be the ideal image representation scheme for localisation, as it produced the most noticeable event signatures. The representation was once again tuned to achieve an ideal compromise between temporal and spectral resolution, taking into account that the audio files in question were now far longer than the audio snippets previously dealt with. A line detection algorithm, based upon trainable B-COSFIRE filters [34] was adapted to detect vertical lines within the gammatonegrams. A combination of experiments with respect to the width of the line detector were undertaken so as to achieve the optimal detection accuracy. A single filter detecting lines of 2 pixels in width at a search length of 10 pixels was found to yield the best results, handling both narrow, impulsive events as well as wider, sustained events.

As demonstrated in figures 8 and 9 in the previous section, the application of the line detection algorithm resulted in a representation similar to that of the initial gammatonegram, but with a considerable reduction in noise energy. Figure 11 illustrates the summed response of the line detection algorithm on a test audio file holding 33 events. Spikes in the plot relate to the detection of a vertical line, marking the position of a suspected event.

As may be observed, far more than 33 peaks have been detected. The motivation is that, at this stage, a set of temporal regions suspected of holding an event are marked, prior to classification. During the classification stage, a number of these suspected events will be deemed to be noise. This overcomes the computational cost tied to the windowing method, which requires the analysis of the complete audio file.

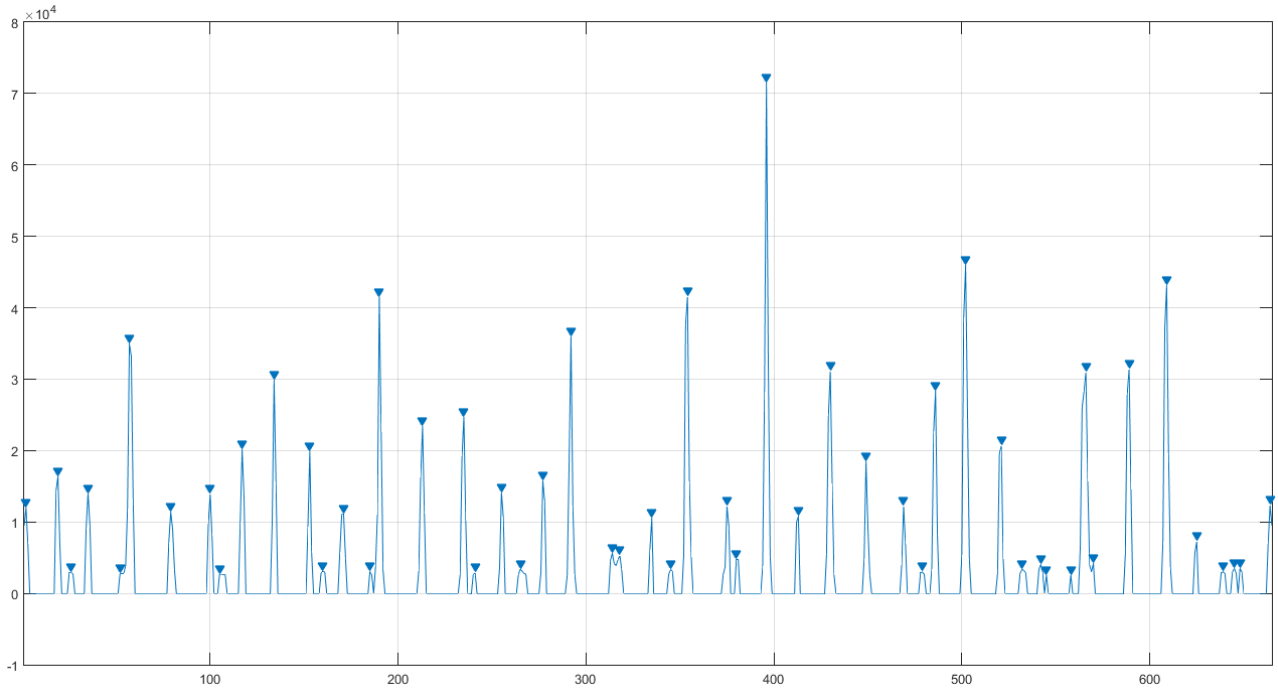


Figure 11: Summed response of the B-COSFIRE vertical line detection algorithm on a 10dB SNR audio file

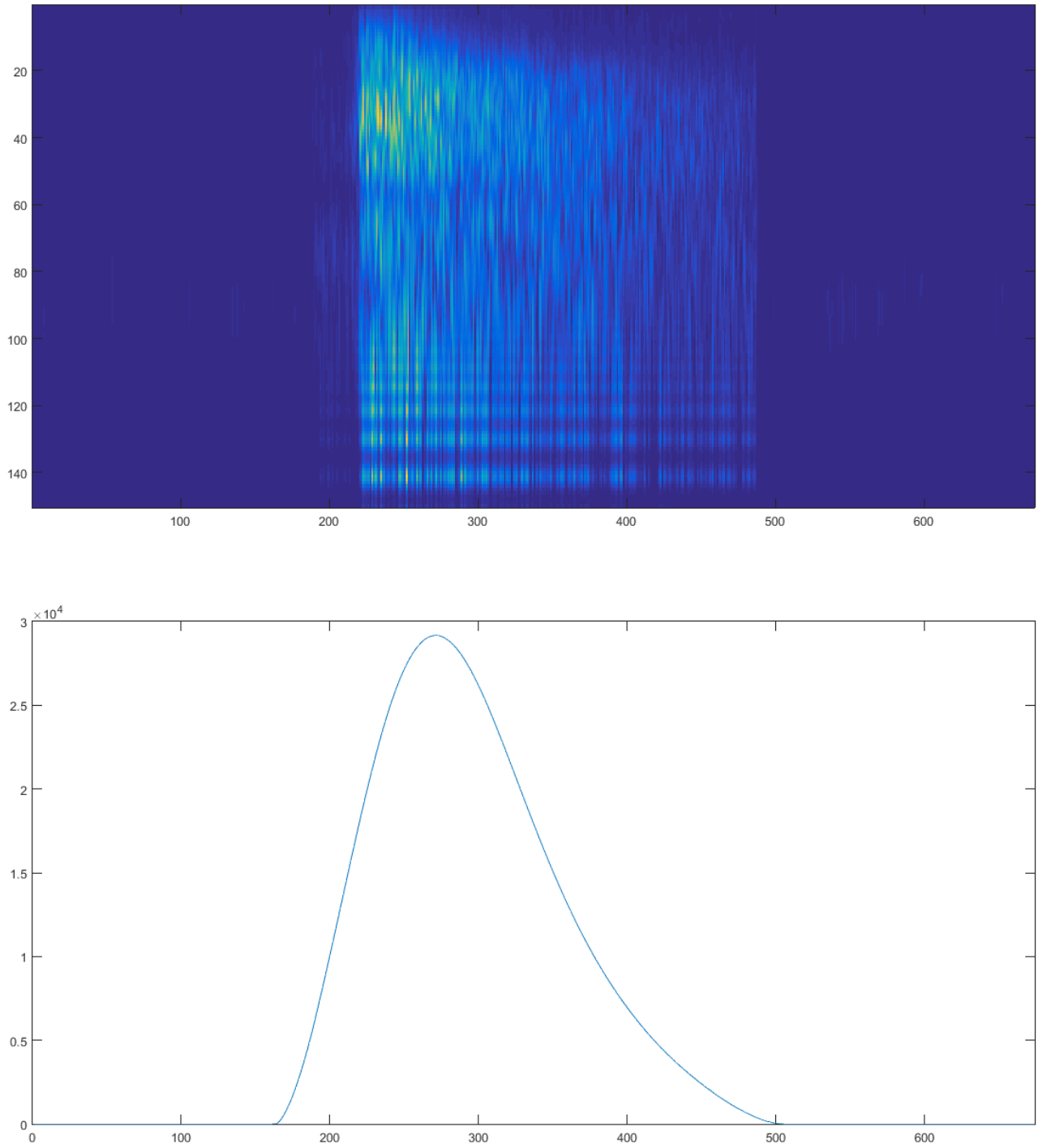


Figure 12: Comparison of gunshot signature with its respective summed response

A suspected event is considered to start at the first non-zero value before a peak, and is considered to end at the first zero value after a peak. A distribution of the summed response between the start and end of each suspected event was taken. This distribution splits the suspected event into temporal regions, with each region holding a localisation score (summed response). This procedure acts as another localisation stage on a finer scale, and is implemented so as to identify the temporal regions within the suspected event which hold the most fruitful features. Considering figure 12 on the previous page, a distribution of localisation score on the illustrated summed response would show that the majority of signature features occur within the temporal region of 200 and 400. Specifically, features existing around the 275 temporal point will be assigned priority as they lie at the center of the event signature. These regional distributions are then normalised to produce weights, which will be assigned to keypoints extracted from the suspected events. Therefore, keypoints at the center of an event signature are given a greater weight than keypoints towards the edges of the signature.

4.6 Classification

Multiple statistical and mathematical approaches to classification exist, each with a large set of parameters to be tailored to the application in question. Support Vector Machines are amongst the most popular classifiers within computer vision, due to their compatibility with high-level image descriptors, such as histogram-of-features, and their flexibility in handling both linear and non-linear problems through the employment of kernels. For this reasons, SVMs[5] were the preferred classification approach from the outset.

One characteristic of traditional SVM implementations is their inability to deal with multi-class problems. For this reason, the proposed design involved the training of a set of SVMs; one for each event class. The preliminary tests of the classification module, however, achieved poor results, with a high false rejection rate as well as a high false alarm rate; highly problematic for a system proposed to run within a real-world context. A solution to this problem was sought through the extensive research of classification approaches within the context of event detection. In [2] and [4], Foggia et al. suggests the training of a dedicated SVM classifier for the classification of background noise as a remedy to these low ratings. An extra SVM class was thus trained for the detection of noise, with each suspected event being assigned the label of the SVM class which achieved the greatest score. This alteration reduced the false alarm rate noticeably, however a lack of progress

[5]SVM scripts imported from the LibSVM library [35]

with respect to the false rejection rate was observed.

In [27], Clavel et al. highlight the trouble of classifying events of varying noise intensity. The training of a classifier on clean data with a high SNR value achieves fair results on test data of equal noise quality, but is inadequate for the classification of noisier events. The superimposition of substantial noise with events, coupled with the training of a classifier on clean data, leads the classifier to falsely reject the large majority of test events as noise. Equally, the training of classifiers upon noisy data triggers a sizeable increase in false detection rate when classifying cleaner data, with background noise being considered to constitute an event. Valenzise et al. reiterate this predicament in [28], mentioning the need for a trade-off between false rejection and false alarm rates, by tuning the noise level of the training data according to the expected test data conditions.

The proposed solution for this problem is similar to that implemented by Dufaux et al. in [36]. The authors recommend training a set of SVM classifiers for different noise levels, so as to handle both high and low SNR conditions. In the case of the proposed system, an SVM classifier was trained for each of the SNR levels, ranging from 5dB to 30 dB in 5 dB increments, for each of the event classes. This resulted in 4 classes (3 events + background noise) holding 7 SVMs each; 6 trained upon data of a specific SNR level, with the final SVM trained upon data across all SNR levels. Each suspected event is assigned a classification score by each of the 28 SVMs, with the final classification relating to the class of the SVM producing the greatest score. This proposal was found to increase execution time, but resulted in a significant improvement in both the false rejection and false alarm rates.

A prominent feature within SVMs which provides room for optimisation is the implementation of a kernel function. A set of tests were initiated to search for the optimal SVM kernel and regularisation parameter λ for the problem in hand. Initial tests focused on the standard kernels packaged with the inbuilt SVM functions, namely: linear, quadratic, polynomial, Gaussian radial basis function and multilayer perceptron kernels. The optimal parameter pairing was found to be a polynomial kernel with a λ value of 0.001, however improvements in accuracy were minuscule and inconsistent. In [37], the suitability of the histogram intersection kernel for the classification of colour-based images is recorded. Referring to equation 3 on the following page, the histogram intersection kernel considers two histograms (a and b) relating to training and testing data. A summation over the set of bins (n) within the histograms is made, with the minimum bin value of either of the two histograms taken. Classification is then performed upon the newly constructed kernel,

yielding the classification score and final label. This kernel, requiring no regularisation parameter, was found to achieve the greatest validation scores, and was thus the kernel of choice early into the development of the system.

$$K(a,b) = \sum_{i=1}^{n} min(a_i, b_i) \qquad a_i \geq 0 \qquad b_i \geq 0 \tag{3}$$

At this stage of the implementation, a modest evaluation score was being achieved, with room for improvement with respect to the false alarm (true negative) rate. A number of suspected events, with a noise ground truth label were erroneously being classified as events. A neighbourhood reduction process, whereby suspected events within a neighbourhood would be re-evaluated, was proposed as the final stage of the pipeline. A neighbourhood size was defined so as to perform a sequential search for suspected events across the audio file. If two or more suspected events were found to exist within the same neighbourhood, a comparison between the events is made so as to eliminate all but one as noise. Two measures of comparison were considered: localisation score and classification score. Validation testing ruled out classification score as inconsistent, with localisation scores acting as a far more credible measure for this decision problem. A range of neighbourhood sizes were also considered, with the optimal being found to be a 3 second diameter. This procedure was based upon the assumption that no two events within the dataset could occur within a temporal range of less than 3 seconds.

The final outcome was deemed to be highly satisfactory, concluding the implementation of the system. The predicted events were then compared to the ground truth during an evaluation procedure, producing salient data relating to the functionality and performance of the system. A description of the final evaluation will be provided in the following section.

5 Evaluation

We shall proceed by highlighting the proposed method of evaluation for the system, considering approaches commonly adopted in literature. A set of experiments designed to test and compare different approaches and system parameters shall then be conducted, before producing the final system results. The section will then be closed off with a discussion of the results obtained.

5.1 Method of Evaluation

For the purpose of system testing and parameter tuning, datasets are typically segmented into three parts: training data, validation data and testing data. This practice ensures the generalisation of the system and prevents bias, by considering the test set to be real-world data, which cannot be accessed prior to the final release of the system. The MIVIA Audio Events Dataset on which the proposed system was developed has been readily split into a training and testing set. The set of audio files containing events superimposed with background noise were segmented at a ratio of 70% to 30% respectively by Foggia et al. in [4]. For the purpose of cross-validation, the training set was further split during the first stages of the pipeline, into a training and validation set. A training percentage parameter β, dictating the ratio of training data to be assigned to the validation or training set, was defined. A randomiser was then implemented to randomly, with a probability of 1 - β, assign a training event to the validation set. This cross-validation procedure was found to suffice, with a β value of 80% considered ideal.

In the context of classifier evaluation, True Positives (TP) and False Positives (FP) are typically adopted to assess the accuracy of assigned labels. For purposes of standardisation and comparison with similar systems, we shall elaborate on the above classification definitions. Typically, a true positive sample is a sample which has been correctly classified with its appropriate label. Within the setting of the defined system, and as perceived in [2, 4], a true positive shall be considered to exist whenever any overlap occurs between a suspected event and a ground truth event, with both elements agreeing on an event label. Alternatively, a false positive shall be considered to occur when a suspected event is erroneously assigned an event label, during an event-free interval.

$$True\,Positive\,Rate = \frac{TP}{Total\,number\,of\,events\,to\,detect} \tag{4}$$

$$False\,Positive\,Rate = \frac{FP}{Total\,number\,of\,noise\,intervals} \tag{5}$$

$$Precision = \frac{TP}{Total\,number\,of\,events\,detected} \tag{6}$$

$$False\,Rejection\,Rate = \frac{Number\,of\,events\,missed}{Total\,number\,of\,events\,to\,detect} \tag{7}$$

The accuracy of the proposed system shall be measured by considering the true positive rate (TPR); the average correct classification rate of each of the event classes, as dictated in equation 4 above. This will allow us to gauge the effectiveness of this system as a whole. The detection rate is separately computed through the false positive rate (FPR), as the ratio of detected false positives to the number of intervals between two events, as illustrated in equation 5 above. This measure will allow us to assess the effectiveness of the localisation stage, determining whether the adopted novel approach holds water. System precision (P), in equation 6 above, is a measure of the classification accuracy of the system, disregarding the detection accuracy and allowing for an evaluation of the systems classification abilities. Finally, the false rejection rate (FRR), also known as the miss rate, is depicted in equation 7. It is the ratio measure of events of interest wrongly classified as noise and thus, falsely rejected.

A confusion matrix, comparing ground truth labels with predicted categories, shall be constructed so as to display the proportion of true positives and false positives across the different event classes. The system results shall then be further broken down with respect to SNR levels, so as to analyse the system's effectiveness at detecting different events in varying conditions. This evaluation will allow us to better understand the capabilities of the system, while directly comparing the results achieved with those attained by Foggia et al. in [4].

In the following subsection we shall discuss the various experiments undertaken upon the validation set during the construction and tuning of the system. The system shall then be tuned with the most suitable parameters, before evaluating the final performance upon the test set.

5.2 Experiments

5.2.1 Image Representation

Motivation for experimentation upon the different image representations stems from the thorough discussion, in the previous sections, of their characteristic differences and the possible effects they may have on the system. So as to solely focus on the effects of the different image representations, all other system parameters were set to standard rational values, typically adopted within literature. Specifically, the Bag-of-Words model with a vocabulary size of 500 and no spatial tiling was adopted. Furthermore, neighbourhood reduction with a neighbourhood size of 3 seconds based upon localisation score was implemented within the pipeline, due to the significant effects on the results observed. The neighbourhood reduction stage will be further evaluated at a later phase, following further interpretation and insight into the system's competence.

	TPR	FPR	P	FRR
Linear-Frequency Spectrogram	84.65%	6.04%	81.47%	2.11%
Logarithmic-Frequency Spectrogram	91.37%	2.01%	91.44%	2.08%
Gammatonegram	86.71%	4.18%	87.03%	3.90%

Table 1: Table of experimental results for different image representations

Referring to table 1, right off the bat, it is clear that all three representations achieve satisfactory results with overall accuracy ratings of over 80%. On closer inspection though, the effects of the different image representations are apparent. Comparing all evaluation criteria save the false rejection rate, linear-frequency spectrograms achieve the worst results across the board, with the lowest true positive rate of 84.65%. In relation to the other representations, it achieves a poor precision in the lower 80 percentile range, illustrating the struggle of classifying such representations. This outcome was anticipated in section 2, whereby the linear-scale representation was expected to inadequately express lower frequency bands with respect to their high-pitched counterparts.

Although results improved with the employment of the gammatonegram, the best outcome was attained by the logarithmic-scale spectrogram. These results confirm the importance of incorporating logarithmic scales for the classification of events mapped onto the frequency domain. The additional application of the gammatone filter bank to the logarithmic representation was found to hinder results, decreasing classification precision and thus overall accuracy. An increase in the false positive rate and false rejection rate

followed, due to the system's inability to consistently distinguish between events of interest and noise.

This experiment thus concludes the suitability of the logarithmic-scale spectrogram representation for audio event segments. All further experiments shall therefore be conducted utilising this representation with the goal of fine tuning all procedures within the pipeline.

5.2.2 Vocabulary Size and Spatial Information

As previously explained, the spatial encoding employed correlates directly to the feature descriptor size; as does the vocabulary size. Given the relevance of these two system parameters, it was considered suitable to perform a joint evaluation through one experiment. Based off the results of the previous experiment, logarithmic-frequency spectrogram representation was applied in all cases, with the only variables changed being the vocabulary size and spatial pyramid levels. The outcome of this experiment shall dictate the most suitable combination of these parameters to be tuned for the final evaluation.

Vocabulary Size K	Spatial Levels	TPR	FPR	P	FRR
1000	1	91.71%	2.01%	91.78%	2.08%
2000	1	91.59%	4.97%	88.69%	1.70%
200	2	90.76%	4.97%	87.89%	1.70%
400	2	91.97%	2.01%	92.01%	2.05%
50	3	93.03%	2.02%	93.07%	2.05%
100	3	93.26%	2.01%	93.30%	2.05%

Table 2: Table of experimental results for different vocabulary sizes and spatial tiling

Table 2 highlights the results achieved across the various vocabulary size and spatial level values. The vocabulary sizes set for evaluation were respective of the levels incorporated within the spatial pyramid, so as to produce event descriptors of similar length. Observing the first two rows, results similar to the previous experiment with the logarithmic-frequency representation were achieved. The increase in the vocabulary size K yielded no noticeable improvements, with a value of 2000 contrarily reducing the true positive rate, most likely due to over-fitting.

The central two rows portray results of the system with a 2 level spatial pyramid encoding. This results in a final image descriptor of size 5K, supporting the reduction in

vocabulary size. The produced results unexpectedly show a slight decrease in accuracy with K set to 200, and a slight improvement upon the current base-line with a larger vocabulary size. Considering the rich visual signature produced by each of the events, it is apparent that a 2 level pyramid does not encode spatial information at a sufficient level of granularity.

Finally, results for the 3 level spatial pyramid implementation exhibit marginal improvements in both precision and general accuracy. The greatest accuracy score achieved was that of 93.26%, considering a vocabulary size of 100. This outcome follows previous expectations, as the system is better able to distinguish between different events through the comparison of the spatial location of each of the event keypoints. These parameters, however, result in each event being described by a vector of 21K, lengthening execution time noticeably. Nonetheless, the results were considered satisfactory and were deemed to be computed within a reasonable time-frame. Due to the computational burden brought about by increased spatial pyramid levels, together with the marginal improvements in accuracy, further experimentation with additional pyramid levels was considered unnecessary.

5.2.3 Neighbourhood Reduction

Neighbourhood reduction is a vital stage of the system pipeline, acting as a filter for noise segments which, up to this point, were suspected of being events. This procedure is the final stage of the pipeline, and may be employed upon two reduction criteria: localisation score or classification score. We shall conduct an accuracy comparison between these different scoring criteria with a neighbourhood size of 3 seconds, as well as a base case whereby no neighbourhood reduction is applied, so as to verify the relevance of this procedure.

Neighbourhood Reduction	TPR	FPR	P	FRR
Localisation Score	93.26%	2.01%	93.30%	2.05%
Classification Score	90.91%	4.85%	91.02%	4.96%
None	93.53%	12.99%	83.96%	1.78%

Table 3: Table of experimental results for neighbourhood reduction

Observing table 3 above, the first row depicts the results already illustrated in the previous experiments, whereby neighbourhood reduction based upon localisation score was conducted. When classification score is set to decipher between events and noise,

a marginal reduction in accuracy is observed due to the decreased localisation and classification accuracy. This outcome is observed due to the similarity between events and noise in suboptimal, noisy conditions, whereby the classifier struggles to identify the most distinctive event within the neighbourhood resulting in the erroneous labelling of events as noise. Meanwhile, the comparison of suspected events by means of localisation score is found to be more suitable, as, although noisy events may be challenging to distinguish from noise solely through their visual signature, they leave a more prominent response across the spectrum, achieving a distinguishable localisation score.

The most interesting results were achieved in the final test, where no neighbourhood reduction stage was applied. The true positive rate illustrated is the highest accuracy rate achieved during system testing, with the false rejection rate dropping to 1.76%, due to the lack of event filtering. However, one must consider the great reduction in both the false positive rate and the precision rate. These reductions are brought about by a significant increase in false positives, whereby various background noises were not filtered out through neighbourhood reduction, but instead passed through as events of interest, raising false alarms. Considering the proposed application of the implemented system within a real-world setting, false alarms are not catastrophic, but must be kept to the utmost minimal. Therefore, within the context of the system, a minimal increase ($\sim 0.25\%$) in false rejection rate, and thus decrease in accuracy, in favour of a reduction in the false positive rate, was considered beneficial.

5.3 Discussion

Following the conduct of the above mentioned experiments, the system was tuned to execute upon the test set employing logarithmic spectrogram image representation, with the Bag-of-Words model based upon a vocabulary of size 100. A 3-level spatial pyramid encoding was adopted for the description of events, with neighbourhood reduction applied considering a 3 second neighbourhood window, and localisation score acting as the comparator. The complete training set, consisting of 16,726 events, was used to train the system prior to the detection and classification of the test audio files holding 5,376 events of interest.

	TPR	FPR	P	FRR
Proposed System	91.13%	4.35%	89.59%	2.64%
Foggia et al.	86.7%	2.6%	97.7%	10.7%

Table 4: Comparison of system results with those achieved by Foggia et al. in [4]

The final results achieved by the proposed system are outlined in table 4 above, together with those obtained by Foggia et al. in [4] upon the same dataset. As one may observe, the final results are slightly lower than those attained during validation. However, this is expected with all training models, given the biased tuning towards validation sets. The proposed pipeline has surpassed Foggia et al.'s system by more than 4% in terms of true positive rate. This is especially understandable when comparing the false rejection rates of the two systems. While the latter dismisses one out of every ten events of importance as noise, a minute score of 2.64% is achieved by the former, explaining the improved detection rate and overall score. This observation in particular confirms the suitability of the proposed localisation stage for such a system. However, while digesting these results, it is vital to recall the differences between the two systems; while Foggia et al.'s approach provides a solution for real-time audio surveillance, the proposed approach acts in an off-line manner, and therefore follows a far more lenient boundary with respect to computational efficiency.

		Predicted Events			
		Background Noise	Breaking Glass	Gunshot	Scream
Actual events	Background Noise	1814	34	154	46
	Breaking Glass	36	1717	45	2
	Gunshot	55	149	1587	9
	Scream	51	27	103	1595

Table 5: Confusion matrix comparing actual test events with their predicted classification

Table 5 above illustrates the confusion matrix produced by the proposed system, allowing us to better understand the effectiveness of the system at detecting each of the event classes. Breaking glass was by far the easiest event class to detect and classify, achieving the highest true positive rate with a minimal number of misclassifications and false rejections. This is most certainly due to the event's distinct signature, which holds features comparable to those of the other event classes, but within dissimilar spatial regions. Meanwhile, the true positive rates for gunshots and screams are slightly inferior, but nevertheless

satisfactory. The importance of training a noise classifier as well as the implementation of the neighbourhood reduction stage is reiterated in the above table, whereby 1,814 audio snippets were suspected of holding an event of interest during the localisation stage, but were later correctly rejected as noise by the classifier or neighbourhood reduction stage.

SNR Level	Breaking Glass	Gunshot	Scream	Total	Foggia et al. Total
5dB	93.00%	83.00%	73.65%	83.26%	81.1%
10dB	96.67%	87.33%	89.19%	91.07%	85%
15dB	95.67%	89.33%	94.26%	93.08%	87%
20dB	96.00%	89.00%	94.26%	93.08%	88.4%
25dB	95.33%	90.33%	93.58%	93.08%	88.7%
30dB	95.67%	90.00%	93.92%	93.19%	90%
Total	95.39%	88.17%	89.81%	91.13%	86.7%

Table 6: Breakdown of True Positive Rate achieved for each event class at each SNR level. A comparison is made between the final results of the proposed system and those achieved by Foggia et al. in [4]

A detailed evaluation is produced in table 6 above, providing insight into the system's capabilities at detecting and classifying different events within different conditions. The above data re-confirms the system's ability to better detect and classify breaking glass when compared to the other events, with a gratifying score of 93% achieved within highly unsuitable conditions of 5dB SNR. Notwithstanding this, the system has proven highly capable of dealing with both impulsive and sustained sounds, achieving gratifying results within all three classes.

It is interesting to note the slight increase in accuracy of both breaking glass and scream detection within marginally suboptimal conditions, when compared to their improved SNR counterparts. This response could be due to the additive effect of noise upon the event signature, coupled with the training of multiple classifiers. Within certain thresholds of noise intensity, these events may be easier to distinguish from one another, resulting in a decrease in misclassifications and thus, an increase in accuracy.

As was expected, greater accuracy scores were achieved within optimal conditions, with a maximum SNR total of 93.19% in exemplary conditions of 30dB SNR. Results degrade minimally as the SNR decibel level falls, until a marginal reduction is observed at the 5dB level. Similar trends are observed within the results achieved by Foggia et al., reiterating the challenge of detection and classification posed by highly noisy environments. Observing

the final two columns in the table, it is clear that the proposed system out-performs Foggia et al.'s real-time system in all audio conditions, cementing the success of the proposed pipeline.

6 Future Work

The system evaluation phase has confirmed the effectiveness of the proposed pipeline, achieving satisfactory results, even in challenging noisy environments. Although all the aims and objectives set out at the start of this project have been met, further improvements and alterations to the system could bring about noticeable enhancements in both functionality and efficiency. One pitfall of the current implementation is the ineptitude to detect adjoining or overlapping events. Inspiration may be taken from similar applications within literature, such as that of Dennis et al. [15], whereby a Generalised Hough Transform voting system is employed to identify overlapping sound events.

Considering that the system scope has been limited to an off-line, interpretive manner, system efficiency is not of critical importance when compared to real-time systems. However, a system efficient in the management and consumption of resources, and which executes within a respectable time-frame is always sought after. The implementation of a separate localisation stage, as opposed to a windowing method, has brought about improvements in efficiency. However, future efforts could be focused on the redesign of a more efficient, but equally effective, classification stage, offering an alternative to the current employment of 28 SVM classifiers.

The procedural steps involved in this pipeline are not domain specific, thus allowing this system to be trained and tweaked for the detection of any set of events. Future work could see the generalisation abilities of the system tested, and the scope expanded to a general event detector, having the system trained to detect dozens of event classes and subclasses, each with their own characteristics and hurdles.

7 Conclusion

Following the design, implementation and evaluation of the initial proposal, a number of inferences may be made with respect to the selected approach and intermediary stages. The ramifications of the selection of image representation schemes has been confirmed, with scaling of the frequency spectrum found to have a direct effect on the results produced. The proposed separation of the localisation and classification stages has proved successful at handling both impulsive and sustained events, which conventional windowing approaches, common within literature, struggle to deal with.

The system was able to obtain satisfactory results across all SNR levels and event classes, meeting all the aims and objectives laid out from the onset. Through the transformation of audio data onto the visual domain, and the employment of computer vision techniques for the analysis and description of events, a greater robusticity to noise was achieved when compared to the direct analysis of low-level audio features. We may therefore confirm the functionality and applicability of the employment of computer vision techniques to overcome extensive noise within an automated audio aggression detection system.

References

[1] Jerry H Ratcliffe, Travis Taniguchi, and Ralph B Taylor. The crime reduction effects of public cctv cameras: a multi-method spatial approach. *Justice Quarterly*, 26(4):746–770, 2009.

[2] Pasquale Foggia, Nicolai Petkov, Alessia Saggese, Nicola Strisciuglio, and Mario Vento. Audio surveillance of roads: a system for detecting anomalous sounds. *IEEE Transactions on Intelligent Transportation Systems*, 17(1):279–288, 2016.

[3] Marco Crocco, Marco Cristani, Andrea Trucco, and Vittorio Murino. Audio surveillance: a systematic review. *ACM Computing Surveys (CSUR)*, 48(4):52, 2016.

[4] Pasquale Foggia, Nicolai Petkov, Alessia Saggese, Nicola Strisciuglio, and Mario Vento. Reliable detection of audio events in highly noisy environments. *Pattern Recognition Letters*, 65:22–28, 2015.

[5] Marco Cristani, Manuele Bicego, and Vittorio Murino. Audio-visual event recognition in surveillance video sequences. *IEEE Transactions on Multimedia*, 9(2):257–267, 2007.

[6] Robert C Maher. Audio forensic examination. *IEEE Signal Processing Magazine*, 26(2):84–94, 2009.

[7] Bruce Masterton, Henry Heffner, and Richard Ravizza. The evolution of human hearing. *The Journal of the Acoustical Society of America*, 45(4):966–985, 1969.

[8] RD Patterson, Ian Nimmo-Smith, John Holdsworth, and Peter Rice. An efficient auditory filterbank based on the gammatone function. In *a meeting of the IOC Speech Group on Auditory Modelling at RSRE*, volume 2, 1987.

[9] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[10] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[11] Jun Yang, Yu-Gang Jiang, Alexander G Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of*

the international workshop on Workshop on multimedia information retrieval, pages 197–206. ACM, 2007.

[12] Stephanie Pancoast and Murat Akbacak. Bag-of-audio-words approach for multimedia event classification. In *Interspeech*, pages 2105–2108, 2012.

[13] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE, 2010.

[14] Jonathan Dennis, Huy Dat Tran, and Haizhou Li. Spectrogram image feature for sound event classification in mismatched conditions. *IEEE Signal Processing Letters*, 18(2):130–133, 2011.

[15] Jonathan Dennis, Huy Dat Tran, and Eng Siong Chng. Overlapping sound event recognition using local spectrogram features and the generalised hough transform. *Pattern Recognition Letters*, 34(9):1085–1093, 2013.

[16] J-L Rouas, Jérôme Louradour, and Sébastien Ambellouis. Audio events detection in public transport vehicle. In *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, pages 733–738. IEEE, 2006.

[17] Pasquale Foggia, Alessia Saggese, Nicola Strisciuglio, and Mario Vento. Cascade classifiers trained on gammatonegrams for reliably detecting audio events. In *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on*, pages 50–55. IEEE, 2014.

[18] Rui Cai, Lie Lu, Hong-Jiang Zhang, and Lian-Hong Cai. Highlight sound effects detection in audio stream. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 3, pages III–37. IEEE, 2003.

[19] Silvia Pfeiffer, Stephan Fischer, and Wolfgang Effelsberg. Automatic audio content analysis. In *Proceedings of the fourth ACM international conference on Multimedia*, pages 21–30. ACM, 1997.

[20] Pedro Cano, Eloi Batlle, Ton Kalker, and Jaap Haitsma. A review of audio fingerprinting. *Journal of VLSI signal processing systems for signal, image and video technology*, 41(3):271–284, 2005.

[21] Jonathan T Foote. Content-based retrieval of music and audio. In *Voice, Video, and Data Communications*, pages 138–147. International Society for Optics and Photonics, 1997.

[22] Nicolas Scaringella, Giorgio Zoia, and Daniel Mlynek. Automatic genre classification of music content: a survey. *IEEE Signal Processing Magazine*, 23(2):133–141, 2006.

[23] Dong Yu and Li Deng. *Automatic speech recognition: A deep learning approach*. Springer, 2014.

[24] Robert C Maher and Steven R Shaw. Deciphering gunshot recordings. In *Audio Engineering Society Conference: 33rd International Conference: Audio Forensics-Theory and Practice*. Audio Engineering Society, 2008.

[25] Brian M Brustad and John C Freytag. A survey of audio forensic gunshot investigations. In *Audio Engineering Society Conference: 26th International Conference: Audio Forensics in the Digital Age*. Audio Engineering Society, 2005.

[26] Hafiz Malik. Acoustic environment identification and its applications to audio forensics. *IEEE Transactions on Information Forensics and Security*, 8(11):1827–1837, 2013.

[27] Chloé Clavel, Thibaut Ehrette, and Gaël Richard. Events detection for an audio-based surveillance system. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 1306–1309. IEEE, 2005.

[28] Giuseppe Valenzise, Luigi Gerosa, Marco Tagliasacchi, Fabio Antonacci, and Augusto Sarti. Scream and gunshot detection and localization for audio-surveillance systems. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pages 21–26. IEEE, 2007.

[29] Vincenzo Carletti, Pasquale Foggia, Gennaro Percannella, Alessia Saggese, Nicola Strisciuglio, and Mario Vento. Audio surveillance using a bag of aural words classifier. In *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, pages 81–86. IEEE, 2013.

[30] M. Molinari. Xml toolbox. `https://www.mathworks.com/matlabcentral/fileexchange/4278-xml-toolbox`, 2005.

[31] D. P. W. Ellis. logsfgram.m. https://github.com/dpwe/alignmidi/blob/master/logfsgram.m, 2013.

[32] D. P. W. Ellis. Gammatone-like spectrograms. http://labrosa.ee.columbia.edu/matlab/gammatonegram/, 2009.

[33] Andrea Vedaldi and Brian Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 1469–1472, New York, NY, USA, 2010. ACM.

[34] George Azzopardi, Nicola Strisciuglio, Mario Vento, and Nicolai Petkov. Trainable cosfire filters for vessel delineation with application to retinal images. *Medical image analysis*, 19(1):46–57, 2015.

[35] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[36] Alain Dufaux, Laurent Besacier, Michael Ansorge, and Fausto Pellandini. Automatic sound detection and recognition for noisy environment. In *Signal Processing Conference, 2000 10th European*, pages 1–4. IEEE, 2000.

[37] Annalisa Barla, Francesca Odone, and Alessandro Verri. Histogram intersection kernel for image classification. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 3, pages III–513. IEEE, 2003.