Investigating audio classification to automate the trimming of recorded lectures

By

Devandran Govender

Supervised by

Professor Hussein Suleman

Minor dissertation submitted in partial fulfilment of the requirements for the degree of Master of Science in Information Technology

Department of Computer Science

University of Cape Town

February 2018

DECLARATION

I know the meaning of plagiarism and declare that all the work in this dissertation, save for that is properly acknowledged, is my own.

Signature: Signed by candidate

ACKNOWLEDGEMENTS

I extend my thanks to:

- My supervisor, Prof. Hussein Suleman, for his guidance and patience.
- Catherine Fortune for her continued support and motivation.
- Stephen Marquard, for his advice and help.
- Theodoros Giannakopoulos, the developer of the *pyAudioAnalysis* library.

ABSTRACT

With the demand for recorded lectures to be made available as soon as possible, the University of Cape Town (UCT) needs to find innovative ways of removing bottlenecks in lecture capture workflow and thereby improving turn-around times from capture to publication. UCT utilises Opencast, which is an open source system to manage all the steps in the lecture-capture process. One of the steps involves manual trimming of unwanted segments from the beginning and end of video before it is published. These segments generally contain student chatter. The trimming step of the lecture-capture process has been identified as a bottleneck due to its dependence on staff availability.

In this study, we investigate the potential of audio classification to automate this step. A classification model was trained to detect two classes: speech and non-speech. Speech represents a single dominant voice, for example, the lecturer, and non-speech represents student chatter, silence and other environmental sounds. In conjunction with the classification model, the first and last instances of the speech class together with their timestamps are detected. These timestamps are used to predict the start and end trim points for the recorded lecture.

The classification model achieved a 97.8% accuracy rate at detecting speech from non-speech. The start trim point predictions were very positive, with an average difference of -11.22s from gold standard data. End trim point predictions showed a much greater deviation, with an average difference of 145.16s from gold standard data. Discussions between the lecturer and students, after the lecture, was predominantly the reason for this discrepancy.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

Abbreviation	Term
ASC	Audio Signal Classification
CILT	Centre for Innovation in Learning and Teaching
CRRM	Cepstrum Resynthesis Residual Magnitude
DCT	Discrete Cosine Transform
DFB	Distance-From-Boundary
DLSF	Differential Line Spectral Frequencies
FFMPEG	Fast Forward MPEG
FFT	Full Fourier Transform
FLAC	Free Lossless Audio Codec
FLV	Flash Video
FM	Frequency Modulation
FN	False Negative
FP	False Positive
f_o	Fundamental Frequency
GS	Gaussian Classifier
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HOC	Higher Order Crossing
IP	Internet Protocol
k-NN	K-Nearest Neighbour
JSON	JavaScript Object Notation
LPC	Linear Predictive Coding
LP-ZCR	Linear Prediction Zero Crossing Ratio
LSF	Line Spectral Frequencies
LSF-HOC	Line Spectral Frequencies with Higher Order Crossings
LSF-ZCR	Line Spectral Frequencies with LP-ZCR
MFCC	Mel Frequency Cepstrum Coefficients
MM	Markov Model
MP4	MPEG-4
ms	Milliseconds
MPEG	Moving Picture Experts Group
NDSF	Normalized Dynamic Spectral Features
NFL	New Feature Line
NN	Nearest Neighbour
PPV	Positive Predictive Value
RASTA	“RelAtive SpecTrA”
RMS	Root Mean Square
s	Seconds
SMIL	Synchronised Multimedia Integration Language

STFT	Short-Term Fourier Transform
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
UCT	University of Cape Town
VAD	Voice Activity Detection
WAV	Waveform Audio File Format
WBLT	Web-based Lecture Technologies
ZCC	Zero Crossing Count
ZCR	Zero Crossing Rate

1. INTRODUCTION

1.1 Lecture recording in higher education

Lecture recording systems capture audio, video and presentation slides during a lecture, which are thereafter combined and published as a single video, so that students can playback the lecture at their convenience for studying and revision purposes. Lecture recording at institutes of higher learning is now fairly common practice and has been proven to be an important resource to students [1, 2]. Studies have also shown that lecture recording and other Web-based lecture technologies (WBLT) have been well received by students [3, 4].

Today's students face increased challenges of balancing their studies, work and family commitments [5, 6]. Therefore, students have shown increased appreciation for the flexibility that online resources, such as lecture recordings, provide [7, 8]. With the increased interest by students for lectures to be recorded, it follows that there is an increase in demand for the published recordings to be made available as soon as possible. Institutions will therefore need to be innovative in addressing workflow bottlenecks and finding ways of improving the turn-around time from capture to publication.

1.2 Lecture recording at the University of Cape Town

Lecture recording has been deemed a core business service at the University of Cape Town (UCT) and the Centre for Innovation in Learning and Teaching (CILT) are the custodians of this service. To manage and administer the lecture capture process, CILT has opted to utilise Opencast [9], which is an open source Java based framework. Opencast manages the various stages involved in the lecture capture process, which include:

Scheduling: Before the academic year commences, course convenors schedule all courses that they wish to be recorded. The course codes, dates, times and venues

are all retrieved automatically from UCT's timetabling software. Once the recordings have been scheduled, the lectures are automatically recorded accordingly.

Encoding and processing: After a lecture has been recorded, the raw media files are ingested by Opencast and enriched with metadata, preview images, captioning and text analysis to improve discoverability and accessibility.

Editing and trimming: Any irrelevant content that exists in the raw media is marked for removal during this stage. Opencast thereafter excludes these segments from the final published video. Metadata can also be updated or corrected during this stage.

Distribution: The finalised recording is published for on-demand viewing or download via Sakai.

The Opencast editor, which is a web-based video editor, is used to trim and edit recordings. Figure 1.1 below shows an unsegmented video loaded within the Opencast editor, with the three main regions of the editor labelled as *A*, *B* and *C*. Region A contains the video streams (IP camera and projector feed) and the playback controls. Region B contains the timeline and the composite toolbar which are used to select segments that are to be trimmed. Region C contains clickable tabs that display information related to the selected segments, metadata and any editor comments.

During the trimming process, staff will use the video controls and skim through the video in search of unnecessary content, for example student chatter, which predominantly occurs at the beginning of the recording (before the lecture starts) and at the end (after the lecture concludes). The composite toolbar is used to select segments that are to be excluded from the final published video. Once selected, the respective segments in the composite toolbar are highlighted and information about the segments appears in the segments section, as shown in Figure 1.2.

The editing and trimming stage is a manual and subjective process, that is highly dependent on the availability of staff. This dependence on human intervention does impact the publishing of recordings negatively, making it a primary bottleneck of the system. This is especially true on Friday evenings, when videos enter the trim queue and can only be attended to on Monday as staff are not available over the weekends.

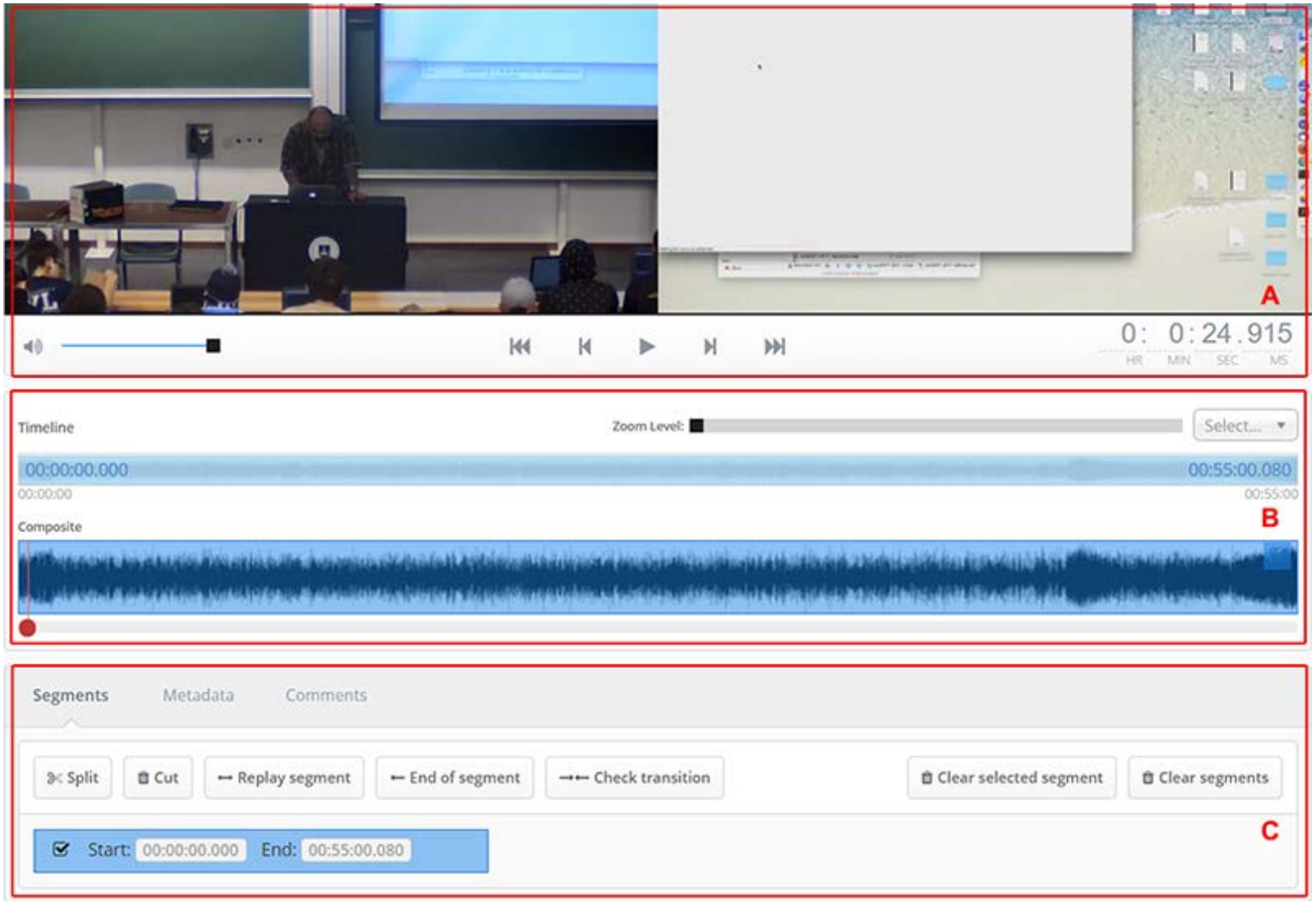

Figure 1.1: An unsegmented video loaded within the Opencast interface.

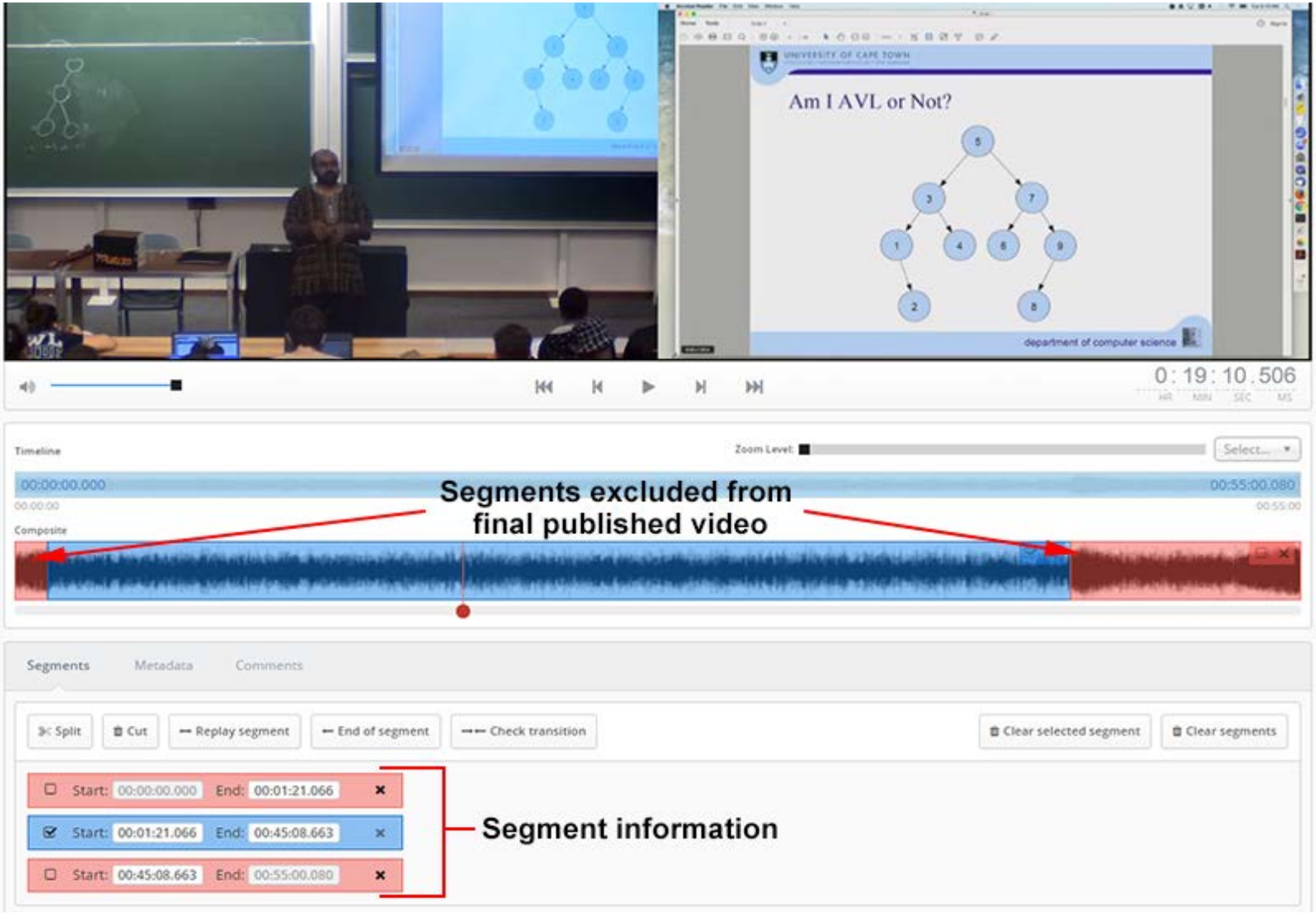

Figure 1.2: The Opencast editor with segments selected for trimming. The areas in red indicate the segments that will excluded in the final published video.

1.3 Motivation

Since the inception of lecture recording at UCT in 2013, there has been a steady increase in the number of recordings published each year, as clearly indicated in Figure 1.3 below.

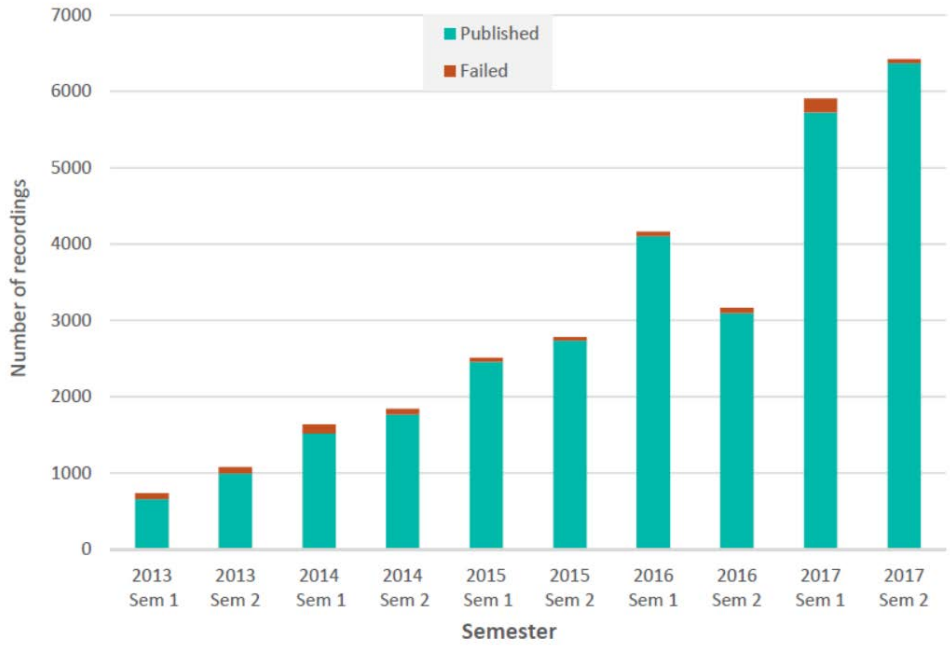

Figure 1.3: The increase in published lecture recordings at the University of Cape Town from February 2013 to December 2017. The drop in the second semester of 2016 was due to classes being cancelled due to unrest on campus when students protested for a zero increase in tuition fees. Source: Centre for Innovation in Learning and Teaching, University of Cape Town.

Coupled with the increase in recordings, there has also been a demand for recordings to be made available sooner than the current turn-around time. If we were to extrapolate the pattern in Figure 1.3, a safe deduction would be that the demand is surely to increase in the years to follow. This means that the required dependence on staff availability is not sustainable or practical. Therefore, alternate intuitive methods need to be investigated. Automating the trimming task could potentially alleviate the bottleneck and remove the required dependency on staff, thereby improving the turn-around time for published recordings. However, automation would require a level of

intelligence to distinguish relevant content from irrelevant content and mark these segments accordingly.

We therefore propose utilising audio signal classification (ASC) to analyse the audio stream from a lecture recording and identify the respective segments for trimming. ASC is a machine learning process by which an audio signal is analysed, a set of audio features extracted from it, and then used to identify a group of classes to which the signal most likely belongs. An audio classification system must be able to analyse an audio signal and detect the type of audio [10], for example speech, music, noise and silence. Therefore, the inclusion of such a system in UCT's lecture capture workflow could potentially identify irrelevant content, such as student chatter, from relevant content, such as lecturer speech, and mark these accordingly for trimming.

1.4 Limitations of this study

This study does not make use of a custom classification system but instead utilises an open source library that performs a range of audio-related functionalities, which include feature extraction and classification. Furthermore, this study excludes the actual implementation of the audio classification model within UCT's lecture capture framework (Opencast).

1.5 Research questions

The aim of this study is to evaluate the accuracy and efficacy of audio signal classification in distinguishing speech from non-speech, as a means of automating the trimming of the recorded lectures at UCT.

The main research question is:

> *How accurately can audio signal classification distinguish speech from non-speech?*

The secondary question is:

> *How do the start and end trim points, determined using audio classification, compare to gold standard data?*

1.6 Methodology

Audio files from previous recordings are downloaded and segmented according to speech (single dominant voice) and non-speech (student chatter, silence, environmental noise).

Using the segmented files, we then implement 10-fold cross validation to train and test a Support Vector Machine (SVM) classification model. Four performance metrics; Accuracy, Precision, Recall and F-measure, are used to evaluate the performance of the classification model. We pay particular attention to the accuracy at which the model detects the speech class.

Finally, in conjunction with the classification model, we determine the start and end trim points for a recorded lecture. The performance is evaluated by comparing predicted trim points against trim points determined manually by staff. The manually determined trim points are considered most accurate and reliable, and are therefore used as the gold standard in this study.

1.7 Thesis structure

This study is divided into five chapters. Chapter 2 presents an overview of existing literature related to audio classification. Chapter 3 provides a detailed overview of the data and methodology implemented to train and evaluate the chosen classification, as well as the steps taken to evaluate the trim point predictions. Chapter 4 presents and discusses the results of this study. This is followed by the conclusion in Chapter 5 where we summarise results and findings and identify possible opportunities for future work.

2. LITERATURE REVIEW

2.1 Introduction

This chapter provides a background review of existing literature pertaining to ASC. The essential components of ASC are discussed in Section 2.2. In this section, we first briefly discuss feature extraction and selection. Thereafter, we provide an overview of the common physical and perceptual features of audio signals. Following this, the popular classification models in the realm of ASC are discussed. The chapter is concluded with Section 2.3, where an overview and discussion of the contributions of previous studies in this research field are provided.

2.2 Core concepts of audio signal classification

While research into ASC has provided many different methodologies, these techniques generally involve two stages of processing [11]. Firstly, a variety of discernible and measurable features are extracted from the audio signal. Thereafter, the extracted features are fed into a pattern classification model to categorise the audio into respective classes.

2.2.1 Feature extraction

Before an audio signal can be classified, the features within that signal first need to be extracted and analysed. These features represent the characteristics of the audio signal and will ultimately decide the class of that signal. The techniques employed during feature extraction can either involve the analysis of the actual waveform of the audio signal, or the analysis of the spectral representation of the audio signal. During the feature extraction stage there is reduction of data from the audio signal as sound data contains much redundancy [12]. This is done by breaking down the audio signal into successive short-time or short-term windows or frames, which are generally no larger than 100ms [13]. A set of features are then calculated for each frame, resulting in a feature vector [14].

For better results, the concept of a texture window was introduced by Tzanetakis and Cook [15], which is much longer than a short-time window, generally in seconds (s) and not milliseconds (ms). For each texture window, the short-time processing is carried out and the feature sequence from each texture window is used to determine feature statistics [14, 15]. The feature sequence from each texture window is not directly the values obtained during each short-time window analysis, but are combined statistical values for all short-time analysis windows within the texture window [13]. This provides long-term characteristics of the audio signal, for example the average value for the energy of the audio signal.

2.2.2 Feature selection

There are many features that could be extracted from an audio signal. However, it is important to select a particular set of features, as implied by Burred and Lerch [13], as reducing the number of features selected not only improves computational costs but may also improve accuracy and the level of performance of the classification system.

Therefore, Burred and Lerch [13] stated that selected features should have the following general properties:

- **Invariance to irrelevancies**: Good features should display invariance to irrelevancies of the input signal, such as noise, amplitude scaling and bandwidth.
- **Discriminative power**: The goal of feature selection is to attain discrimination between classes of audio patterns. This means that features should therefore take similar values for the same class but different values across classes.
- **Uncorrelated to other features**: Each feature selected needs to provide as much new information about the input signal as possible. Therefore, preventing redundancies in the feature space is important.

2.2.3 Features of audio signals

Audio features generally fall into two categories: perceptual features and physical features [12, 16, 17].

Perceptual features refer to the properties of audio that correspond to the way humans perceive sound [16]. They are subjective attributes of audio and therefore cannot be measured by direct physical means. Examples of perceptual features include pitch, loudness, timbre and rhythm.

Physical features refer to properties of audio that correspond to actual physical properties of the signal [12]. Physical features are easier to identify and extract as they are directly related to the physical properties of the actual sound signal and can therefore be physically measured. Examples of physical features that have been used in audio analysis include fundamental frequency (f_o), the zero-crossing rate (ZCR), energy, entropy of energy, spectral centroid, spectral spread, spectral flux, spectral rolloff and Mel frequency cepstral coefficients (MFCC) [16, 18-20]. Stevens, et al. [21] define Mel as a unit of pitch. While many other physical features exist, the above mentioned physical features are discussed below as they are utilised in this study.

Understanding the different features of audio is fundamental to any audio classification system.

2.2.3.1 Perceptual features

2.2.3.1.1 Pitch

Pitch is the quality of a sound signal that is governed by the rate of vibrations producing it, or the degree of highness and lowness in a musical or vocal signal. It is therefore directly proportional to frequency and related to the log of fundamental frequency [12]. According to Guojun and Hankinson [22], only periodic sounds, such as those generated by voiced signals and musical instruments produce pitch. Pitch estimation is an important feature in voiced/unvoiced classification systems [23]. Figure 2.1 illustrates the concept of high and low pitch for an audio signal.

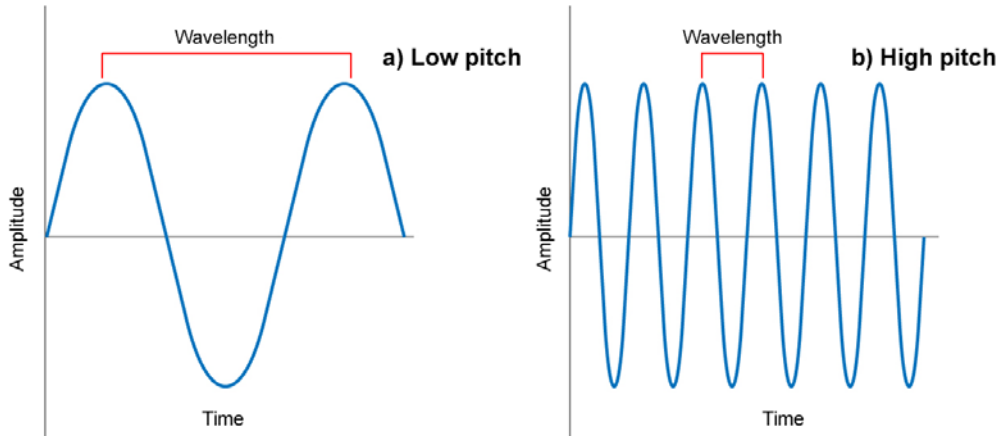

Figure 2.1: Illustration of pitch. a) Low pitch with low frequency. b) High pitch with high frequency.

2.2.3.1.2 Loudness

Loudness refers to the perception of signal strength or intensity [24]. It is therefore a subjective measure of how soft or loud a signal is. It is approximated by the level of the audio signal's root-mean square (RMS), measured in decibels [25]. A signal with a high amplitude is therefore perceived as louder than a signal with a low amplitude. Figure 2.2 illustrates the concept of loudness for an audio signal.

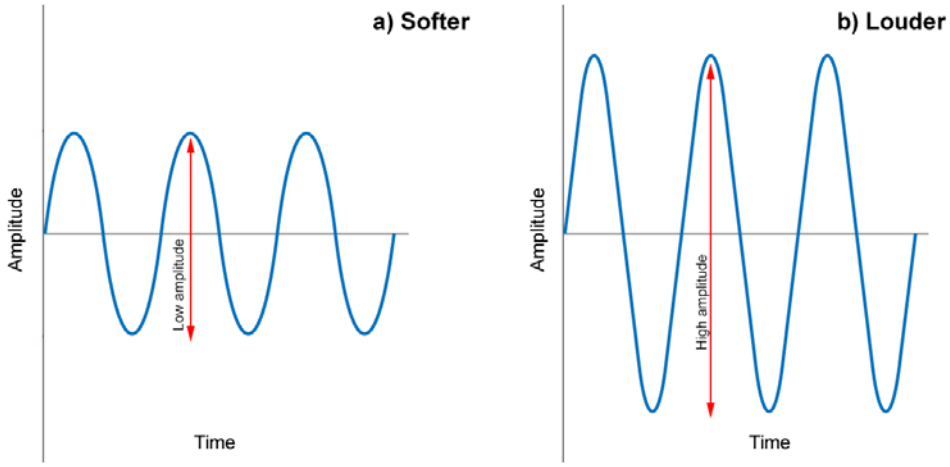

Figure 2.2: Illustration of loudness. a) A soft signal, b) A loud signal.

2.2.3.1.3 Timbre

Timbre refers to the tone of a sound signal and is independent of pitch and loudness. This attribute of sound, whilst not easy to quantify [12], allows us to differentiate between musical instruments and voices [10]. Zhang and Kuo [17] provide a detailed discussion of timbre and stated that it is an important feature in distinguishing classes of environment sound but conceded that at the same time it was very difficult to model properly or measure. Figure 2.3 illustrates the concept of timbre.

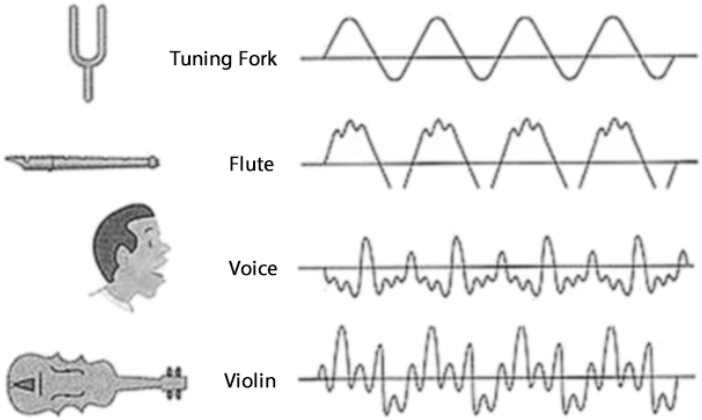

Figure 2.3: Illustration of the concept of timbre. Wave structure for each instrument is notably different. Source: https://byjus.com/physics/timbre/

2.2.3.1.4 Rhythm

Rhythm refers to features that display structural regularity of the sound signal [10]. For example, the continuous structure of a heartbeat can be referred to as its rhythm. Figure 2.4 shows the rhythmic structure of a heartbeat. In music, rhythm characterises the movement of music signals over time and contains information such as the regularity of the rhythm, the time signature and beat. It is a significant feature in the perception of sounds like footsteps, the ticking of a clock and knocking on a door [17].

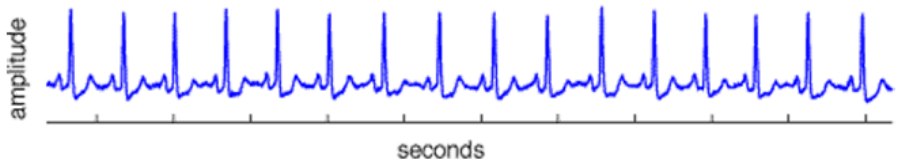

Figure 2.4: Rhythmic structure of a heartbeat. The pattern of the signal repeats itself over time. Source: http://sethares.engr.wisc.edu/htmlRT/soundexchap1.html

2.2.3.2 Physical features

Figures 2.7 to 2.14 provide visual representations of the various physical features discussed in this section. Features were extracted from an audio signal that contained gunshots, music and speech. The *pyAudioAnalysis* library, which will be discussed in Chapter 3, was used to extract the respective features. For each figure, we label each audio type as follows: a) gunshots, b) music, c) speech.

2.2.3.2.1 Fundamental frequency

Fundamental frequency (f_o) is the lowest frequency of a periodic signal or waveform [16]. In music, harmonics refer to the frequencies of vibrations within an instrument [26]. The lowest frequency produced by any musical instrument is the fundamental frequency, or the first harmonic. If we were to consider a guitar string vibrating without any driving or damping force (natural frequency), the harmonic with the lowest frequency and longest wavelength would be the fundamental frequency. The wavelength would be equivalent to twice the length of the guitar string. Figure 2.5 provides an illustration of the first seven harmonics produced by a vibrating guitar string.

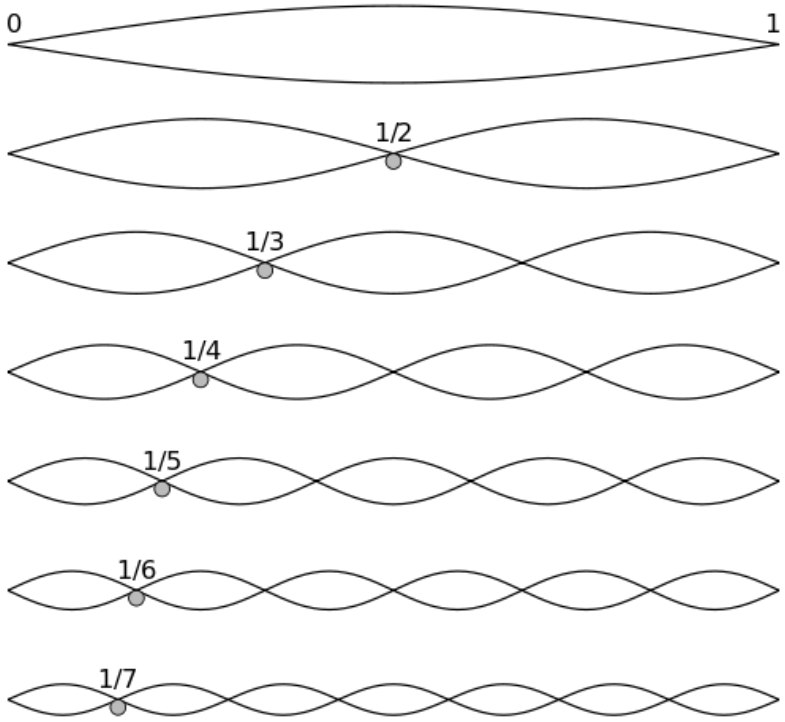

Figure 2.5: The first seven harmonics, with the first harmonic being the fundamental frequency, produced by a vibrating guitar string.
Source: https://commons.wikimedia.org/wiki/File:Harmonic_partials_on_strings.svg

Fundamental frequency is only relevant for signals that are periodic or pseudo-periodic [12]. Periodic audio signals refer to signals that repeat indefinitely, while pseudo-periodic signals almost repeat. Fundamental frequency can be defined as follows.

If T is the period of a waveform for the following equation:

$$x(t) = x(t + T) \text{ for } t \in \mathrm{R}$$

where:

$x(t)$ is the value of the waveform at t; then

$$f_o = \frac{1}{T}$$

Fundamental frequency is most useful when observing how a sound signal changes over time and has multiple applications in audio signal classification. It is effective in the detection of word boundaries, as shown by Rao and Srichland [27] and also in music detection and discrimination [25].

2.2.3.2.2 Zero-crossing rate

Subramanian, et al. [10] define zero-crossing rate (ZCR) as how often the audio signal amplitude changes from the positive spectrum to the negative, or vice-versa (crosses 0) within a given frame. Figure 2.6 illustrates the concept of zero-crossing for an audio signal.

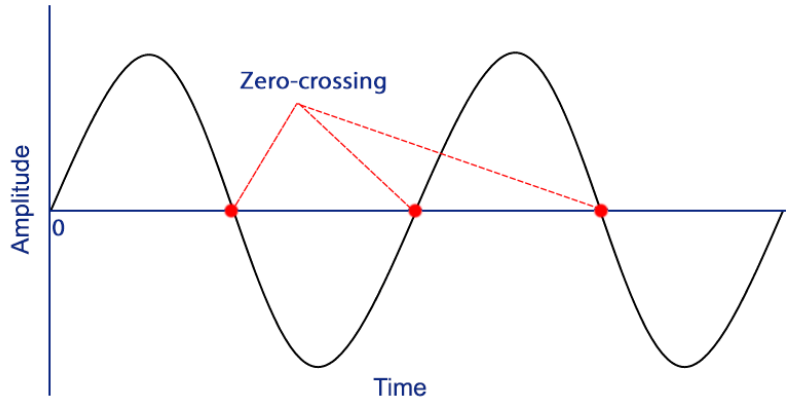

Figure 2.6: Concept of zero-crossing for an audio signal

ZCR is calculated as follows for frame x_r of length N:

$$\mathrm{ZC}_r = \frac{1}{2}\sum_{n=1}^{N} |\mathrm{sign}(x_r[n]) - \mathrm{sign}(\mathrm{x}_r[\mathrm{n}-1])|$$

where:

r refers to the number of the current frame;

$x_r[n]$ refers to the frame in the time domain, where n is the time index; and the sign function is defined by:

$$\text{sign}(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$

ZCR provides a good indication of the spectral content of a signal. According to Gerhard [16], ZCR was initially used as a means to determine the fundamental frequency of a signal but has subsequently proved to be an effective feature in itself. ZCR is an integral component in classification systems where voice/music discrimination is important [19, 28].

Figure 2.7 shows the zero-crossings rate of a series of successive analysis frames for the audio signal containing gunshots, music and speech. We see very distinctive ZCR patterns for each of the different sound types.

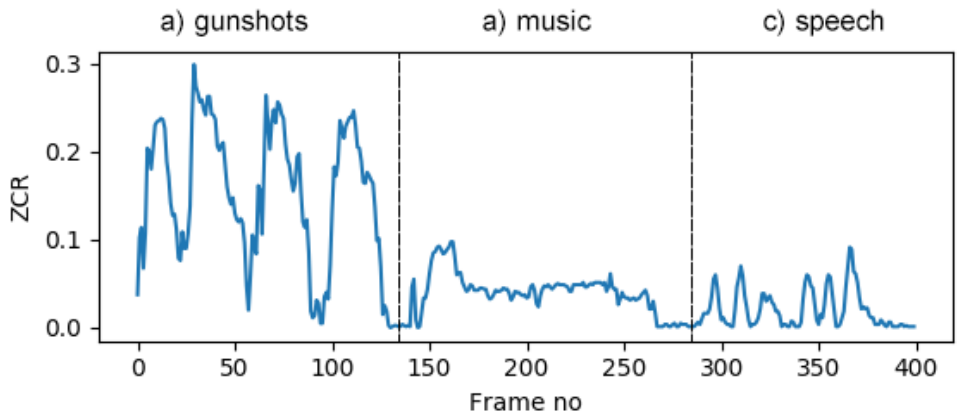

Figure 2.7: ZCR for an input signal containing a) gunshots, b) music and c) speech.

2.2.3.2.3 Energy

This is a measure of the quantity of signal at any given time [12]. The energy of an audio signal is calculated on a short-time basis. This is accomplished by the application of a window function on the signal at a given time, squaring the samples and then taking the average [17]. Zhang and Kuo [17] provide the following formula for energy:

$$E_m = \sum_m (x(n)W(n-m))^2$$

where:

m is the time index of the short-time energy,
$x(n)$ is the discrete time audio signal,
$W(n)$ is the window (frame) of length N where n = 0,1, 2,…, N -1

Zhang and Kuo [17] also state that, in speech signals, energy is the basis for discriminating between voiced components from un-voiced components. Furthermore, energy can be used to detect the presence of silence in a signal [12]. Energy and loudness are related [16]. Therefore, energy is directly proportional to the amplitude of a sound wave. Figure 2.8 provides a visual representation for the change in energy. Once again, we can clearly distinguish between the three audio types. We see sudden changes in energy for a) gunshots, b) music producing a relatively flat change in energy, and c) speech producing small spikes in energy.

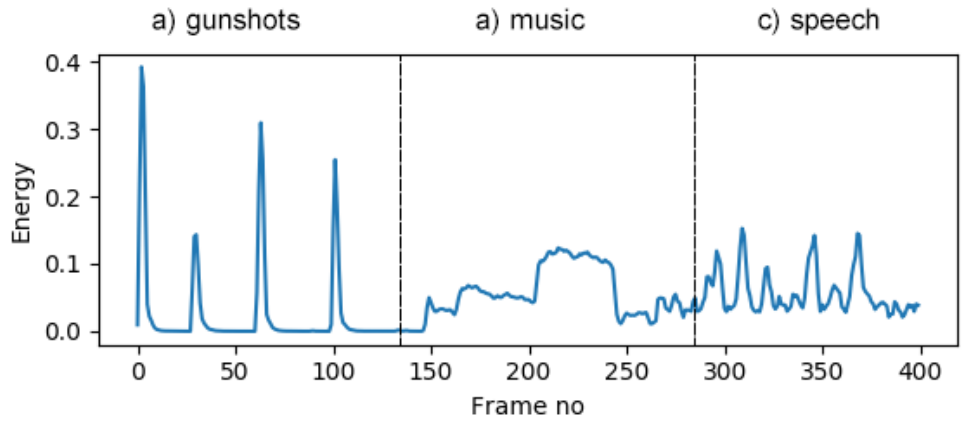

Figure 2.8: The change in energy of an input signal that contains a) gunshots, b) music and c) speech.

2.2.3.2.4 Entropy of energy

This is a measure of abrupt changes in an audio signal [14]. To calculate the entropy of energy I_r, the analysis frames are further segmented into K sub-frames, which are of a fixed duration. Thereafter, the normalised energy (σ_i^2), which is the energy of each sub-frame i, is divided by the energy of the entire frame. From this, the entropy of energy for frame r is calculated as follows:

$$I_r = -\sum_{i=1..K} \sigma_i^2 \log_2 \sigma_i^2$$

Ekštein and Pavelka [29] stated that noise signals have the highest entropy, while periodic signals like speech have relatively lower entropy values. They therefore concluded that entropy is a significant feature in signal processing and has application in speech recognition and voice activity detection. Figure 2.9 shows the energy entropy sequence for the audio signal mentioned earlier. Once again, we can clearly see the abrupt changes in energy for the gunshots. Music is relatively flat, while speech produces small spikes.

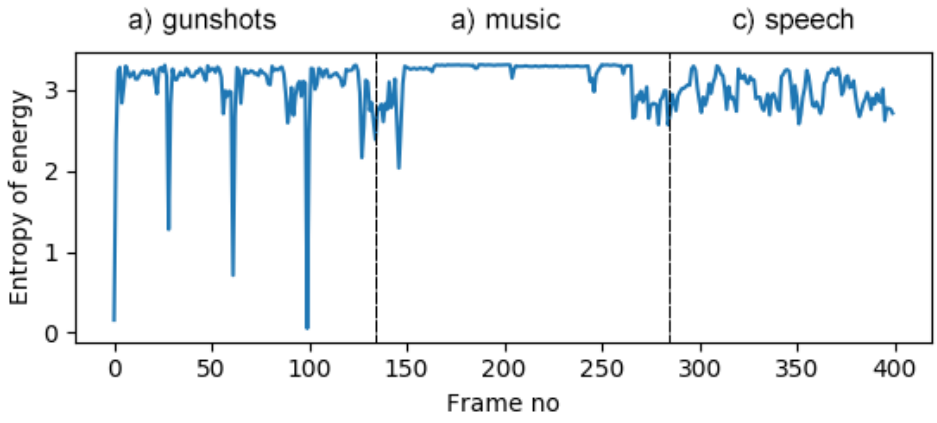

Figure 2.9: Entropy of energy for an audio signal containing a) gunshots, b) music and c) speech.

2.2.3.2.5 Spectral centroid

Spectral centroid is a measure of the spectral shape or the average frequency of the signal [12]. It is also referred to as the “centre of gravity of the spectrum” [14] or the “balancing point of the spectral power distribution” [19]. According to Burred and Lerch [13], spectral centroid is calculated as follows:

$$C_r = \frac{\sum_{k=1}^{N/2} f[k] \mid X_r[k] \mid}{\sum_{k=1}^{N/2} \mid X_r[k] \mid}$$

where:

r refers to the number of the current frame;
N is the number of Full Fourier transform (FFT) points;
$X_r[k]$ denotes the short-time Fourier transform of frame x_r; and
$f[k]$ is the frequency at bin k.

Spectral centroid provides a good indication of whether or not the spectrum has a high concentration of low or high frequencies [30]. High values indicate high frequencies and low values indicate low frequencies. It is an effective feature for audio classification tasks [10], for example voiced/unvoiced speech discrimination and music/speech discrimination [12]. In Figure 2.10 we see that the spectral centroid sequence produced for gunshots has high values while music and speech are relatively lower. This means that the gunshots produce high frequencies, while music and speech produce lower frequencies.

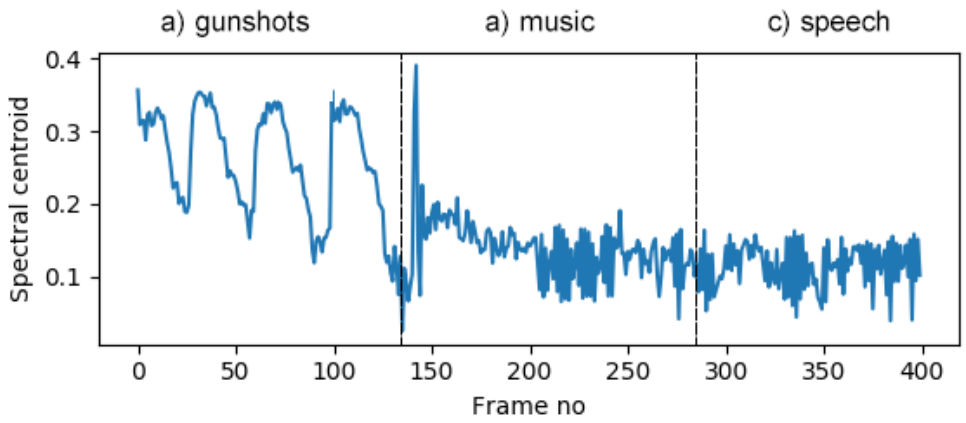

Figure 2.10: Spectral centroid for an input signal containing a) gunshots, b) music and c) speech.

2.2.3.2.6 Spectral spread

Jia-Ching, et al. [30] and Burred and Lerch [13] define spectral spread as a measure of how the spectrum is concentrated around the centroid (centre of gravity). Low values indicate that the spectrum is highly focused around the centroid, while high values indicate that it is spread largely on either side of the centroid. Burred and Lerch [13] define spectral spread with the following equation:

$$SS_r = \sqrt{\frac{\sum_{k=1}^{N/2}[\log_2\left(\frac{f[k]}{1000}\right) - \mathrm{ASC}_r]^2 P_r[k]}{\sum_{k=1}^{N/2} P_r[k]}}$$

where:

r refers to the number of the current frame;

N is the number of Full Fourier transform (FFT) points;

$f[k]$ is the frequency at frequency bin k;

P_r is the spectral power at frame r; and

ASC_r is defined as:

$$ASC_r = \frac{\sum_{k=1}^{N/2} \log_2\left(\frac{f[k]}{1000}\right) P_r[k]}{\sum_{k=1}^{N/2} P_r[k]}$$

Figure 2.11 shows the spectrum for gunshots, music and speech. While music and speech show a relatively similar spread, we see a vastly different pattern for gunshots. This is expected as this is an effective feature when discriminating between tone-like and noise-like sounds [30].

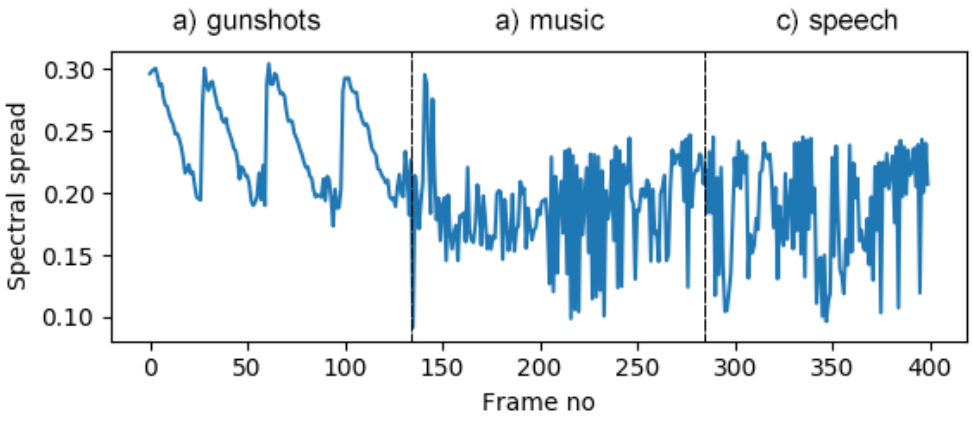

Figure 2.11: Spectral spread for an input signal containing a) gunshots, b) music and c) speech.

2.2.3.2.7 Spectral flux

Tzanetakis and Cook [15] define this feature as a measure of the rate of change in the local spectrum between successive frames. It is determined by the squared difference between the normalised magnitudes of successive frames, across one analysis window [13].

$$F_r = \sum_{k=0..S-1} (N_{r,k} - N_{r-1,k})^2$$

where:

$N_{r,k}$ is the energy of the *r*-th frame for the *k*-th sample.

Spectral flux has been found to be an effective feature when discriminating between music and speech [15, 19, 31]. There is however some discrepancy between the authors' findings, as Lie, et al. [31] stated that the spectral flux values for speech are higher than that of music, while the other two studies claimed the opposite. The

spectral flux curve in Figure 2.12 shows that speech does produce relatively higher values than music or gunshots.

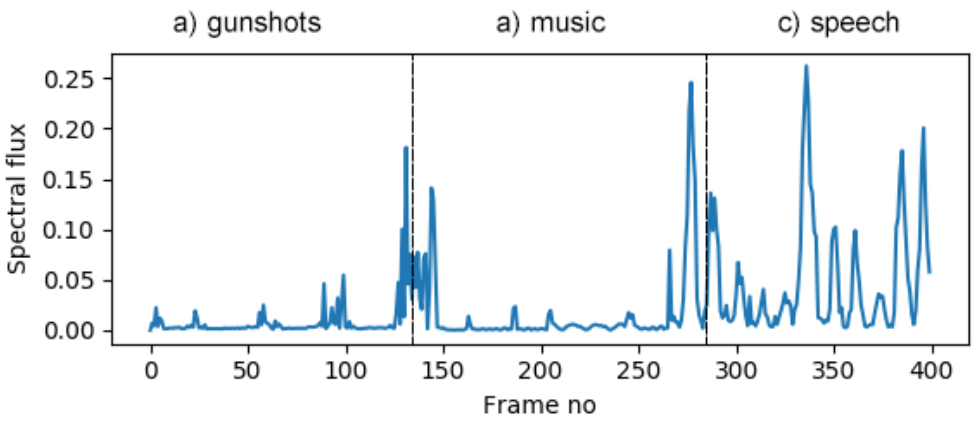

Figure 2.12: Spectral flux curve of an input signal for a) gunshots, b) music and c) speech.

2.2.3.2.8 Spectral rolloff

This is defined by some authors as the frequency below which 85% of the magnitude distribution of the spectrum is concentrated [13, 15], while others such as Scheirer and Slaney [19] define it as 95% of the power spectral distribution. However, both agree that it is also a good measure of spectral shape. It measures the skewness of the spectral shape, with brighter sounds producing higher values [18]. According to Burred and Lerch [13] spectral rolloff is a useful feature when discriminating between voiced and unvoiced speech. Burred and Lerch [13], define this feature as follows:

$$\sum_{k=1}^{M} | X_r[k] | \leq 0.85 \sum_{k=1}^{N/2} | X_r[k] |$$

If M is the largest value for frequency bin index k, for which the above equation is satisfied, then the spectral rolloff is $R_r = f[M]$

where:

$f[M]$ is the frequency at the largest frequency bin M.

Figure 2.13 confirms the statement made by Giannakopoulos, et al. [18], with gunshots (bright sounds) producing high values.

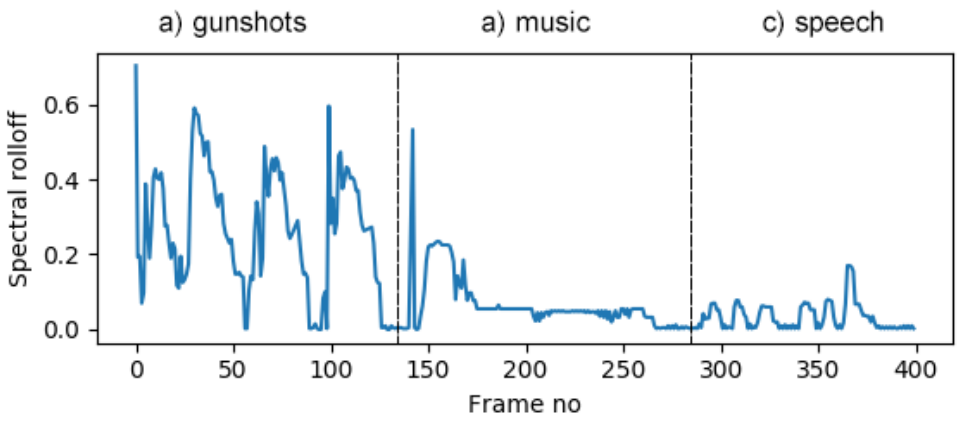

Figure 2.13: Spectral rolloff for an input signal containing a) gunshots, b) music and c) speech.

2.2.3.2.9 Mel frequency cepstral coefficients

Mel frequency cepstral coefficients (MFCC) is a representation of an audio signal's spectrum considering the non-linear perception of pitch by humans as described by the mel scale [13]. The mel scale refers to a scale of pitches that are of equal distance from each other [10]. Subramanian, et al. [10] further state that MFCC are one of the most used features in speech recognition. Studies have also confirmed that MFCC are also effective in representing music signals [32].

Figure 2.14 shows a visual representation of 13 MFCC for the same input signal. We see that graphs are relatively flat for music, with some spikes for gunshots and speech.

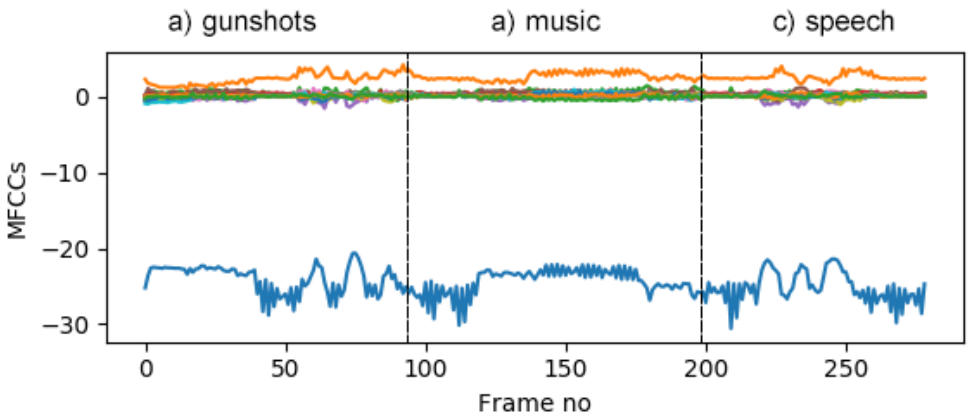

Figure 2.14: MFCC for an input signal containing a) gunshots, b) music and c) speech.

The process involved in extracting or creating MFCC for speech consists of 5 steps [32] as illustrated in Figure 2.15.

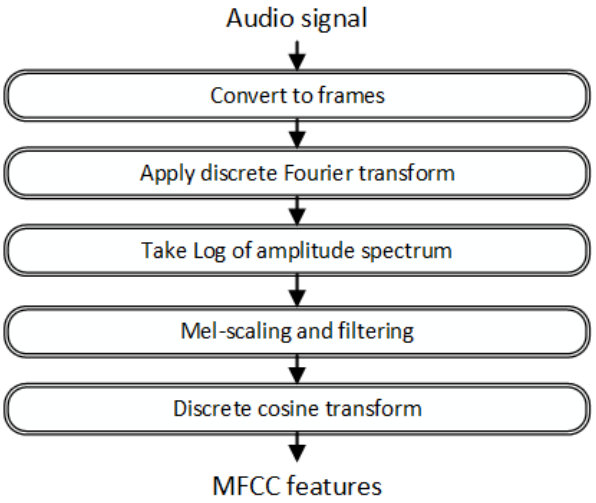

Figure 2.15: Steps involved in MFCC feature extraction. Source: [32]

Firstly, the audio signal is broken down in multiple frames or windows by the application of a windowing function. Thereafter, the discrete Fourier Transform is applied to each frame. Next the logarithm of the amplitude spectrum is taken as the perceived loudness of an audio signal is said to be approximately logarithmic [32]. The next step involves the smoothing of the spectrum resulting in 40 filter values per frame simulating the frequency perception of the human hearing system. Then the logarithm

of the coefficients is taken, and a discrete cosine transform (DCT) is applied to decorrelate them. Typically, 13 of the resulting coefficients are used for speech recognition [15].

2.2.4 Classification models

Once the feature selection has been completed, the input signal needs to be assigned a class. An efficient classification model is fundamental to any type of classifier. Depending on the level of classification required, a typical classification system would utilise a single model. In complex classification systems however, where hierarchical classification is required, multiple classification models can be combined to form hybrid or multi-class classification strategies [33, 34]. Some of the common classification models used in ASC include Hidden Markov Model, k-Nearest Neighbour, Gaussian Mixture Models and Support Vector Machine.

2.2.4.1 Hidden Markov Model (HMM)

A Markov Model (MM) is a stochastic model with a finite set of states, which have some form of measure or property (observable event), and a set of transitions between states [12]. There is a related probability for each state, and the system proceeds from state to state based on the current state and the probability of transition to a new state [35]. Figure 2.16 provides an example of a Markov process, which has three states (*Stat1*, *Stat2* and *Stat3*), and 3 corresponding observations (*Obs1*, *Obs2* and *Obs3*). The model present finite states, with a probabilistic transition between states. Given a sequence of observations, for example: *Obs1-Obs3-Obs3*, one would be able to determine the state sequence that formed the sequence of observations was *Stat1-Stat3-Stat3*. The probability of the sequence is the product of the transitions, which is 0.05 (0.2 x 0.5 X 0.5).

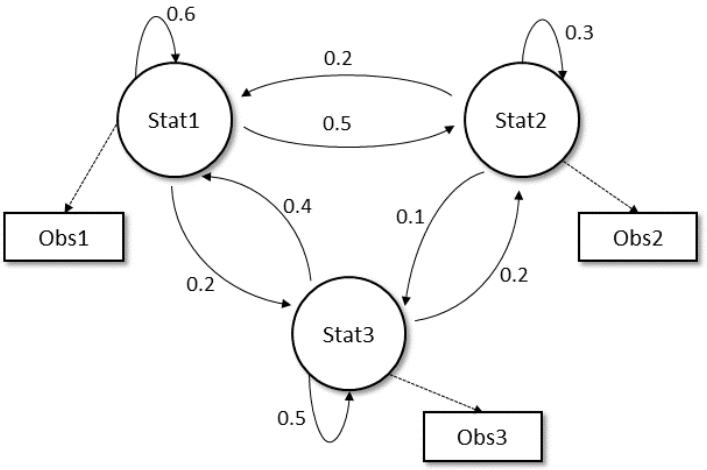

Figure 2.16: Markov process with three states (Stat1, Stat2, Stat3) and three observations (Obs1, Obs2, Obs3). The selected state transitions and their associated probabilities are indicated by arrows.

A Hidden Markov Model (HMM) is where the state sequence is "hidden" [36]. To explain this statement, we refer to Figure 2.17, which is a modification of the original Markov model presented in Figure 2.16. In the new model (Figure 2.17) all observation symbols are allowed from each state, with a probability. Therefore, if we were to consider the earlier observation sequence (*Obs1-Obs3-Obs3*), we are now unable to determine the exact state sequence responsible for the observation sequence, hence the state sequence is "hidden". According to Blunsom [36], even though the exact state sequence cannot be determined, the probability that the model produced the sequence, and the state sequence that most probably produced the observations, can be calculated.

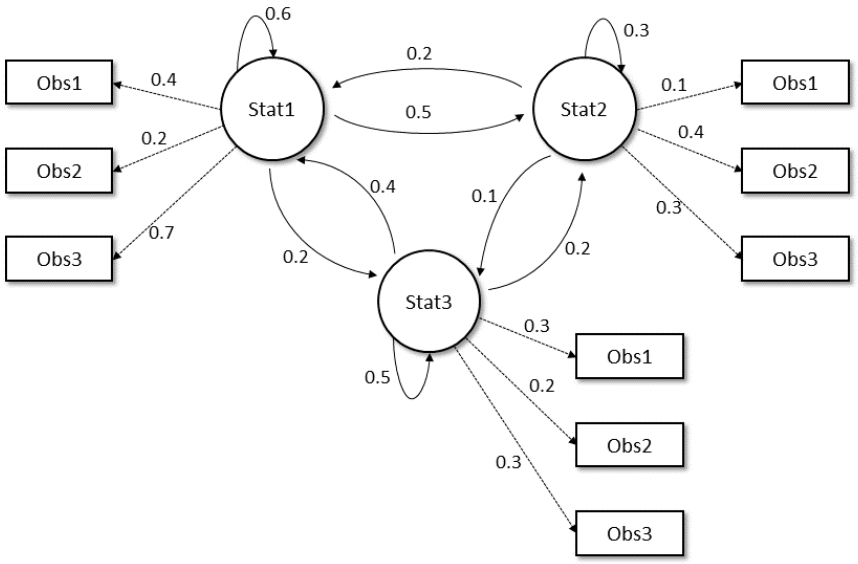

Figure 2.17: HMM with three states and three probabilistic observations. State transitions and their probabilities are indicated by arrows. Adapted from Blunsom [36].

According to Rabiner [35], there are three basic problems of interest that must be solved to make this model suitable in real-world applications.

1. Firstly, given an HMM model and a sequence of observations, what is the probability that the sequence was generated by the model?
2. Secondly, what is the optimal state sequence used by the model to generate the observation sequence?
3. Lastly, how can the model parameters be adjusted to optimise the probability of an observation sequence?

Rabiner [35] also addressed these problems and provided possible solutions.

When used in audio classification, the input signal is treated as an observation, and the HMM classifier tries to determine which HMM could possibly produce that observation/signal [12]. The classification system should contain several HMMs, each representing a specific category. The audio class that corresponds to the HMM and is most likely capable of producing the input signal is then interpreted as the class to which the input signal belongs.

Although HMM has contributed significantly to audio classification and speech recognition, there are some inherent limitation of this statistical model for speech. Rabiner [35] mentions the following limitations:

- The assumption that successive observations or frames of speech are independent.
- The assumption that distributions of individual observation parameters can be well represented as a mixture of Gaussian densities.
- The assumption that the probability of being in a state at a specific time *t* is solely dependent on the state at time *t* – 1, because dependencies generally extend through multiple states for speech sounds.

2.2.4.2 *k*-Nearest Neighbour (*k*-NN)

According to Cover and Hart [37], *k*-NN is the simplest classification procedure when there is limited prior knowledge of the data distribution. It is a non-parametric pattern recognition method utilised in both classification and regression [38]. This method of classification involves labelling an input feature vector according to the class of the training vectors that are closest to it in the feature space [10]. *K*-NN classification therefore consists of two stages. Firstly, the nearest neighbours are determined, and secondly the class for input feature vector is determined based on the nearest neighbours.

To explain the concept of *k*-NN, we refer to Figure 2.18 below, where each of the samples other than Sample *a* has been classified as *X* or *O*. In a *k*-NN classification model, the *k* nearest (closest) neighbours (samples) near Sample *a* would be used to assign a classification label. Assignment of the classification label follows a “majority-voting” rule [39], which states that the classification label assigned should be that which occurs most among the nearest neighbours.

If *k* = 1, as indicated by the blue circle in Figure 2.18, the label nearest to Sample *a* is *O*, therefore Sample *a*, which is unknown, would be assigned label *O*. However, if *k* = 5, as indicated by the red circle in Figure 2.18, then there are two samples with label *O* and three samples with label *X* that are nearest to Sample *a*. By *X* being in the majority, it would therefore be assigned to Sample *a*.

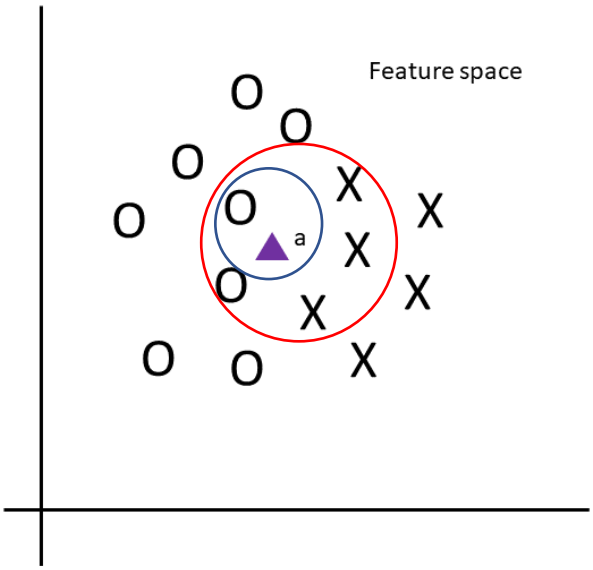

Figure 2.18: Illustration showing a 1-nearest neighbour (indicated by the blue circle) and 5-nearest neighbour (indicated by the red circle) classification decision.

The above example also illustrates two important considerations of this model. Firstly, it is assumed that the *k* neighbours have similar influence on the predictions regardless of their relative distance from Sample *a*. Therefore a suitable distance metric needs to be defined [38]. Secondly, the performance of this model is highly dependent on the selection of *k*. When *k* is small, estimates can be very poor due to data sparseness and noise, resulting in a non-linear model, while large *k* values result in linear models [10]

While *k*-NN is a simple and easily implementable classification model, Imandoust and Bolandraftar [38] highlighted some of its limitations such as poor runtime performance given a training set that is large, high computational costs and high sensitivity to irrelevant features.

2.2.4.3 Gaussian Mixture Model (GMM)

Subramanian, et al. [10] defines a Gaussian Mixture Model (GMM) as a weighted sum of Gaussian probability density functions, referred to as Gaussian components of the model, that describe a class. Gaussian probability density functions are generally bell-shaped curves and are defined by parameters such as mean and variance. Figure

2.19 illustrates this concept, in which the solid line represents the linear combination of the three separate Gaussian distributions (dotted lines).

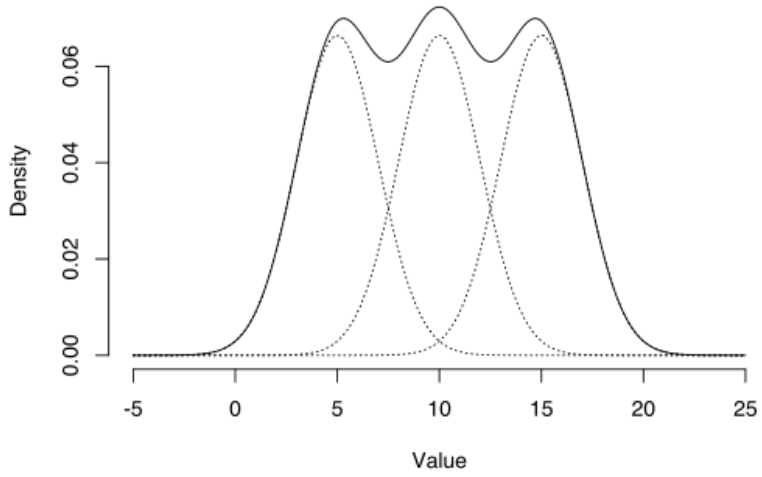

Figure 2.19: An example of a Gaussian mixture, illustrating how complex distributions can be modelled by a mixture of Gaussian distributions. Source: https://commons.wikimedia.org/wiki/File:Gaussian-mixture-example.svg

In the context of data classification, a GMM classifier models each class as a combination of Gaussian densities [13]. Each class k is represented by the following multidimensional conditional density:

$$p(\boldsymbol{x}|w_k) = \sum_{m=1}^{M} w_{km} p_{km}(\boldsymbol{x})$$

where:

w_k is the event that belongs to class k;
$\boldsymbol{x}$ denotes a feature vector;
w_{km} are the weights of the mixture;
M is total number of densities or components in the mixture; and
p_{km} is the normal density

$p(\boldsymbol{x}|w_k)$ which is also referred to as the conditional density is the likelihood of class k in respect to x [13].

GMM based classification systems are a popular approach for speaker recognition systems [40, 41], because Gaussian components have been shown to represent some basic speaker-dependent spectral shapes and Gaussian mixtures are also capable of modelling arbitrary densities [42]. GMMs are also a popular choice for speech recognition systems and noise-tracking applications [43].

While GMMs are a popular choice for the above-mentioned systems, there are limitations to this model. Yu and Deng [43] state that GMMs are statistically ineffective when modelling data that cannot be represented by linear-hyperplanes.

2.2.4.4 Support Vector Machine (SVM)

SVM is a family of machine-learning algorithms, originally developed for 2-class or binary discriminant learning [44]. SVMs function by finding a suitable boundary in the feature space to discriminate between the two classes [45]. This optimal decision boundary, or separating hyperplane, maximises the margin of separation between the closest points of the classes [46]. The points that lay on the margin boundaries are called support vectors.

To understand this concept, we refer to Figure 2.20a, which represents a 2-class (*Class 1* and *Class 2*) classification problem. The blue squares represent *Class 1* and red circles *Class 2*. A decision boundary is represented by the separating hyperplane. The three points to be classified are points *A*, *B* and *C*. Point *A*, when compared to points *B* and *C*, is farthest from the decision boundary, therefore a prediction could be confidently made that the value is *Class 2*. Conversely, point *C* is extremely close to the decision boundary. While it may be on the side of the boundary on which we would predict *Class 2*, a minor change in the decision boundary could cause the prediction to be *Class 1*. Therefore, we would be more confident of the prediction at point *A* than *C*. Point *B* lies in-between these cases. Therefore, given a training data set, an optimal separating hyperplane or decision boundary, with a maximum margin, is required that would allow confident predictions to be made as indicated in Figure 2.20b.

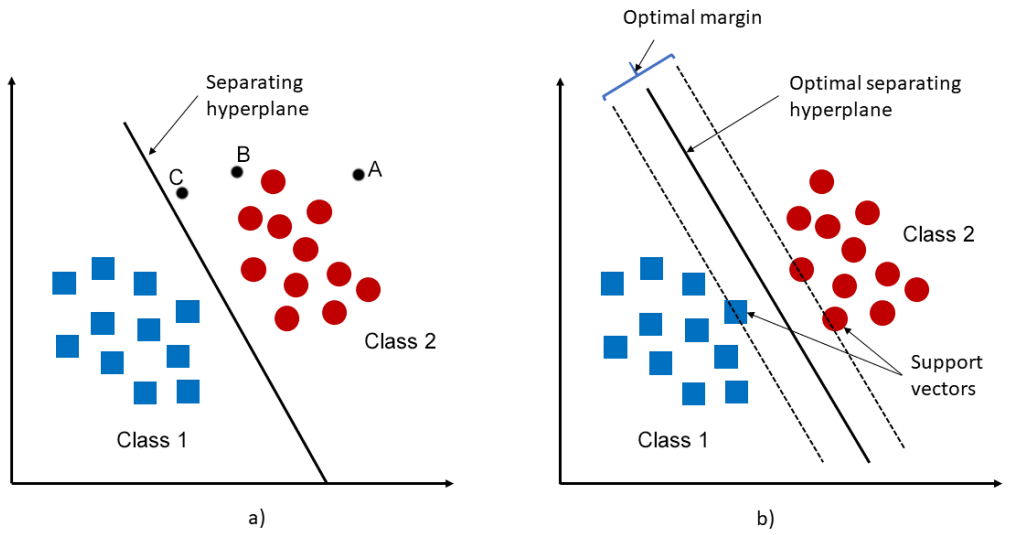

Figure 2.20: SVM separating Class 1 from Class with a separating hyperplane or decision boundary as in (a), and at the point where the margin is greatest (optimal margin) as in (b).

For multi-class (more than 2) classification, it involves decomposing the multi-class problem into a series of 2-class problems, which then can be addressed by multiple SVMs [47]. For example, if *x* is the number of classes, the SVM algorithm is run *x*(*x*-1)/2 times for each possible pair of classes, and then allocated a point. The class that receives the most points of all the 2-class SVMs is the chosen class (winner).

While SVMs are among the best performing machine-learning algorithms with regards to accuracy [48], there are limitations to its efficiency. Size and speed in both the training and testing phases is said to be a limiting factor [49]. While the speed in the testing phase has been mostly solved, the training times for large datasets is still problematic [49].

2.3. Facets of audio classification

Audio signal classification is a diverse research field. From the earliest versions of speech detection [50] and speech-music discrimination [28], to content-based retrieval systems [51] as well as video segmentation and classification systems [52], all are founded on the principles of ASC.

2.3.1 Speech and speaker recognition

According to Gerhard [12], interest in ASC, from a research perspective, was to address the problems associated with speech classification such as speech recognition and speaker recognition.

2.3.1.1 Speech recognition

Speech recognition involves the conversion of a speech signal into a sequence of words by an algorithm or computer. Research into speech recognition has been conducted for many decades, with some of the earliest contributions dating back to the early 1950s. A milestone contribution in isolated word recognition was by Atal and Rabiner [50], who proposed a pattern recognition approach to determine if a speech signal should be classified as voiced speech, unvoiced speech, or silence. This was based on the measurements of five features, namely: ZCR, energy, autocorrelation coefficient, first predictor coefficient from a linear predictive coding (LPC) analysis, and the energy of the prediction error. LPC is a popular technique in speech analysis that uses a linear combination of the past time-domain samples, for example, $s[n-1]$, $s[n-2],\ldots, s[n-M]$, to predict a current time-domain sample $s[n]$ [53]. This is explained by the following equation:

$$s[n] = -\sum_{i=1}^{M} a_i s[n-i]$$

where:

$s[n]$ is the predicted sample, and
a_i and $i = 1,2,\ldots,M$ are referred to as the predictor or LPC coefficients.

The classification model utilised by Atal and Rabiner [50] was based on a minimum non-Euclidian distance rule assuming that the parameters measured had a distribution that was in line with a multidimensional Gaussian probability density function.

The 80s saw HMMs become a popular classification choice in speech recognition [35] and a shift in focus to continuous speech (natural speech) recognition [54, 55]. Numerous advancements with pattern recognition techniques followed, with

discriminative and kernel based (SVM) methods growing in popularity [56]. Other recent studies have seen authors explore emotional speech recognition [57] where the emotional state of a speaker can be determined from their voice.

Three approaches have been proposed for speech recognition, namely, the acoustic phonetic approach, the pattern recognition approach and the artificial intelligence approach [58].

The acoustic phonetic approach was initially proposed by Hemdal and Hughes [59], and suggests that spoken language consists of a finite set of distinctive phonetic units or phenomes, which are broadly characterised by sets of properties that are revealed in the speech signal over time. It involves the segmentation and labelling of the speech signal into acoustic phonetic units. A problem that this approach faces is that there is a high degree of variation in the phonetic properties of the signal between speakers and neighbouring sounds [60].

The pattern recognition approach does not involve any feature extraction or segmentation. This method has just two steps: speech pattern training, and pattern recognition through pattern comparison [58]. This approach is founded on a well formulated mathematical framework that establishes consistent speech pattern representations, in the form of a statistical model, for example HMM, for pattern comparison. To determine the classification of an unknown, a direct comparison is made between the unknown utterances (speech) and each pattern learned during the training stage.

The artificial intelligence approach is based on concepts of both the acoustic phonetic and pattern recognition methods [60]. It utilises a robust system for segmentation and labelling and neural networks for learning the relationships between phonetic patterns and inputs.

According to Tran [60], the pattern-recognition approach has become the major method for speech recognition due to the simplicity of use, high performance and robustness to varying acoustic phonetic realisations. A key element in this approach is the use of statistical models such as HMMs to model patterns instead of a fixed template.

2.3.1.2 Speaker recognition

Speaker recognition is the process of distinguishing who is speaking based on information obtained from a speech signal. There are two approaches to speaker recognition: text-dependent and text-independent [61]. In text-dependent speaker recognition systems, the speaker is required to utter a prescribed piece of text. While there is no such requirement in text-independent speaker recognition, where utterances are said to be unconstrained, there are some other limitations such as the length of what is spoken [62].

Speaker recognition can involve either verification or identification. Speaker verification refers to the use of a machine to verify an individual's claimed identity from their speech signal [63]. For example, in a voice activated access control system, an identity claim is made by an unknown speaker. An utterance from this unknown speaker is compared with a model for the speaker based on the identity claim. Only if a match is made above a certain threshold, is the claim accepted. In speaker identification, there is no identity claim as the system decides if a speaker is a specific person or whether they belong to a certain group by determining which of the voices known by the system best matches the input voice sample [63].

The earliest speaker recognition systems date back to the early 1960s when Pruzansky [64] proposed a pattern matching method for automatic recognition of talkers. The utterances from 10 talkers were converted into time-frequency-energy patterns, where some of each talker's utterances were used to form reference patterns and some for test patterns. Recognition was determined by cross-correlating the test patterns with the reference patterns, thereafter selecting the talker corresponding to the reference pattern with the highest correlation. Recognition scores of 89% were reported in this study.

Atal [65] presented an overview of speaker recognition, listing suitable parameters (features) of a speech signal that could be used for speaker recognition. These included energy, pitch, short-time spectrum, predictor coefficients, timing, the rate of speaking and formant frequencies, which refer to the resonant frequencies of the vocal tract [66].

Other early work includes Furui [67], who proposed using a set of time-based functions obtained from acoustic analysis of fixed sentence-long utterances. For this, the author opted to extract predictor coefficients, which were then transformed into cepstrum coefficients, by means of LPC analysis. Very low mean error rates were reported.

Recently, the focus has been to improve robustness. Pelecanos and Sridharan [68] proposed a feature mapping approach that constructs a stronger representation of all cepstral feature distributions, thereby enhancing recognition robustness during adverse environmental conditions. Other techniques include “RelAtive SpecTrA” (RASTA), that extracts important information from the modulation spectrum [69], and Normalised Dynamic Spectral Features (NDSF) which is a spectral feature set that was introduced by Chougule and Chavan [70] for mismatch conditions in speaker recognition. The authors noted that NDSF enhance robustness by a reduction in additive noise and channel effects, generally caused by sensor mismatch.

2.3.1.3 Challenges with speech and speaker recognition

Some of the common problems that both speech and speaker recognition systems face are noise and speaker variability, which is influenced by accents. Both problems degrade the performance of such systems.

Many authors have contributed new methods to improve speech detection in noisy environments. Ramirez, et al. [71] proposed an algorithm that measures the long-term spectral divergence between speech and noise. It then determines the speech/non-speech choice by comparing the long-term spectral envelope to the average noise spectrum. Germain, et al. [72] proposed a voice activity detection (VAD) method that was founded on non-negative matrix factorisation. They trained a universal speech model from a corpus of clean speech (without noise) and did not include a noise model. The speech model was robust enough to detect speech in a variety of noisy audio signals.

To investigate the impact of accent on speech recognition, Arslan and Hansen [73] used a 20-word isolated speech database, where a HMM classifier was trained with five tokens of each word, from speakers of American English. They tested the model using American English, with people born in America, as well as second language

American English speakers from Turkey, Germany and China. The recognition rate obtained was 99.7%, 92.5%, 88.7% and 95.3% respectively, thus confirming that accent does impact speech recognition.

Responding to this, they extracted spectral and energy based features, which were then used to develop an HMM based accent classification algorithm. Both mono-phone and whole word models were considered, with the latter capturing accent information more efficiently. Their classification system was able to correctly identify accents from 4 classes (accents), with 93% accuracy. Kumpf and King [74] also proposed an automatic classification system of foreign accents for Australian English. The classification system, based on “accent dependent parallel phoneme recognition” was developed to process an input containing continuous speech and then distinguish between native Australian English and foreigners such as Lebanese and Vietnamese, speaking English. The average accuracy for accent classification was 85.3%.

2.3.2 Speech and music discrimination

A popular area of interest in audio segmentation and classification is speech and music discrimination, where the purpose is to analyse a given audio signal and segment the signal according to speech and music. Research into speech and music discrimination saw Saunders [28] propose a classification technique that provided real-time discrimination of speech and music from broadcast FM radio, while Scheirer and Slaney [19] presented a classification system that was capable of distinguishing speech from music, over a wide array of digital audio input. A further technique was put forward by El-Maleh, et al. [75] who proposed a robust narrowband speech and music discrimination system. While these authors performed studies in real-time speech and music discrimination, each utilised different classification techniques.

In a relatively simplistic approach, Saunders [28], focussed on just two physical features: the ZCR and energy of the audio signals, with a multivariate GMM.

Contrary to Saunders (1996), Scheirer and Slaney [19] used 13 different features in their application: 4Hz modulation energy, percentage of “low energy” frames, spectral rolloff, spectral centroid, spectral flux, ZCR, cepstrum resynthesis residual magnitude (CRRM), pulse metric and the variances of spectral rolloff, spectral centroid, spectral

flux, ZCR and CRRM. CRRM refers to the 2-norm of the vector residual post cepstral analysis and smoothing, while pulse metric is a feature that uses autocorrelation to determine the amount of rhythm that exists within a 5-second frame [19]. The authors evaluated four different classification models in their study: a simple Gaussian classifier (GS), two variants of *k*-NN and a GMM.

In their approach, El-Maleh, et al. [75] used line spectral frequencies (LSF), which provide alternate representations of LPC coefficients, as the core feature set. They also introduced a new feature, the linear prediction zero-crossing ratio (LP-ZCR), which they defined as the "ratio of the zero-crossings count (ZCC) of the input and the ZCC of the output of the linear prediction analysis filter." They utilised 4 features in total, namely: LSF, differential line spectral frequencies (DLSF), line spectral frequencies with higher order crossings (LSF-HOC) and line spectral frequencies with LP-ZCR (LSF-ZCR). DLSF are definied by successive differences of the LSF. Higher order crossings (HOC) refer to the ZCC of a filtered signal. El-Maleh, et al. [75] also compared *k*-NN classifiers against GMMs.

Although different audio classification techniques were employed, all three studies produced above 90% accuracy rates. Table 2.1 provides a high-level comparison of these respective studies. From the results of these studies it becomes apparent that it is not the number of features selected that is important, but the selection of a set of specific features to achieve a certain outcome.

Authors	**Features used**	**Classifier**	**Accuracy**
Saunders (1996)	ZCR, energy	Multivariate GMM	90%
Scheirer & Slaney (1997)	Spectral rolloff, spectral centroid, spectral flux, ZCR, 4Hz modulation energy, percentage low energy frames, CRRM, pulse metric, variance of (spectral rolloff, spectral centroid, spectral flux, ZCR and CRRM)	GS, *k*-NN, GMM	93.2%
El-Maleh et al. (2000)	LSF, LSF-ZCR, LSF-HOC, DLSF	*k*-NN, GMM	95.9%

Table 2.1: Comparison of studies focussed on speech and music discrimination. Even though each study differed in their approach, all achieved +90% accuracy.

Wyse and Smoliar [76] expanded on the concept of music and speech discrimination by also including speaker discrimination. The initial step was to separate the audio signal into music or speech. Music discrimination was based on the average length of time in which peaks exist in a narrow frequency range. Finally, they used a combination of changes in pitch, timing cues and spectral features to determine the transition of speakers.

Kimber and Wilcox [77] included more classes in their contribution, where cepstral coefficients were selected for features and GMMs together with HMMs were used as classification models to segment audio into speech, music, laughter and non-speech. Results from their studies showed that the segmentation and classification model proposed fared relatively well against manual hand labelling.

Other work involving music classification includes instrument classification. Ubbens and Gerhard [78] proposed an instrument classification system strictly using a time-domain feature set. The authors claimed that features extracted from the time-domain are not typically used in classification as they can be unreliable at times. However, they have a lower computational cost than the frequency domain. In their study, they compared their time-domain based classification model against spectral and MFCC based models. Even though the time-domain based model did not produce better results, it was very comparable.

2.3.3 Content-based retrieval systems

With the rapid growth of audio and other multimedia data, there is a demand for efficient and automated content-based retrieval of audio from multimedia databases [79, 80]. Attempting to retrieve audio data utilising pure text-based retrieval mechanisms can prove to be a daunting task as metadata can be subjective and therefore never completely reliable. Content-based retrieval systems were introduced to address these shortcomings of existing database models with regards to storage, indexing and retrieval of audio and other multimedia data [81]. Content-based retrieval systems provide a richer experience, allowing users to query multimedia databases more efficiently. The “Muscle Fish Database” [25] allows users to search for audio data using the following methods:

- **Simile**: saying one sound is like another. For example, “like the sound of a flock of seagulls.”
- **Acoustic/perceptual features**: describing the sound by using common physical distinctive features such as brightness, pitch and loudness.
- **Subjective features**: using personal descriptions to describe sounds. For example, “a thunderous sound”.
- **Onomatopoeia**: attempting to make a sound similar in some quality to a sound you are searching for. For example, “making a chirp-chirp sound to find birds”.

Therefore, ASC would be a fundamental component of such a system for two simple reasons [22]:

- Different types of audio should be processed differently.
- The “search space” after classification is restricted to a specific class during retrieval, thereby improving efficiency.

Zhang and Kuo [82] proposed a hierarchical audio content analysis and classification system, which they claimed would archive audio data more appropriately for efficient retrieval. This system was divided into three stages of implementation. The first stage involved “coarse-level” classification where simple features such as energy, ZCR and fundamental frequency were used to classify audio signals into basic classes of speech, music, environmental sounds and silence. Further classification of each basic class was carried out in the next stage. The authors referred to this level of classification as “fine-level classification”. For the fine-level classification, features were extracted from the time-frequency representation of the audio signal to show minor differences in timbre, pitch and change pattern for the different classes. The chosen classifier was HMM and a single model was built for each class. In the final stage of implementation, an audio retrieval system was built with two retrieval approaches: query-by-example and query-by-keywords.

In the following year, Zhang and Kuo also introduced a real-time audio segmentation and classification scheme for content-based audio management that classified audio signals into basic classes such as speech, music, song, silence and speech with background music [83]. Once again, they opted for simple audio features such as ZCR, the energy function, fundamental frequency and spectral features for their

classification system. Statistical and morphological analysis for temporal curves of the selected features were performed to distinguish the different types of audio.

In additional research, Zhang and Kuo also devised a content-based audio retrieval system that showed two stages of classification [17]. In the first stage, audio signals were classified into high-level categories such as speech, music and noise by analysing the short-term features of the signal. These were then further classified into finer classes such as rain, applause and bird sounds. For this, the authors analysed the time-frequency of the audio signal and utilised a HMM. In the above-mentioned studies by Zhang and Kuo, classification performance was above 90% accuracy.

Srinivasan, et al. [20] reported an accuracy of greater than 80% for their classification approach that could detect and classify audio comprising mixed classes such as combinations of music and speech together with background or environmental audio. They too isolated simple features such as the average energy and average ZCR.

Li [84] presented a method using a combination of perceptual features such as brightness, bandwidth and energy, together with MFCC. He also introduced a new method for pattern classification called New Feature Line (NFL). This method gathers information within multiple prototypes per class by utilising linear interpolation and extrapolation of each pair of prototypes in the class. He reported that this new method outperforms other pattern classification methods like Nearest Neighbour (NN).

An accuracy rate of greater than 96% was also reported by Lie, et al. [31]. They presented an audio segmentation and classification approach that segmented and classified audio signals into speech, music, environmental sound and silence by introducing new features such as noise frame ratio and band periodicity, which were shown to have been extremely effective in discriminating different audio types. Another innovation that Lie, et al. [31] contributed was real-time automatic speaker segmentation.

Guodong and Li [51] improved on the work presented by Li [84] by introducing two new elements: the inclusion of a new metric, called distance-from-boundary (DFB), for content-based audio retrieval, and utilising SVM as the classification model instead of NFL. The authors reported a marked improvement over NFL and other popular classification models such as *k*-NN, with an error rate of only 8.1% when classifying

198 sounds into 16 classes. This illustrates an important point that the choice of classifier is extremely important given the same feature set.

Chien-Chang, et al. [85] expanded on Guodong and Li [51], by incorporating additional wavelet functions and a bottom-up SVM. They reported a reduced feature set and an improvement in the classification error rate reported by Guodong and Li [51].

Table 2.2 provides a high-level comparison of the above-mentioned studies and their contributions to content-based retrieval.

Authors	**Methodology**	**Outcome/s**
Zhang & Kao 1998	Utilised ZCR, F_0 and energy.	A multi-level, hierarchical classification model.
Zhang & Kao 1999a	Utilised ZCR, F_0, energy and spectral features.	A new segmentation and classification method for content-based audio management.
Zhang & Kao 1999b	Utilised same features as Zhang & Kao (1998).	A two-level classifier for content-based audio retrieval.
Srinivasan et. al 1999	Utilised average energy and average ZCR.	A mixed-class classification model.
Li (2000)	Utilised MFCC and perceptual features. (Example: brightness, bandwidth and energy).	A new method for content-based audio classification and retrieval utilising a new pattern classifier called nearest feature line (NFL).
Guodong and Li (2003)	Utilised similar features as Li (2000).	Showed marked improvement in classification over Li (2000) by using a new metric (DFB) and using SVM instead of NFL.
Chien-Chang et. al (2005)	Same as Guodong & Li (2003).	Improved error rate of classification by including wavelet functions and a bottom-up SVM classifier.

Table 2.2: Comparison of studies in content-based audio retrieval.

2.3.4 Video segmentation, classification and indexing

While studies have demonstrated that the information provided by audio classification and segmentation is invaluable to understanding the content of audio signals, they have also shown that it can be utilised in understanding and analysing video content.

Patel and Sethi [23] proposed extracting audio features such as pitch, ZCR, spectrogram and average magnitude from the sub-band level of the MPEG encoded audio streams for video indexing.

There were two studies that incorporated audio analysis in the segmentation and classification of television programs that were presented. Liu, et al. [86] utilised a neural network classifier with 12 audio features to obtain an overall accuracy rate of 86.8% in discriminating between news reports, commercials, weather forecasts, football games and basketball games. Liu, et al. [87] improved on the results and presented an 11.9% increase in accuracy by utilising HMM with the same experimental setup.

Boreczky and Wilcox [88] proposed a technique for video segmentation using HMM. Features utilised for segmentation were not exclusively image-based but were also motion and audio based. Whereas other studies involving the use of audio in video segmentation have classified audio into different classes, the authors chose to instead calculate an audio distance measure, which is the distance calculated between adjacent intervals of audio. A further difference in this study was that the authors did not classify video and audio features separately, but rather combined them within the HMM framework.

Zhang and Kuo [89] proposed a system that performed automatic segmentation and classification of audio-visual data using audio content analysis. They classified audio into classes such as speech, music, song, environmental sound, speech with music background, environmental sound with music background and silence. For this, they extracted and analysed audio features such as the short-time energy function, short-time average ZCR, spectral peaks and the short-time fundamental frequency. While traditional frameworks focus entirely on visual cues or changes, such as histogram

differences and motion vectors, their system included the audio classification to provide segmentation and indexing that was semantically correct.

To aid with efficient searching of e-learning content repositories, Ying and Chitra [90] proposed an SVM based technique to segment and classify the audio from instructional videos according to seven audio classes, namely speech, silence, music, environmental sound, speech with music, speech with environmental sound, and environmental sound with music. Twenty-six audio features, that could capture the spectral and temporal differences for the seven classes, were chosen. These included mean variance of ZCR, mean short-time energy and mean spectral flux. An accuracy rate of 97.9% was reported.

Baillie and Jose [52] segmented recorded soccer matches into important events such as goal scoring, goal attempts, cautions or card issuing by the umpire. This was achieved by analysing the levels of crowd response during a soccer match. They utilised features such as MFCC

within a HMM framework.

To distinguish violent content from non-violent content in movies, Giannakopoulos, et al. [18] proposed an SVM binary classification system that analysed the audio signal. Audio such as speech and music indicated non-violence while audio such as gunshots, screaming and explosions indicated violent scenes. Features that were utilised included energy entropy, signal amplitude, short-time energy, ZCR, spectral flux and spectral rolloff. An accuracy rate of 85% was reported.

2.4 Summary

Audio signal classification has given rise to numerous other research interests such as audio segmentation and classification, content-based audio retrieval and video scene segmentation and classification.

Much of the research has focused on improving accuracy rates of existing classification methods by introducing new feature sets, and changing or combining classification models. The literature has shown that while feature selection is a compulsory step in any classification system, it is not the number of features, but the

selection of specific features that is fundamental to the performance. Furthermore, performance is highly dependent on the choice of classification model when the feature set remains the same.

There has been promising work done with audio classification in recent years in the audio-visual area, with audio classification being incorporated in the segmentation and classification of video data. Results from studies have shown a marked improvement over the traditional use of just image/video content for segmentation and classification. Additionally, studies have illustrated that audio analysis in the context of video segmentation and classification provides important semantic information that would normally be excluded.

3. METHODOLOGY

3.1 Introduction

The intention of this study is to investigate if audio classification could be used to classify and segment a lecture recording audio signal into classes that represent speech and chatter. These classes could thereafter be utilised to automatically detect the start and end trim points for the recorded lecture as part of the current workflow in the lecture recording process at UCT.

This chapter discusses the tools and libraries utilised, the selection and preparation of the data set, audio feature sets extracted and the classification model chosen and implemented. The chapter concludes with a discussion of the metrics used to evaluate the performance of the selected classification model and the evaluation of the trim point predictions.

3.2 Audio classification process

While most audio signal classification systems employ a variety of different principles or algorithms, they generally follow the same process. The classification system receives an audio signal, audio features are extracted and selected from the signal, which are then passed onto the classifier, which contains a particular statistical model, and the signal is then finally assigned a class. Figure 3.1 illustrates this process [10].

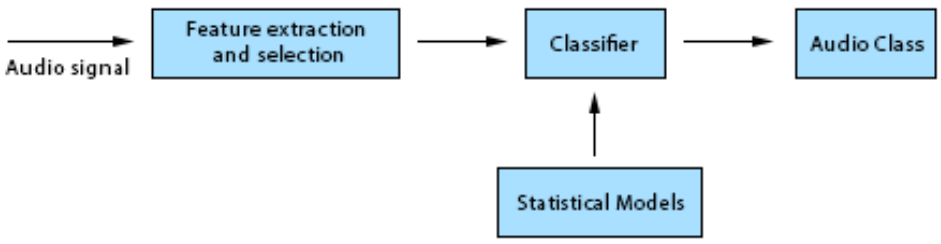

Figure 3.1: A typical audio classification process

Two classification classes were chosen for this study. They are:

1. Speech: This class represents a dominant voice, generally the lecturer presenting or addressing the students.
2. Non-speech: Chatter between students is the predominant component of this class. Chatter implies the lack of a dominant voice, with multiple people talking at the same time. Silence and environmental noise, such a squeaking door, are also included.

The above classes were chosen because a typical lecture consists of student chatter, speech from the lecturer, and on occasion silence.

3.3 Tools and libraries

As the primary goal of this study is to utilise audio classification in determining the start and end trim points for a recorded lecture, we did not develop a custom classification system, but instead decided to use existing open source applications or libraries. After reviewing freely available classification systems, an open source Python library called *pyAudioAnalysis* [14] was chosen as it proved to be a versatile library. This library provides a broad range of audio-related functionalities which include: classifying an unknown audio segment according to predefined classes, segmenting an audio file and classifying it into homogeneous segments, extracting audio thumbnails from music tracks, removing silence areas from a recording, etc. In this study, we utilise *pyAudioAnalysis* to:

- Extract audio features.
- Train a classification model.
- Perform cross-validation experimentation to extract performance metrics.
- Segment audio files to determine the trim points.

In addition to *pyAudioAnalysis*, FFMPEG was used to convert audio files from FLAC format to WAV. Furthermore, Adobe Audition CS6 was used to segment audio files into respective classes to train the chosen classifier.

3.4 Dataset and sampling

As indicated in Chapter 1, Opencast is used at UCT to manage and administer the lecture capture process. The raw audio files captured during a lecture recording were used in this study. Three media streams are currently captured when a lecture is recorded, namely:

1. a presenter or lecturer video stream from an IP camera,
2. a presentation video stream from a data projector or document camera, and
3. the audio stream from a lapel or boundary microphone.

All three media streams are saved on centralised storage in FLV, MP4 and FLAC formats, respectively. Together with associated metadata, they comprise the “media-package” for the published recording. Metadata includes but is not limited to information such as venue name and course series name.

Two datasets are used in this study. The first dataset (Dataset 1) is used to train and test the classification model, while the second (Dataset 2) is used to evaluate the algorithm that determines the start and end trim points. Dataset 1, comprises a total of 150 audio files, which were downloaded from Opencast and converted to WAV format. To ensure a good range in audio quality, the total number of audio files was spread across 10 different venues (15 audio files per venue). The training data set was manually created by editing the downloaded audio and creating segments that purely contained speech and non-speech. This resulted in a total of 6862 audio files as listed in Table 3.1.

Audio Class	Speech	Non-speech
No. of audio files	3476	3386

Table 3.1: The training dataset, consisting of 3467 samples for the speech class and 3386 samples for the non-speech class.

Dataset 2 comprises 50 additional audio files, which were also downloaded from Opencast and converted to WAV format.

3.5 Classification model

The *pyAudioAnalysis* library includes several classification models, which include k-NN, random forests, gradient boosting and SVM. The SVM model, with a linear kernel was chosen for this study. For classification, *pyAudioAnalysis* implements a cross-validation procedure to determine the optimal classifier parameter.

SVM has become a popular choice for audio classification, with multiple studies comparing its efficacy with other classifiers such as Hidden Markov Model and k-NN and concluding that it has a better performance [85, 91, 92]. SVM classification models have also been shown to be far more effective than other models when there is a training data set available [92]. Furthermore, Lu, et al. [92] showed that the computational demand for training and testing an SVM model is far less than *k*-NN, resulting in quicker training and testing experiments.

3.6 Audio features

A total of 34 audio features are extracted on a short-term basis, resulting in a sequence of short-term feature vectors of 34 elements each. A frame (window) size of 50ms and frame step of 25ms is used for the short-term feature extraction. The 25ms frame step enables a 50% overlap. Figure 3.4 illustrates the frame size and frame step utilised by *pyAudioAnalysis* during the feature extraction process.

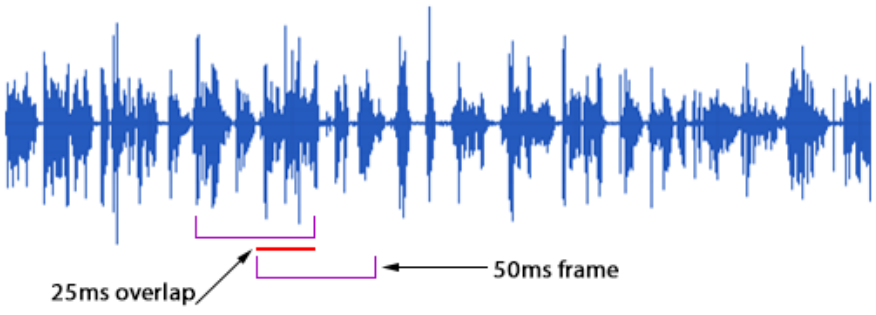

Figure 3.2: 50ms frame size and 25ms frame step for the feature extraction process.

Additionally, the feature sequence is processed on a mid-term basis. This is where the signal is first divided into mid-term segments and for each segment, short-term

processing is performed. The feature sequence from each mid-term segment is used to calculate feature statistics, for example the average value for ZCR. This means that each mid-term segment is characterized by a set of statistics. A complete list of all features is presented in Table 3.2.

No.	Feature	Description
1	Zero-crossing rate (ZCR)	Rate of sign-changes of a particular frame.
2	Energy	The sum of squares of the signal values, which are normalised by the length of the frame.
3	Entropy of energy	A measure of abrupt changes.
4	Spectral centroid	The spectrum’s centre of gravity.
5	Spectral spread	The spectrum's second central moment of the spectrum.
6	Spectral entropy	The entropy of the normalized spectral energies for a set of sub-frames.
7	Spectral flux	The squared difference between the normalized magnitudes of the spectra of the above sub-frames.
8	Spectral rolloff	The frequency below which 85% of the magnitude distribution of the spectrum is concentrated.
9-21	Mel frequency cepstral coefficients (MFCC)	A cepstral representation where the frequency bands are not linear but distributed according to the mel scale.
22-33	Chroma vector	A 12-element representation of the spectral energy in 12 equal-tempered pitch classes of western music.
34	Chroma deviation	The standard deviation of the above 12 chroma coefficients.

Table 3.2: Audio features utilised by pyAudioAnalysis.

3.7 Classification and trimming

The *pyAudioAnalysis* library also provides segmentation and classification functionality. This refers to splitting an audio signal into homogenous segments and applying a classification model on each of these segments, resulting in a sequence of class labels. Successive segments that share the same label are merged into larger segments.

We utilise this feature of the library to produce a list of segment timestamps and corresponding class labels, which are then processed by an algorithm written in Python. A link to the GitHub repository for the algorithm is provided in Appendix 1. The process to determine the start and end trim points for the audio file, is illustrated in Figure 3.5 below.

To determine the start trim point, the algorithm finds the first speech segment and utilises the corresponding timestamp. To determine the end trim point, the algorithm finds the last speech segment and utilises the corresponding timestamp.

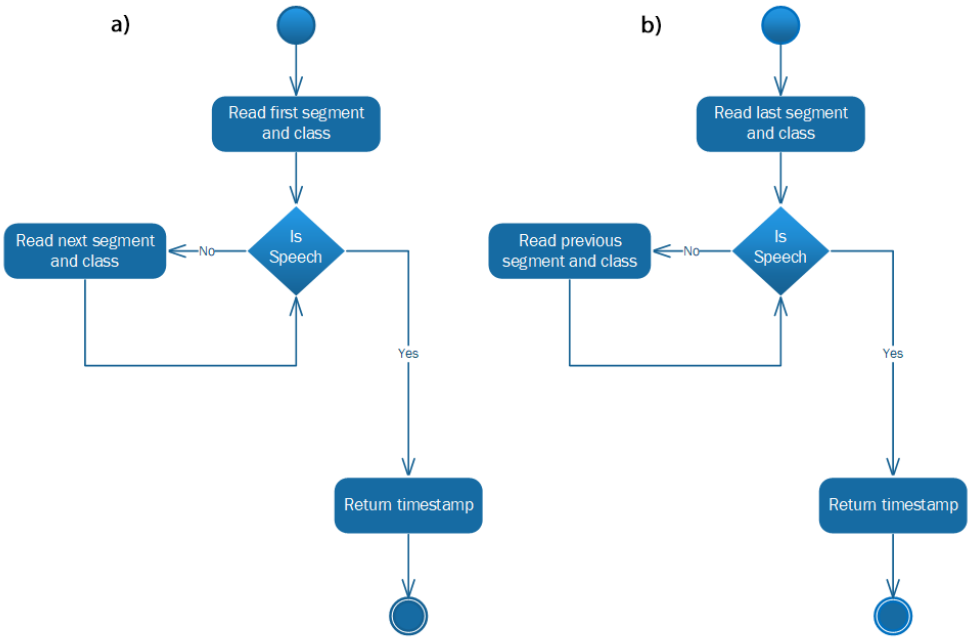

Figure 3.3: Activity diagram illustrating start and end trim point prediction. a) Algorithm returns the start trim point. b) Algorithm returns the end trim point.

3.8 Evaluation process

Evaluation is a fundamental step in any classification model. Two aspects are evaluated in this study. Firstly, we evaluate the performance of the SVM classifier using 10-fold cross-validation, which is a popular technique in evaluating predictive models in machine learning experimental design. It combines both training and testing. Thereafter, we evaluate the algorithm that predicts the start and end trim points.

3.8.1 Classification evaluation

In the 10-fold design of this study, 10 different subsets of equal size were created by partitioning the dataset. The procedure involved training and testing the SVM model 10 times. For each iteration of the test, it involved training on nine of the subsets, and testing on one. The results from the 10 experiments were entered into a confusion matrix. Thereafter, metrics were generated to determine the performance of the SVM classifier model.

For the evaluation process, we utilise the methodology employed by Shaikh, et al. [93]. In their study on the performance evaluation of classification methods for heart disease, they utilised evaluation metrics such as Precision, Recall, Accuracy and F-measure.

These 4 metrics can be generated from a confusion matrix. According to Subramanian, et al. [10], a confusion matrix is used to evaluate the performance of an audio classification system by counting the cross-validation instances that are predicted correctly and incorrectly. This matrix can be utilised as the basis for accuracy analysis as it shows if a particular class has been incorrectly classified as another [10]. In the confusion matrix presented in Table 3.3, the columns represent actual speech and non-speech, while the rows represent what the SVM classifier predicted as speech and non-speech. Audio samples that are correctly predicted as speech are True Positives (TP), while those that are correctly predicted as non-speech are True Negatives (TN). False Positives (FP) represent instances where non-speech is incorrectly predicted as speech. False Negatives (FN) represent instances where speech is incorrectly predicted as non-speech.

	Actual speech	Actual non-speech
Predicted speech	TP	FP
Predicted non-speech	FN	TN

Table 3.3: Confusion matrix for speech and non-speech

Using this matrix, the accuracy of the classifier can be determined by the proportion of misclassified audio files. This means that the smaller the proportion of misclassified audio files, the greater the accuracy of the classifier. As mentioned earlier, evaluation metrics are also derived from this confusion matrix. These will be discussed next.

3.8.1.1 Precision

This is the measure of the proportion of the correctly predicted speech audio to all the audio predicted as speech. Shaikh, et al. [93] defines precision as the positive predictive value (PPV), as expressed in the following equation,

$$Precision = \frac{TP}{TP + FP}$$

3.8.1.2 Recall

Recall, also known as sensitivity, is the probability that speech can be identified by the classifier, as expressed in the following equation.

$$Recall = \frac{TP}{TP + FN}$$

3.8.1.3 Accuracy

This is a common measure and has been used in many studies in audio classification [20, 82, 94]. While it is a very common metric, it is often used in conjunction with other metrics as it can be misleading [95] when there is a large class imbalance, as equal weighting is assigned to both false positives and false negatives. Accuracy is defined

as the proportion of correctly predicted speech and non-speech, and is expressed in an equation as follows.

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

3.8.1.4 F-Measure

This is a combined metric. According to Shaikh, et al. [93] F-measure is a weighted score and is determined by the harmonic mean of precision and recall. It therefore determines the efficacy of the classifier in predicting a particular class by utilising both precision and recall. As each class is handled individually, it is a preferred measure when there is an imbalance in datasets. F-measure is expressed in an equation as follows.

$$F - measure = \frac{2\,(precision \times recall)}{precision + recall}$$

The metrics discussed above, namely precision, recall, accuracy and F-measure, were used to rate the performance and efficacy of the SVM classification model utilised by *pyAudioAnalysis*.

3.8.2 Trim point evaluation

For this study, we utilised audio files from lecture recordings that have already been published. Opencast has a record of the original media files as well as the trim points that were set during the editing and trimming stage (manual trimming). This information is saved in Synchronised Multimedia Integration Language (SMIL) format and we use the trim points in these files as gold standard data. To evaluate the algorithm, we compare the predicted trim points of the 50 audio files in Dataset 2 to gold standard data, and plot the average error. This would provide a good indication of how the predicted trim points deviate from recordings trimmed during the manual editing and trimming stage of the lecture recording process.

3.9 Summary

This chapter has discussed the methodology followed in preparing the dataset used in this study. A total of 6862 audio files were used to train and test the SVM classification model using two audio classes: speech and non-speech. It also provided a comprehensive list of 34 features utilised by the *pyAudioAnalysis* library during the classification process. The chapter also included an overview of the steps involved in evaluating the performance of the chosen classification model, explaining the metrics utilised in the evaluation process. Furthermore, we introduced the algorithm utilised in determining the trim points and the process in evaluating its performance.

4. RESULTS AND DISCUSSION

4.1 Introduction

Two experiments are carried out in this study. Firstly, we determine the performance of the SVM classification model by performing 10-fold cross validation on our data set of 6862 audio files. Secondly, using the segmentation and classification functionality of the *pyAudioAnalysis* library, we determine the trim points of 50 audio files and compare these to gold standard data. Data for these experiments was obtained as outlined in Chapter 3. The chapter begins with Section 4.2, where we present and discuss the performance metrics for the SVM classification model. Thereafter, we discuss the performance of the trim point prediction algorithm in Section 4.3. The chapter is then concluded with Section 4.4, where we discuss some considerations, should the proposed solution be implemented.

4.2 SVM classification model performance

Combining the 10 tests of the 10-fold cross validation produced the confusion matrix as depicted in Table 4.1.

	Actual speech	**Actual non-speech**
Predicted speech	3376	44
Predicted non-speech	100	3342

Table 4.1: Confusion matrix for speech to determine the performance of the SVM classification model utilised by the pyAudioAnalysis library.

The results indicate that 3376 audio files were correctly identified as speech and 3342 were correctly identified as non-speech. There were 100 speech files that were incorrectly identified as non-speech, and 44 non-speech files that were incorrectly identified as speech. Using the metrics that follow, we present the analysis of the performance of the classification model.

4.2.1 Accuracy

This is the ratio of the sum of correctly predicted speech and non-speech (6718) to the total number of audio files in the test data (6862). This is an indication of the average performance of the classification model in correctly identifying speech and non-speech, implying that the classification model correctly classified 6718 audio files and incorrectly classified 144 audio files from the total of 6862 files. The value obtained for this metric was 97.9%. While this indicates the classification model has good accuracy, authors such as Brownlee [95] have stipulated that accuracy on its own is not a sufficient metric to measure the performance of a classification model.

4.2.2 Precision

This is the measure of the proportion of correctly predicted speech (3376) to all the audio predicted as speech (3420). Only 44 audio files from a total of 3386 from the non-speech class were incorrectly identified as speech, thus producing a value of 98.7% for this metric. The classification model utilised in this study therefore has a high precision rate when classifying audio into speech and non-speech. This high precision further indicates that this classification model does not produce a high number of false positives.

4.2.3 Recall

This is the proportion of speech that was correctly identified (3376) to the total number of actual speech files in the test data (3476). The value obtained for this metric was 97.1%. This indicates that the classification model utilised has a 97.1% probability of correctly identifying speech. This also indicates that this classification model does not produce a high number of false negatives.

4.2.4 F-Measure (F-score)

This is a metric that is most important as it focuses on how accurately the classification model predicts speech by utilising a combination of precision and recall. Both precision and recall relate to the classification of speech, which makes this a useful measure of

how effective the model is when predicting speech. The value obtained for this study was 97,9%.

4.2.5 Summary

The above metrics indicate that the classification model utilised in this study has a very high probability of correctly identifying speech from non-speech. Table 4.2 presents a summary of the results.

Performance measure	Results
Accuracy	97.9%
Precision	98.7%
Recall	97.1%
F-Measure	97.9%

Table 4.2: Summary of performance metrics obtained for the classification model utilised in this study.

The above results also compare very well against the binary SVM classifier presented by Giannakopoulos, et al. [18] and the multi-class SVM classifier presented by Siantikos, et al. [96], with the former study reporting accuracy, precision and recall values of 85.5%, 82.4% and 90.5% respectively, and the latter reporting an overall F-measure score of 73.8%.

4.3 Trim point predictions

In this experiment, we compare the predicted trim points against gold standard data, which were obtained from manually trimmed lecture recordings. The predicted trim points were determined using the segmentation and classification functionality of the *pyAudioAnalysis* library as described earlier in Chapter 3. Table 4.3 lists the differences between predicted values and gold standard data, for the start trim points and end trim points, for each of the 50 sample audio files utilised for this experiment.

Sample	Gold Std. Start (s)	Predicted Start (s)	Difference (s)	Gold Std. End (s)	Predicted End (s)	Difference (s)
1	3.000	8.000	5.000	2628.854	2924.000	295.146
2	222.444	222.000	-0.444	2490.063	2698.000	207.937
3	58.263	47.000	-11.263	2477.466	2970.000	492.534
4	55.149	57.000	1.851	2731.985	2887.000	155.015
5	55.119	127.000	71.881	2894.654	2902.000	7.346
6	1.000	18.000	17.000	2878.591	2878.000	-0.591
7	169.232	189.000	19.768	2913.994	3214.000	300.006
8	229.477	229.000	-0.477	1534.345	1539.000	4.655
9	63.139	71.000	7.861	2727.688	2721.000	-6.688
10	152.873	148.000	-4.873	2813.670	2814.000	0.330
11	108.888	109.000	0.112	2664.711	2926.000	261.289
12	300.909	82.000	-218.909	2596.455	2939.000	342.545
13	53.030	57.000	3.970	2918.170	2850.000	-68.170
14	26.940	30.000	3.060	2734.320	2736.000	1.680
15	146.928	150.000	3.072	3035.447	3043.000	7.553
16	83.007	91.000	7.993	2853.476	3012.000	158.524
17	90.971	104.000	13.029	2609.813	2604.000	-5.813
18	73.942	83.000	9.058	2910.170	2908.000	-2.170
19	29.403	31.000	1.597	2737.682	3009.000	271.318
20	32.588	40.000	7.412	2643.563	2646.000	2.437
21	76.109	82.000	5.891	2614.137	2986.000	371.863
22	121.715	121.000	-0.715	2723.598	2728.000	4.402
23	541.662	338.000	-203.662	3084.325	3229.000	144.675
24	490.704	497.000	6.296	3070.932	3075.000	4.068
25	3.000	15.000	12.000	2783.864	2771.000	-12.864
26	120.737	117.000	-3.737	2872.455	2861.000	-11.455
27	65.585	74.000	8.415	2368.240	2379.000	10.76
28	171.977	174.000	2.023	2873.211	2873.000	-0.211
29	44.066	67.000	22.934	2717.525	2723.000	5.475
30	126.288	136.000	9.712	2632.131	2936.000	303.869
31	75764	80000	4.236	2743795	2748000	4.205
32	3000	15000	12.000	2783864	2771000	-12.864
33	25922	32000	6.078	2774142	3302000	527.858

Sample	Gold Std. Start (s)	Predicted Start (s)	Difference (s)	Gold Std. End (s)	Predicted End (s)	Difference (s)
34	297845	96000	-201.845	2072113	2708000	635.887
35	1000	21000	20.000	2398307	2406000	7.693
36	1000	4000	3.000	2734430	3166000	431.57
37	52993	44000	-8.993	2816672	2818000	1.328
38	1000	0	-1.000	2679333	3005000	325.667
39	319753	105000	-214.753	2700224	2704000	3.776
40	2827	0	-2.827	3252395	3261000	8.605
41	52618	48000	-4.618	4448198	4455000	6.802
42	18832	20000	1.168	3599960	3610000	10.040
43	1000	5000	4.000	878667	1217000	338.333
44	73658	47000	-26.658	2697160	2575000	-122.160
45	88574	100000	11.426	2844665	3005000	160.335
46	318675	327000	8.325	2836076	3002000	165.924
47	18888	19000	0.112	2625981	2926000	300.019
48	1000	18000	17.000	2336376	3009000	672.624
49	54312	65000	10.688	2563862	2778000	214.138
50	1000	7000	6.000	2677404	3010000	332.596

Table 4.3: Difference between predicted and gold standard data for start and end trim points.

We then plot the differences (Figure 4.1) for both the start and end trim points to show how the predicted values deviate from gold standard (manually trimmed). The deviation of the predicted values from gold standard for the start trim points is represented by the blue line and the orange line represents the deviation of the predicted values from gold standard for the end trim points.

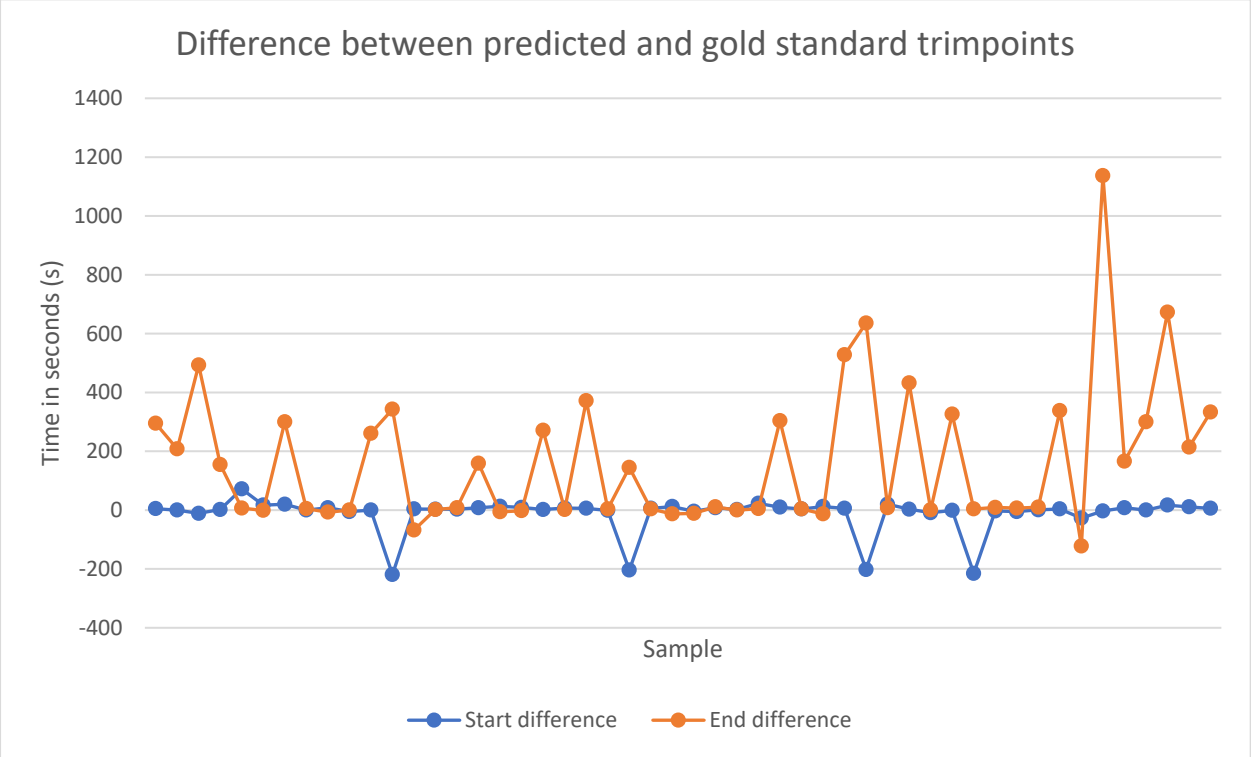

Figure 4.1: Deviation of the predicted start and end trim points from gold standard data for 50 audio files.

The standard deviation for the trim point differences are listed in Table 4.4. The values obtained were 60.52s for the start trim point differences and 193.43s for the end trim point differences.

	Start trim point	End trim point
Mean	-11.22s	145.16s
Standard deviation	60.52s	193.36s
Standard error	8.56s	27.35s

Table 4.4: Mean, standard deviation and standard error for the start trim point differences and end trim point differences as listed in Tables 4.3.

4.3.1 Evaluation of trim point predictions

We evaluate the performance by assessing the extent to which the predictions deviate from gold standard data. In Figure 4.1, each point on the graph represents a prediction and the closer a prediction is to 0 seconds, the less it deviates from the gold standard, as 0 on the Y-axis indicates an exact match. For the start trim point predictions, five out of the 50 samples (10%) were beyond a single standard deviation of 60.52s. Eighteen out of the 50 samples (36%) were beyond a single standard deviation of 193.36s for the end trim point predictions.

4.3.1.1 Start trim point predictions

The trim points showed very little deviation from gold standard data. Forty-five samples were less than 30s from gold standard data, the remaining five samples deviated as follows: Sample 5 – 71.881s after gold standard; Sample 12 – 218.909s before gold standard; Sample 23 – 203.662s before gold standard; Sample 34 – 201.845s before gold standard; Sample 39 – 214.753s before gold standard data. To understand the discrepancy with the predictions that displayed a high deviation, we listened to the sample audio files and viewed the untrimmed lecture recording to confirm if speech was classified correctly, or if there was a misclassification, and recorded observations. Table 4.5 lists our findings.

Sample	Observation	Reason for discrepancy
5	First occurrence of speech was correctly identified.	Manual trim point was set at 55.119s due to human judgement.
12	First occurrence of speech was correctly identified when a student addressed the class.	Manual trim point was set at 300.909s. The student addressing the class was deemed irrelevant to the lecture.
23	First occurrence of speech was correctly identified at 338s when the lecturer chatted to a student.	Manual trim point was set at 541.662s due to human judgement.
34	First occurrence of speech was correctly identified at 96s. Dominant voice clearly present in conversation between students.	Manual trim point set when lecturer began speaking.
39	First occurrence of speech was correctly identified at 105s when lecturer addressed the class.	Manual trim point was set at 319.753s due to human judgement.

Table 4.5: Observations and reasons for the discrepancy of samples that demonstrated a high deviation from gold standard for the start trim point predictions.

4.3.1.2 End trim point predictions

The results for the end trim point predictions where not as consistent as the start trim point predictions. Twenty-five out of the 50 samples were within a range of 30s from gold standard data. The remaining 25 samples displayed high deviations. They were as follows: two samples ranged between -68s and -123s before gold standard data, and 23 samples ranged between 144s and 673s after gold standard data. To get a better understanding of the high deviations, we listened to the sample audio files and viewed the untrimmed lecture recordings. We list our findings in Table 4.6.

Sample	Observation	Reason for discrepancy
1, 2, 3, 4, 7, 12, 16, 19, 23, 30, 34, 36, 43, 47, 48, 49	The last occurrence of speech was correctly identified after gold standard data.	Discussion between lecturer and students after the lecturer was excluded due to human judgement.
11, 21, 33, 38, 45, 46, 50	The last occurrence of speech was correctly identified after gold standard data.	Student talking after lecture, therefore dominant speech. This speech segment was excluded due to human judgement.
13	The last occurrence of speech was incorrectly identified before gold standard data.	Student addresses class from the back of the classroom. Boundary microphone did not sufficiently project student voice, therefore, not detected as "dominant speech". Manual trim point, however, included this segment.
44	The last occurrence of speech was incorrectly identified before gold standard data.	Recording was of poor quality. Lecturer's voice was not very prominent due to a high presence of ambient noise.

Table 4.6: Observations and reasons for the discrepancy of samples that demonstrated a high deviation from gold standard for the end trim point predictions.

Our observations showed that the last occurrence of speech was correctly identified after gold standard data for 23 of the 25 samples. A dominant voice was present in discussions between lecturers and students or among students themselves after the lecture had concluded. Including these discussions, as identified by the automated solution, will result in videos of larger file-sizes than manually trimmed recordings, thereby impacting storage negatively. However, it does not negatively impact the quality of the recording, as we are not losing important lecture information.

In the remaining two samples, the last occurrence of speech was incorrectly identified before gold standard data. In Sample 13, the boundary microphone failed to sufficiently project a student's voice, resulting in speech not being detected. In Sample 44, there was a high amount of ambient noise present and the lecturer was not very prominent, resulting in speech not being detected.

Therefore, while the classification model has a high probability of detecting speech, there are some considerations. Firstly, it cannot discriminate between different voices. While this is outside the scope of this study, it is something that can be investigated in

a future study. Furthermore, a segment will always be classified as speech if a dominant voice is present, regardless if there is chatter in the background.

4.3.2 Summary

The start trim point predictions were very promising, with 90% of predictions within 30s of gold standard data. End trim point predictions were not as consistent. The end trim point predictions for 25 out of 50 samples were within 30s of gold standard data. Of the remaining 25 samples, there were two misclassifications, resulting in end trim points being predicted before gold standard data. Misclassifications were either due to a high amount of ambient noise or the boundary microphones not projecting a dominant voice sufficiently. For the remaining 23 samples, the presence of a dominant voice resulted in the end trim points being predicted after gold standard data. This occurred as a result of discussions amongst students, or between the lecturer and students, after the lecture had ended, which had been deemed irrelevant during the manual trimming process.

4.4 Considerations

The results of this study show that audio classification has application in automating the identification of trim points for recorded lecturers at the University of Cape Town, with some considerations.

4.4.1 Publication time and storage

As mentioned in Chapter 1, the trimming of recorded lectures adversely affects the publication time in the current lecture capture solution at UCT, as it is a manual process that is completely dependent on staff. Implementing the automated system presented in this study into the lecture capture workflow could remove the need for human intervention during the trimming stage and therefore improve publication turn-around time. However, published video files, which have been automatically trimmed, will generally have a larger file-size than those that are manually trimmed. This is because manually trimmed videos will exclude discussions between lecturers and students after the lecture concludes, whereas automatically trimmed videos would include this if a dominant voice was detected.

4.4.2 Value to students

While the results of this study have shown that the end trim point predictions are generally far greater than gold standard data, resulting in longer published videos, this is essentially not a shortfall. The inclusion of these discussions between lecturer and student(s) is not removing any value from the recording. An argument could be made that it is, in fact, adding academic value, as some of these discussions could prove beneficial to other students viewing the published recording.

4.4.3 Video download

Since published videos with automated trimming could potentially have a larger file-sizes than manually trimmed videos, video download could be negatively impacted. This is dependent on bandwidth availability.

5. CONCLUSION

5.1 Summary

Audio classification formed the fundamental theoretical framework for this study. The efficacy of audio classification in predicting the start and end trim points of recorded lectures at the University of Cape Town was investigated. A custom classification model was not developed, but instead an open source python library, *pyAudioAnalysis*, was utilised. This library provided multiple audio-related functionalities including feature extraction, classification and segmentation. Support Vector Machine (SVM) was chosen as the classifier for this study.

Two experiments were performed. In the first experiment, 150 audio files from previously recorded lectures were downloaded and segmented into speech (identified by a dominant voice) and non-speech (student chatter and other environmental sounds). Using, the segmented audio files, 10-fold cross-validation was performed to train and test the SVM classification model to discriminate between speech and non-speech. The resulting confusion matrix was then used to calculate performance metrics for the classification model.

In the second experiment, a further 50 audio files from previously recorded lectures were downloaded. The segmentation and classification functionality of the *pyAudioAnalysis* library was used to determine the start and end trim points for these audio files. To evaluate the accuracy of the predicted trim points, they were compared to gold standard data obtained from manually trimmed recordings.

5.2 Answers to research questions

Two questions were posed in Section 1.4 that was considered relevant in evaluating the outcome of this study. The answers to the proposed questions are examined as follows:

1) *How accurately can audio signal classification distinguish speech from non-speech?*

Following the methodology discussed in Chapter 3, the evaluation results discussed in Section 4.2 show that the SVM classification model has 97.8% probability of accurately distinguishing speech from non-speech. Additional performance metrics: precision, recall and F-measure; were also calculated. Values obtained were 98.7%, 97.1% and 97.9% respectively (Table 4.2). These results indicate that the SVM classification model does not produce a high number of false positives or false negatives, and thus has a very high probability of correctly distinguishing speech from non-speech.

2) *How do the start and end trim points, determined using audio classification, compare to gold standard data?*

Section 4.3 discussed the evaluation of the trim point predictions, where we noted some inconsistencies. While the start trim point predictions were predominantly within 30s from gold standard data, most of the end trim point predictions were far greater than gold standard data (Table 4.3). Upon closer inspection, it was discovered that the deviations were in most instances attributed to the presence of a dominant voice being detected post lecture, being either a private discussion between lecturer and student(s), or amongst students themselves. Therefore, although the end trim point predictions deviated greatly from gold standard data, they were predominantly technically correct.

5.3 Future work

In its current design, the classification model utilised in this study will predict the speech class if a dominant voice is present, regardless if there is chatter in the background. If we were to scrutinise audio from the discussions between the lecturer and student(s), we would see that it could be regarded as a combination of speech and chatter. Therefore, introducing another audio class (speech with chatter), and training the model accordingly, could potentially make the classifier more robust. This could result in end trim point predictions being more aligned with gold standard data.

In Chapter 4, we reported two instances of misclassification, which was mainly due to the audio signal containing large amounts of ambient noise. Therefore, the

performance of the classification model with low quality audio samples warrants further investigation.

The list of features utilised in this study, as listed in Chapter 3, were predefined by the *pyAudioAnalysis* library. While these features have proven to be sufficient in discriminating speech from non-speech, and adequate research has been involved in their selection for general purpose audio signal analysis [14] , the investigation and inclusion of additional features could provide value and possibly improve performance with low quality audio signals.

The inclusion of additional trim points could benefit the existing design as these could be used to exclude other segments that provide no value to the final recording, for example, when classes break for an interval. The inclusion of this feature could not only decrease the file-size of the published video but also maintain the continuity and flow of the lecture. This would also limit the total length of published video students would need to skim through, should they choose to search for a particular point in the recorded lecture.

Furthermore, the audio classification system could potentially be enhanced by combining it with synchronized visual cues from the video recording. This could possibly increase classification performance as the literature has indicated.

Finally, other classification algorithms such as k-NN and HMM could be investigated and their efficacy within UCT’s lecture capture solution evaluated.

6. REFERENCES

[1] S. Cardall, E. Krupat, and M. Ulrich, "Live lecture versus video-recorded lecture: are students voting with their feet?," *Academic Medicine,* vol. 83, pp. 1174-1178, 2008.

[2] S. Davis, A. Connolly, and E. Linfield, "Lecture capture: making the most of face-to-face learning," *engineering education,* vol. 4, pp. 4-13, 2009.

[3] S. K. A. Soong, L. K. Chan, C. Cheers, and C. Hu, "Impact of video recorded lectures among students," in *Proceedings of the 23rd Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education: Who's learning? Whose technology?*, 2006, pp. 789-793.

[4] J. Williams and M. Fardon, "On-Demand Internet-Transmitted Lecture Recordings: Attempting to Enhance and Support the Lecture Experience," presented at the 12th International Conference of the Association for Learning Technology, Manchester, England, 2005.

[5] C. McInnis and R. Hartley. (2002, 17 August 2017). The impact of full-time study and paid work on the undergraduate experience in Australian universities. [Online]. Available: http://www.cshe.unimelb.edu.au/research/equity/docs/eip02_6.pdf

[6] M. J. Anderson, "Degree of fit: University students in paid employment, service delivery and technology," *Australasian Journal of Educational Technology,* vol. 22, p. 88, 2006.

[7] M. Fardon, "Internet streaming of lectures: A matter of style," in *Proceedings of Educause Australasia*, 2003.

[8] K. Woo, M. Gosper, M. McNeill, G. Preston, D. Green, and R. Phillips, "Web-based lecture technologies: blurring the boundaries between face-to-face and distance learning," *ALT-J,* vol. 16, pp. 81-93, 2008.

[9] M. Ketterl, O. A. Schulte, and A. Hochman, "Opencast matterhorn: A community-driven open source solution for creation, management and distribution of audio and video in academia.," in *International Symposium on Multimedia*, 2009, pp. 687-692.

[10] H. Subramanian, P. Rao, and S. Roy, "Audio signal classification," M-Tech, Electrical Engineering Deptartment, Indian Institute of Technology, Bombay, 2004.

[11] M. McKinney and J. Breebaart, "Features for audio and music classification," in *Conference Proceedings of the International Society of Music Information Retrieval*, 2003, pp. 151-158.

[12] D. Gerhard, *Audio signal classification: History and current techniques*: Citeseer, 2003.

[13] J. J. Burred and A. Lerch, "Hierarchical automatic audio signal classification," *Journal of the Audio Engineering Society,* vol. 52, pp. 724-739, 2004.

[14] T. Giannakopoulos, "PyAudioAnalysis. A Python library for audio feature extraction, classification, segmentation and applications," *PloS one,* vol. 10, p. e0144610, 2015.

[15] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing,* vol. 10, pp. 293-302, 2002.

[16] D. Gerhard, "Audio signal classification: an overview," *Canadian Artificial Intelligence,* vol. 45, pp. 4-6, 2000.

[17] T. Zhang and C. C. J. Kuo, "Hierarchical classification of audio data for archiving and retrieving," in *International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 3001-3004.

[18] T. Giannakopoulos, D. Kosmopoulos, A. Aristidou, and S. Theodoridis, "Violence content classification using audio features," in *Advances in Artificial Intelligence, Lecture Notes in Computer Science* vol. 3955, ed: Springer, 2006, pp. 502-507.

[19] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1331-1334 vol.2.

[20] S. Srinivasan, D. Petkovic, and D. Ponceleon, "Towards robust features for classifying audio in the CueVideo system," in *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, Orlando, Florida, USA, 1999, pp. 393-400.

[21] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America,* vol. 8, pp. 185-190, 1937.

[22] L. Guojun and T. Hankinson, "A technique towards automatic audio classification and retrieval," in *Proceedings of the Fourth International Conference on Signal Processing*, 1998, pp. 1142-1145.

[23] N. V. Patel and I. K. Sethi, "Audio characterization for video indexing," in *Storage and Retrieval for Still Image and Video Databases IV*, 1996, pp. 373-384.

[24] J. Breebaart and M. F. McKinney, "Features for audio classification," in *Algorithms in Ambient Intelligence*, ed: Springer, 2004, pp. 113-129.

[25] E. Wold, T. Blum, D. Keislar, and J. Wheaten, "Content-based classification, search, and retrieval of audio," *IEEE MultiMedia,* vol. 3, pp. 27-36, 1996.

[26] The Physics Classroom, "Fundamental Frequency and Harmonics," 1996. [Online]. Available: http://www.physicsclassroom.com/class/sound/Lesson-4/Fundamental-Frequency-and-Harmonics. [Accessed: 15 January 2018].

[27] G. V. R. Rao and J. Srichland, "Word boundary detection using pitch variations," in *International Conference on Spoken Language Processing*, Philadelphia, USA, 1996, pp. 813-816.

[28] J. Saunders, "Real-time discrimination of broadcast speech/music," in *International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, USA, 1996, pp. 993-996.

[29] K. Ekštein and T. Pavelka, "Entropy and entropy-based features in signal processing," in *Proceedings of PhD workshop systems & control*, 2004.

[30] W. Jia-Ching, W. Jhing-Fa, H. Kuok Wai, and H. Cheng-Shu, "Environmental Sound Classification using Hybrid SVM/KNN Classifier and MPEG-7 Audio Low-Level Descriptor," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, 2006, pp. 1731-1735.

[31] L. Lie, H. Zhang, and J. Hao, "Content analysis for audio classification and segmentation," *IEEE Transactions on Speech and Audio Processing,* vol. 10, pp. 504-516, 2002.

[32] B. Logan, "Mel Frequency Cepstral Coefficients for Music Modeling," in *Proceeding of the International Symposium on Music Information Retrieval (ISMIR)*, Plymouth, USA, 2000.

[33] K. Umapathy, S. Krishnan, and R. K. Rao, "Audio signal feature extraction and classification using local discriminant bases," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 15, pp. 1236-1246, 2007.

[34] J. Weston and C. Watkins, "Support vector machines for multi-class pattern recognition," in *Proceedings of the Seventh European Symposium on Artificial Neural Networks*, 1999, pp. 219-224.

[35] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE,* vol. 77, pp. 257-286, 1989.

[36] P. Blunsom, "Hidden markov models," 2004. [Online]. Available: http://digital.cs.usu.edu/~cyan/CS7960/hmm-tutorial.pdf. [Accessed: 15 October 2017].

[37] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory,* vol. 13, pp. 21-27, 1967.

[38] S. B. Imandoust and M. Bolandraftar, "Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background," *International Journal of Engineering Research and Applications,* vol. 3, pp. 605-610, 2013.

[39] P. Cunningham and S. J. Delany, "k-Nearest neighbour classifiers," *Multiple Classifier Systems,* vol. 34, pp. 1-17, 2007.

[40] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models," *Digital Signal Processing,* vol. 10, pp. 19-41, 2000/01/01/ 2000.

[41] D. A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Communication,* vol. 17, pp. 91-108, 1995/08/01/ 1995.

[42] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing,* vol. 3, pp. 72-83, 1995.

[43] D. Yu and L. Deng, *Automatic speech recognition: A deep learning approach*: Springer, 2014.

[44] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning,* vol. 20, pp. 273-297, September 01 1995.

[45] A. Mammone, M. Turchi, and N. Cristianini, "Support vector machines," *Wiley Interdisciplinary Reviews: Computational Statistics,* vol. 1, pp. 283-289, 2009.

[46] D. Meyer and F. T. Wien, "Support vector machines," *R News,* vol. 1, pp. 23-26, 2001.

[47] N. Guenther and M. Schonlau, "Support vector machines," *Stata Journal,* vol. 16, pp. 917-937, 2016.

[48] J. Maroco, D. Silva, A. Rodrigues, M. Guerreiro, I. Santana, and A. de Mendonça, "Data mining methods in the prediction of Dementia: A real-data comparison of the accuracy, sensitivity and specificity of linear discriminant analysis, logistic regression, neural networks, support vector machines, classification trees and random forests," *BMC Research Notes,* vol. 4, p. 299, August 17 2011.

[49] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery,* vol. 2, pp. 121-167, June 01 1998.

[50] B. Atal and L. Rabiner, "A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 24, pp. 201-212, 1976.

[51] G. Guodong and S. Z. Li, "Content-based audio classification and retrieval by support vector machines," *IEEE Transactions on Neural Networks,* vol. 14, pp. 209-215, 2003.

[52] M. Baillie and J. M. Jose, "An Audio-Based Sports Video Segmentation and Event Detection Algorithm," in *Conference on Computer Vision and Pattern Recognition Workshop*, 2004, pp. 110-110.

[53] L. Deng and D. O'Shaughnessy, *Speech processing: a dynamic and optimization-oriented approach*: CRC Press, 2003.

[54] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-5, pp. 179-190, 1983.

[55] K. F. Lee and H. W. Hon, "Large-vocabulary speaker-independent continuous speech recognition using HMM," in *International Conference on Acoustics, Speech, and Signal Processing*, New York, USA, 1988, pp. 123-126.

[56] S. Furui, "50 years of progress in speech and speaker recognition research," *ECTI Transactions on Computer and Information Technology,* vol. 1, pp. 64-74, 2005.

[57] D. Ververidis and C. Kotropoulos, "Emotional speech recognition: Resources, features, and methods," *Speech communication,* vol. 48, pp. 1162-1181, 2006.

[58] M. Anusuya and S. K. Katti, "Speech Recognition by Machine: A Review," *International Journal of Computer Science and Information Security,* vol. 6, pp. 181-205, 2009.

[59] J. F. Hemdal and G. W. Hughes, "A feature based computer recognition program for the modeling of vowel perception," in *Proc. Symp Models for the Perception ofSpeech and Visual Form*, Cambridge, Mass, 1967, pp. 440-452.

[60] D. T. Tran, "Fuzzy Approaches to Speech and Speaker Recognition," PhD, University of Canberra, 2000.

[61] S. Furui, "An Overview of Speaker Recognition Technology," in *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, Martigny, Switzerland, 1994, pp. 1-10.

[62] A. E. Rosenberg, "Automatic speaker verification: A review," *Proceedings of the IEEE,* vol. 64, pp. 475-487, 1976.

[63] J. P. Campbell, "Speaker recognition: a tutorial," *Proceedings of the IEEE,* vol. 85, pp. 1437-1462, 1997.

[64] S. Pruzansky, "Pattern-Matching Procedure for Automatic Talker Recognition," *The Journal of the Acoustical Society of America,* vol. 35, pp. 354-358, 1963.

[65] B. Atal, "Automatic recognition of speakers from their voices," *Proceedings of the IEEE,* vol. 64, pp. 460-475, 1976.

[66] P. Mermelstein, "Determination of the Vocal-Tract Shape from Measured Formant Frequencies," *The Journal of the Acoustical Society of America,* vol. 41, pp. 1283-1294, 1967.

[67] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 29, pp. 254-272, 1981.

[68] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *IEEE Odyssey: The Speaker and Language Recognition Workshop*, Crete, Greece, 2001, pp. 213-218.

[69] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing,* vol. 2, pp. 578-589, 1994.

[70] S. V. Chougule and M. S. Chavan, "Robust Spectral Features for Automatic Speaker Recognition in Mismatch Condition," *Procedia Computer Science,* vol. 58, pp. 272-279, 2015/01/01/ 2015.

[71] J. Ramirez, J. M. Górriz, and J. C. Segura, "Voice activity detection. fundamentals and speech recognition system robustness," in *Robust Speech Recognition and Understanding*, M. Grimm and K. Kroscel, Eds., ed. Vienna, Austria: I-Tech Education and Publishing, 2007, pp. 1-22.

[72] F. G. Germain, D. L. Sun, and G. J. Mysore, "Speaker and noise independent voice activity detection," in *Proc. Interspeech*, 2013, pp. 732-736.

[73] L. M. Arslan and J. H. L. Hansen, "Language accent classification in American English," *Speech Communication,* vol. 18, pp. 353-367, 1996/06/01/ 1996.

[74] K. Kumpf and R. W. King, "Automatic accent classification of foreign accented Australian English speech," in *International Conference on Spoken Language Processing*, Philadelphia, USA, 1996, pp. 1740-1743.

[75] K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal, "Speech/music discrimination for multimedia applications," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2000, pp. 2445-2448 vol.4.

[76] L. Wyse and S. Smoliar, "Toward content-based audio indexing and retrieval and a new speaker discrimination technique," *Proc. ICJAI,* vol. 95, 1995.

[77] D. Kimber and L. Wilcox, "Acoustic segmentation for audio browsers," *Computing Science and Statistics,* pp. 295-304, 1997.

[78] J. Ubbens and D. Gerhard, *Information Rate for Fast Time-Domain Instrument Classification*, 2015.

[79] J. Foote, "Content-based retrieval of music and audio," in *Multimedia Storage and Archiving Systems II*, 1997, pp. 138-148.

[80] J. Foote, "An overview of audio information retrieval," *Multimedia Systems,* vol. 7, pp. 2-10, January 01 1999.

[81] A. Yoshitaka and T. Ichikawa, "A survey on content-based retrieval for multimedia databases," *IEEE Transactions on Knowledge and Data Engineering,* vol. 11, pp. 81-93, 1999.

[82] T. Zhang and C. C. J. Kuo, "Content-based classification and retrieval of audio," in *Conference on Advanced Signal Processing Algorithms, Architectures, and Implementations VIII*, San Diego, 1998, pp. 432-443.

[83] T. Zhang and C. C. J. Kuo, "Heuristic approach for generic audio data segmentation and annotation," presented at the Proceedings of the seventh ACM international conference on Multimedia (Part 1), Orlando, Florida, USA, 1999.

[84] S. Z. Li, "Content-based audio classification and retrieval using the nearest feature line method," *IEEE Transactions on Speech and Audio Processing,* vol. 8, pp. 619-625, 2000.

[85] L. Chien-Chang, C. Shi-Huang, T. Trieu-Kien, and C. Yukon, "Audio classification and categorization based on wavelets and support vector Machine," *IEEE Transactions on Speech and Audio Processing,* vol. 13, pp. 644-651, 2005.

[86] Z. Liu, Y. Wang, and T. Chen, "Audio Feature Extraction and Analysis for Scene Segmentation and Classification," *Journal of VLSI signal processing systems for signal, image and video technology,* vol. 20, pp. 61-79, 1998.

[87] Z. Liu, J. Huang, and Y. Wang, "Classification TV programs based on audio information using hidden Markov model," in *IEEE Second Workshop on Multimedia Signal Processing*, Redondo Beach, California, 1998, pp. 27-32.

[88] J. S. Boreczky and L. D. Wilcox, "A hidden Markov model framework for video segmentation using audio and image features," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, Seattle, Washington, 1998, pp. 3741-3744.

[89] T. Zhang and C. C. J. Kuo, "Audio content analysis for online audiovisual data segmentation and classification," *IEEE Transactions on Speech and Audio Processing,* vol. 9, pp. 441-457, 2001.

[90] L. Ying and D. Chitra, "SVM-based audio classification for instructional video analysis," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Quebec, 2004.

[91] X. Changsheng, N. C. Maddage, S. Xi, C. Fang, and T. Qi, "Musical genre classification using support vector machines," in *International Conference on Acoustics, Speech and Signal Processing*, 2003, pp. 429-32.

[92] L. Lu, H. Zhang, and S. Z. Li, "Content-based audio classification and segmentation by using support vector machines," *Multimedia Systems,* vol. 8, pp. 482-492, 2003.

[93] A. Shaikh, N. Mahoto, F. Khuhawar, and M. Memon, "Performance evaluation of classification methods for heart disease dataset," *Sindh University Research Journal-SURJ (Science Series),* vol. 47, 2015.

[94] L. Lu, H. Jiang, and H. Zhang, "A robust audio classification and segmentation method," in *Proceedings of the ninth ACM international conference on Multimedia*, 2001, pp. 203-211.

[95] J. Brownlee, "Classification Accuracy is Not Enough: More Performance Measures You Can Use," 2014. [Online]. Available: http://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/. [Accessed: 1 August 2017].

[96] G. Siantikos, T. Giannakopoulos, and S. Konstantopoulos, "Monitoring Activities of Daily Living Using Audio Analysis and a RaspberryPI: A Use Case on Bathroom Activity Monitoring," in *International Conference on Information and Communication Technologies for Ageing Well and e-Health*, 2016, pp. 20-32.

APPENDIX 1

Source code used in this project is available from GitHub repositories. The links below are to the code repositories for this study.

- *pyAudioAnalysis*
  https://github.com/tyiannak/pyAudioAnalysis

- Algorithm used to detect the start and end trim points
  https://github.com/devangovender/trimpointdetector

Source code and scripts are licensed under the Apache 2.0 license (http://www.apache.org/licenses/LICENSE-2.0.html) except where noted otherwise.