2013

Speech Detection Using Gammatone Features And One-class Support Vector Machine

Douglas Cooper
University of Central Florida

SPEECH DETECTION USING GAMMATONE FEATURES AND ONE-CLASS SUPPORT VECTOR MACHINE

by

DOUGLAS A. COOPER

B.S. University of Central Florida, 2009

A thesis submitted in the partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2013

ABSTRACT

A network gateway is a mechanism which provides protocol translation and/or validation of network traffic using the metadata contained in network packets. For media applications such as Voice-over-IP, the portion of the packets containing speech data cannot be verified and can provide a means of maliciously transporting code or sensitive data undetected. One solution to this problem is through Voice Activity Detection (VAD). Many VAD's rely on time-domain features and simple thresholds for efficient speech detection however this doesn't say much about the signal being passed. More sophisticated methods employ machine learning algorithms, but train on specific noises intended for a target environment. Validating speech under a variety of unknown conditions must be possible; as well as differentiating between speech and non-speech data embedded within the packets. A real-time speech detection method is proposed that relies only on a clean speech model for detection. Through the use of Gammatone filter bank processing, the Cepstrum and several frequency domain features are used to train a One-Class Support Vector Machine which provides a clean-speech model irrespective of environmental noise. A Wiener filter is used to provide improved operation for harsh noise environments. Greater than 90% detection accuracy is achieved for clean speech with approximately 70% accuracy for SNR as low as 5dB.

ACKNOWLEDGMENTS

I would like to thank my wife and family for their loving support and understanding. Many thanks to my advisor, Dr. Wasfy Mikhael for his patience and guidance. The application of this thesis was provided by Mr. David Kotick.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

CN	Comfort Noise
ETSI	European Telecommunications Standards Institute
GTCC	Gammatone Cepstral Coefficients
HOS	Higher Order Statistics
ITU	International Telecommunications Union
LP	Linear Prediction
MFCC	Mel Frequency Cepstral Coefficients
ML	Maximum Likelihood
RBF	Radial Basis Function
SNR	Signal to Noise Ratio
STE	Short Time Energy
SV	Support Vector
SVM	Support Vector Machine
OC-SVM	One Class SVM
VAD	Voice Activity Detection
VoIP	Voice over IP
ZCR	Zero Crossing Rate
ZCRPA	ZCR with Peak Amplitudes

CHAPTER 1 INTRODUCTION

1.1 Motivation

In the past decade, voice communications systems have seen a large shift from traditional telephony based exchanges to internet based methods such as Voice-over-IP (VoIP). The simple phone calls that once were are now integrated into private and public computer networks where voice is exchanged simultaneously with video and data. This large scale shift has given rise to a variety of voice encoding schemes used to reduce bandwidth as well as various data encapsulating protocols in which the information is transported.

To allow for protocols on one communication system to interact with others, network gateways have been developed to provide a "translation" functionality between protocols and encoding schemes as well as provide a decision capability for allowing data to pass between networks. Using metadata (data describing the packet) defined within a network protocol, a decision can be made as to whether or not a particular data packet can pass through the gateway. If the metadata of the packet matches or falls within a specified range, the packet is accepted else it is rejected. For voice communications, this functionality is currently limited to the use of data fields surrounding the voice data, leaving the field containing the sampled speech unchecked.

The inability to ensure that packetized voice data actually contains speech can create a vulnerability as well as provide a means of transporting data for reasons other than intended. If the system was designed to handle only voice communications, it should be carrying voice and nothing else. One example of this would be the ability to pass a computer virus or malware disguised as voice between networks of different security levels. In this case the supposed "voice data" might not be voice at all, but rather binary data that when converted to an analog waveform would be meaningless audible noise. Another potential threat would be steganographic type attacks in which the goal is to embed or "hide" data within video or audio samples as a form of undetectable communications. In this case, the analog audio would seem as nothing more than a normal conversation.

In order to combat these attacks, one of two ways might be considered. One approach would be to employ "reactive" detection methods in which known steganographic techniques are detected followed by an action on the packet. If one were to know attack types ahead of time, "proactive" methods could be employed in which the audio is modified, rendering the hidden messages or data useless while minimizing the impact on audio quality. In either case, if steganographic techniques are improved upon both of these methods may be rendered useless. The advantage of the proactive approach would allow for a greater efficiency as simple techniques can be employed allowing for real time operation, whereas the reactive approach would require all known verification techniques to be implemented. Although specific steganographic techniques are not discussed here, the proposed method could be considered a proactive solution.

In order to allow for gateways to effectively make decisions on the speech packets, a logical approach would be to identify whether or not the packets contain speech. This particular area of interest requires the researcher to investigate methods of speech detection and is the basis of this thesis.

1.2 Introduction

Many methods have been developed in order to provide a solution to the speech detection problem. Although the “application” in this thesis differs from many, the necessity to detect speech on a communications channel has been driven by several major areas of research. Bandwidth reduction techniques have been developed in which speech detection (also called Voice Activity Detection or VAD) serves as a key component by allowing bit-reducing encoding schemes to act on the portions of an audio stream that contain speech while omitting the pauses which are replaced with comfort noise at the destination. Noise reduction and echo cancellation techniques have been enabled through speech detection by allowing noise characteristics and filters to be computed during periods of non-speech since speech can interfere in modeling the channel characteristics. Speech recognition engines have relied on speech detectors to provide a so-called “endpoint detection” in which only periods of speech are considered for processing thereby reducing the load of the recognition system. In all cases, the goal is essentially the same while the method for implementation differs depending on the requirements of the application

For any system that requires the detection of certain characteristics of a signal, whether it be audio, images or sensor data, most can be described in terms of two components: a feature (or set of features) and a detector. Particularly for speech, the complexity of features chosen can be as simple as frame energy to more complex approaches such as cepstral coefficients or wavelets. The detection of patterns can range from simple methods such as fixed or adaptive thresholds to more complex approaches using machine learning. Determining which components will be appropriate for this particular application is the main challenge of this thesis.

One of the major areas of concern when developing a VAD is determining how decisions are affected by varying noise types and levels. The noise that is of concern could be either background noise or embedded data, however for this thesis they may be categorized as one in the same – non-speech. The noises can be stationary or non-stationary with signal-to-noise ratios (SNR) below 0dB considered in literature where accuracy of the VAD is tested under these conditions. In order to combat the effects of noise, developing features that are unaffected by noise have been a major area of interest. Since communications networks cannot foresee all possible environmental characteristics, the ideal detector should have the ability to discriminate between human speech and all other noises.

1.3 Design Requirements

Since the gateway needs to pass packetized audio in real time, it is important that the decision be made in real time. Therefore the desired approach must be able to make decisions at the frame level using not more than 30-40ms of data. Likewise, the algorithm should be computationally efficient so as not to induce too great of a delay in the audio stream since gateways can support many channels simultaneously. With the advent of modern computer technology such as mutli-core processers and offloading to Graphical Processer Units (GPU) the number of computational cycles required is not of great concern. The system should be able to operate in a varying degree of noise levels and discriminate against non-speech sounds, although it is expected that some overlap of speech and non-speech sounds may exist. For an English speaking person, the approach should be independent of gender, dialect and relative talking speed. The following provides a summary of these requirements:

- The system should provide a speech/non-speech decision based on a short duration of time
- The system should be computationally efficient for use in a real-time system
- The system should be robust to noise and other non-speech interference
- The system should be independent of the speaker (i.e. male, female, fast/slow talker)

1.4 Thesis Contributions/Goals

The main goal of this thesis is to develop a method for detecting speech on a communications network that is robust to noise of unknown characteristics. Since the background noise of each speaker location can vary, it is desired that speech be modeled using features that allow for discrimination from non-speech sounds. In order to accomplish this task, utilizing a machine learning approach to achieve accurate modeling and decision boundaries seems a fitting approach. Through exploration of various detection and recognition methods, the Gammatone Cepstral Coefficients (GTCC) was found to provide a noise robust feature set that shows improved noise immunity over traditional features by utilizing human-perception and speech production methods. Since the job of a speech detector is to provide a binary decision, the Support Vector Machine (SVM) has been chosen due to its direct ability to provide a speech/non-speech decision, while also providing excellent machine learning properties and low computational complexity. Since it was found that the standard two-class SVM relies heavily on a descriptive data set characteristic of the noise, a one-class SVM (OC-SVM) was explored which attempts to provide a speech only model.

1.5 Thesis Outline

The structure of this thesis is as follows. Chapter 2 introduces the research performed as part of this thesis by presenting various methods commonly used for speech detection in communications systems. Chapter 3 discusses the design of the system in by providing the technical aspects of each system component as well as discussing potential considerations for integration into an operational environment. Chapter 4 provides a description of the components used to evaluate the proposed VAD; testing and measurement criteria used to determine the performance of the speech detector; and simulation results and discussion. Chapter 5 summarizes the thesis and discusses possibilities for future work.

CHAPTER 2 RESEARCH

2.1 Overview

A number of features have been developed over the years in order to provide a meaningful representation of speech. The most common methods include time domain, frequency domain, speech modeling, statistical and autocorrelation. Additionally, several industry standards have been proposed which utilize a combination of these methods. As stated in the previous chapter, noise is a problem which exists in every environment, so as the requirements for applications become more stringent with regards to noise, the features must become more complex. Additionally, in order to obtain a feature set that is more descriptive of speech, one or more of these features must be used simultaneously. To detect the incoming speech signals, an appropriate detection scheme that uses the chosen feature set for decision must also be considered.

The details in this chapter provides an overview of many of the common features used in speech detection as well as the challenges in detecting these features based on the application described in this thesis. Also, related works are discussed which provide a formulation for the considered approach. The research presented here is merely a highlight of popular methods and is not inclusive of all possible detection/classification methods and variations.

2.2 Speech Features

2.2.1 Time Domain

The simplest and most efficient algorithms that have been used for speech detection are energy and zero crossing rate. Both are time domain algorithms that perform relatively well in high SNR environments, but quickly suffer as background noise levels increase. Additionally, neither method provides a useful means of discriminating against speech and non-speech sounds making them primarily useful for detecting onsets and offsets of non-stationary signals with relatively stationary background characteristics. However, since many speech applications do not expect or care about non-speech noises, these methods can be useful in distinguishing between voiced and unvoiced sounds [1] in a clean speech environment.

The Short Time Energy (STE) of a signal is one of the most prevalent features used in speech detection [2] and is defined as

$$Energy = \sum_{n=1}^{N} x(n)^2 \tag{1}$$

where *x(n)* is the input signal and *N* is the length of the signal being measured. In order to detect a specific energy level, the STE can be compared against a fixed or varying threshold to determine when speech has occurred. A fixed threshold can be used when prior information about the communication channel is known, assuming the background noise is relatively stationary. An adaptive approach can be implemented by calculating the energy in the channel during the first few hundred milliseconds of a transmission which can be fixed as the threshold value for the entire length of the signal or by computing a long-term average during periods of non-speech.

The Zero-Crossing Rate (ZCR) of a signal is determined by the average number of times the signal crosses zero over a given period of time [1], [3] and is defined as

$$ZCR = \sum_{n=1}^{N} |sgn[x(n)] - sgn[x(n-1)]| \tag{2}$$

Its usefulness lies in the fact that it can discriminate against tonal sounds commonly found in voiced speech where the value of the ZCR would be much lower than for unvoiced speech. Since unvoiced speech is similar to that of a stationary noise process, discriminating against unvoiced speech and noise becomes a difficult task at low SNR ranges. As with the energy feature, the ZCR can also be compared against a threshold or range of values for detection purposes.

2.2.2 Frequency Domain

Various frequency domain based features have been developed that take into account the shape and statistical properties of the signal spectrum. These features have been used in studying speech/music discrimination [4] by exploiting the differentiating musical properties of sound and have also been showed to be useful for speech detection tasks [5]. Such measures include centroid, roll-off, and flux [6]; flatness [7]; and band energy ratios [8][9], each of which are defined below.

Spectral centroid measures the center of mass or “brightness” of a sound. Percussive and high frequency sounds push the energy towards the higher bands while speech tends towards the lower bands. The centroid *C* for the i^{th} frame can be computed as

$$C_i = \frac{\sum_{n=1}^{N} M[n] * n}{\sum_{n=1}^{N} M[n]} \tag{3}$$

where *M* is the magnitude of the nth frequency bin and *N* is the total number of bins in the positive frequency range. For voiced speech, the magnitude spectrum will weight heavier on the lower frequency bins and for unvoiced speech the bins will be weighted more equally.

Spectral roll-off is the frequency for which the sum of the lower band is a certain percentage of the total band energy, where the percentage can range from 85 to 95. It represents a “right-skewness” of the spectrum where a higher value indicates a larger spread of energy. The roll-off frequency R_f for the i$^{\text{th}}$ frame can be computed using a cumulative sum such that

$$\sum_{\mathrm{n=1}}^{\mathrm{R_f}} M[n] = P \sum_{\mathrm{n=1}}^{\mathrm{N}} M[n] \tag{4}$$

where P is the percentage as expressed in decimal form. This can equivalently be thought of as the slope of the spectrum. If the spectrum is flat or the majority of the energy is in the upper half of the spectrum, the larger the value.

Spectral flatness, also known as Weiner Entropy, is an indicator of how “random” a signal is, where a high value indicates more randomness and a lower value indicates the presence of tonal qualities. It is mathematically defined as the ratio of the geometric mean to the arithmetic mean of the spectrum and is computed as

$$\text{Flatness} = \frac{\sqrt[\mathrm{N}]{\prod_{\mathrm{n=1}}^{\mathrm{N}} M[n]}}{\frac{1}{N}\sum_{n=1}^{N} M[n]} \tag{5}$$

For white noise the flatness measure approaches one and for pure tones it approaches zero.

Spectral flux is measured as the L1 or L2 norm between the magnitude spectra of consecutive frames and captures the temporal deviations of the spectrum. The flux F for the i$^{\text{th}}$ frame can be computed as

$$F_i = \sum_{\mathrm{n=1}}^{\mathrm{N}} (N_{\mathrm{i}}[n] - N_{i-1}[n-1])^2 \tag{6}$$

where N is the normalized energy between the current and previous frames.

A *band energy ratio* is the ratio of the normalized energy of any specified band to the normalized energy in the total measurement band. Depending on the application, a specific band may or may not be useful, however in [9] a low-band to full-band energy ratio (LFER) metric was introduced which takes into account the fact that more energy is distributed in the lower frequency bands for speech. The authors used a cut-off frequency of 1.5 kHz and defined the ratio to be

$$LFER = \frac{E_l}{E_f} \tag{7}$$

where E_l is the total energy in the low passed version of the signal and E_f is the total energy of the measured spectrum.

2.2.3 Statistical

Several features have been developed using statistical approaches to include Higher Ordered Statistics (HOS) and Maximum Likelihood (ML) estimation. The HOS method estimates the third and fourth order moments of the signal and relies on the assumption that the background noise is approximately stationary [10]. If the background noise is in fact stationary, the higher order moments will be approximately zero for background noise and non-zero for speech signals. When discriminating between speech and non-speech sounds is critical for operation, this approach will fall short since many other non-speech sounds can also be non-stationary in the environment resulting in false detections. Likewise, in higher SNR environments unvoiced sounds may blend with the background noise causing higher miss detection rates, however this may be overcome by combining additional features [9].

An alternative statistical approach [11] uses an ML estimate of the SNR for each bin of the DFT which contributes to a Likelihood Ratio Test (LRT) of two conditions: noise only and speech in the presence of noise. The SNR estimate assumes that each DFT bin is a Gaussian random variable where the noise and speech have different statistics. The result of the LRT is then compared against a predetermined threshold for detecting the presence of speech.

2.2.4 Speech Production and Human Perception Modeling

Some of the most popular methods for speech recognition tasks utilize features that are based on models of the human speech process. The performance of these methods have been further improved by incorporating front ends that mimic the human auditory system in order to improve operation at low SNR. Human speech production is typically represented as a source and filter model (discussed further in Chapter 3) in which a voiced or unvoiced stimulus is convolved by a filter that represents the vocal tract and position of the mouth. For the recognition task, the modeling of the filter is of interest since each position of the mouth can represent a unique sound.

The filter modeling has been performed in two main ways: Linear Prediction (LP) and through Cepstrum computation. The LP approach [12] utilizes an adaptive inverse filtering technique to recover the filter response while the Cepstrum [13] approach converts the convolution of the source and filter to a summation where the filter is then extracted. The LP approach has been a popular method used in speech coding tasks while the cepstrum has been more popular in recognition tasks. In general, the Cepstrum can be thought of as a compression and decorrelation (due to DCT) of the spectrum such that the LP (or any spectral representation) can also be converted to the Cepstrum for further processing. Compression occurs because the number of useful Cepstral coefficients for recognition tasks are in the range of 6 to 20 (energy compaction property) which is typically much less than the number of DFT bins. Decorrelation occurs since the energy contained in the transformed signal has a greater and more equal spread across all useful coefficients which can be a useful property for classification tasks.

Several popular methods have been considered for adding a noise robust component to the speech production model. The common theme among these approaches is the use of a filter bank followed by a non-linear operation applied to the energy contained in each band. The theory behind such an approach stems from the physiological makeup of the human auditory system in which the location and impulse response of hair cells along the basilar membrane act as a bank of band-pass filters which allow for reliable detections in the presence of masking noise.

The Perceptual Linear Prediction (PLP) method [14] is similar to LP with the addition of several pre-processing steps. A set of Bark warped band-pass filters are first applied to the power spectrum where the output of each filter is summed and a cube root operation is then applied to each. The filters in this case are flat on top with exponentially shaped skirts of increasing amplitude with increasing frequency. A more popular approach uses a Mel-warped filter-bank of equal amplitude triangular filters and is typically used in conjunction with the Cepstrum to give the Mel Frequency Cepstrum Coefficients (MFCC). In this case the energy in each band is log compressed. As the physiological responses of the auditory system has been studied extensively, realistic band-pass shapes have been proposed, such as the Gammatone filter [15], in order to provide a more accurate and improved representation of the human auditory system. The shapes of the filters take their response from the Gamma function used in the computation. Application of the Discrete Wavelet Transform (DWT) [16–18] provides a physiological-like response since the cascaded structure of high and low pass filtering with resampling simulates a warped band-passed spectrum.

2.2.5 Autocorrelation

Several autocorrelation measures have been proposed. In [19] the normalized autocorrelation coefficient of unit lag is considered as a feature. In [20] a sub-band autocorrelation approach is presented in which the speech signal is first processed by a bank of band pass filters and then the autocorrelation of each is computed at a lag equal to the inverse of the center frequency of each filter.

2.2.6 Additional Features

Formant tracking has been considered as a speech detection method due to the fact that shapes made by formants are unique to speech signals. The formants of speech signals are the spectral peaks which make up voiced sounds and can take shapes that are straight, convex and concave over small durations of time and for different sounds. In [21] the author used this fact to track local peaks using the DFT of the LPC domain over specified periods of time. In order for a segment of audio to be considered as speech, it must meet certain criteria such as minimum and maximum duration for each shape type, minimum duration for a local formant and maximum difference between peaks of formants. Each of these values is tuned to obtain optimal performance. Although this approach utilizes information that is unique to speech, it requires large durations of time in order to capture this information which may not be reasonable for real-time communications.

The authors in [22] presented a feature called Zero Crossings with Peak Amplitudes (ZCPA). This feature considers computing the upward going crossings at the output of individual filters in a filter bank. The peak amplitude between upward zero crossings are log compressed and then assigned to frequency bins where the assignment to a bin is determined by the inverse of the time period between crossings. The peak amplitudes belonging to the same bin are summed and then the corresponding bins of each filter are summed which results in a singular frequency histogram. The strength of this approach is that it focuses on extracting the energy of dominant frequencies in a sub-band whereas the other sub-band methods, such as MFCC, focus on the energy in the entire sub-band which includes energy from broadband noise. The authors in [23] show that this feature slightly outperforms the perceptual features presented in section 2.2.4, however at the cost of 20 times the computational complexity.

2.2.7 Commercial Implementations

Several standards have been developed to provide a common mechanism for speech detection among communications systems. The International Telecommunications Union (ITU) introduced one of the first major standards in 1996 known as G.729 [24] which describes the implementation of a Voice Activity Detector (VAD) and Comport Noise Generation (CNG) used for Discontinuous Transmission (DTX) of voice.

The G.729 VAD uses the ZCR as well as several features derived from the LP computation which includes Line Spectral Frequencies (LSF), full-band energy and low-band energy. The use of LP based features allows for reduced computational load since the autocorrelation computation which solves for the LP coefficients can be used for both encoding and VAD. The LSF speech feature is an alternative form of the LP coefficients that allows for reduced quantization error in speech coding [25]. The full and low band energies are calculated directly from the autocorrelation coefficients where the full band energy is the same value as the STE feature and the low band energy is typically the energy below 1 kHz.

The implementation of this VAD uses an adaptive detection approach where initial parameters are computed at the start of a transmission and then updated during periods of non-speech. Rather than use the features directly, “difference features” are computed for each frame using the instantaneous values and the running average of each. These difference features are then compared against a series of predetermined boundary conditions.

The ETSI published standards around the same time as the ITU that would be used for the purpose of voice compression in GSM cellular networks. The features and detection used by the ETSI are more robust than the G.729 since they do not rely on time domain related features and LP parameters which are both shown to have poor noise performance [26]. The ETSI published two options in the same standard where the second option has better performance while sacrificing computational efficiency [27].

The ETSI Option 1 considers several features to include sub-band energy levels (using DWT strategy) and pitch, as well as tone detection for informational tones and complex signal analysis for sounds such as music [28]. The energy levels of nine bands are computed across the speech range of 0 to 4 kHz using a sub-band filtering technique where each successive filtering stage is decimated for more efficient computation. Energy levels of the current frame are compared against a long term estimate of the noise. Pitch detection is computed using an autocorrelation method in which the lag with the maximum peak is determined. A pitch flag is set if consecutive frames contain similar pitch. The lag at which the maximum occurs is then used for tone detection in which the energy of the signal at the determined lag is also considered. For complex signal analysis, the maximum of the autocorrelation of the high pass filtered speech is determined and is smoothed using a first order filter. Tone detection and complex signal detection flags are set if they exceed predetermined thresholds. The purpose of detecting these different types of signals is so that they are not replaced by comfort noise which is considered to be annoying.

The ETSI Option 2 utilizes estimates of the background noise and channel energy to compute instantaneous and long-term SNR's. The instantaneous SNR is then quantized and converted into a voice metric value which is compared against a threshold that is determined by the long term SNR. A power spectral estimate is also computed to determine when a noise estimate update is appropriate.

The ETSI also published a feature extraction standard for use in distributed speech recognition [29], in which speech features are computed on a client computer that are then compressed and transmitted across a network for processing on a server. The approach utilizes three measurements of the DFT of the output of a two-stage Wiener filter as input to the VAD. The first measurement considers energy values across the whole spectrum while the second considers energy values in a sub-region of the spectrum where the fundamental pitch is likely. The third measurement considers the variance of the energy in the lower half of the spectrum to account for "acceleration" associated with speech onset. The output of each measurement is combined and the decision is stored into a buffer which acts as a look ahead for decision smoothing.

2.3 Detection

The most basic form of detection is that which employs a singular threshold value based on the known noise properties of the channel. Although efficient, this method does not tell us much about the signal being detected in the sense that no unique description has been achieved. Employing multiple features, each with their own threshold, can help to narrow the possibilities of signals that will be accepted, yet obtaining these threshold values is still a difficult task when environmental characteristics vary and there are a large number of features. In order to remove the need for the standard hard threshold as well as provide a more descriptive representation of the signal, a machine learning approach is considered in order to “model” the desired signal such that detections that do not fit the model will be rejected.

There are an overwhelming number of machine learning methods that can be used to employ detection. These methods can be supervised, unsupervised, semi-supervised and so on such that sampled data can be used to train a classifier which delineates the features as belonging to one of two or more classes. For speech detection these classes can be speech or non-speech; or even voiced, unvoiced and silence. Since the application in this thesis deals with determining whether or not speech is present, the number of possible machine learning approaches can be narrowed to those that deal with binary decisions. A popular method for providing binary decisions is the Support Vector Machine (SVM) which was introduced in [30]. The SVM is a computationally efficient classifier that can be easily optimized for complex descriptions and provides good generalization for datasets that are either under or oversampled. For this reason the SVM was explored for use in the speech detection system.

2.3.1 Support Vector Machine

The SVM is a two-class classifier that is trained using pre-labeled data from two separate classes and determines an optimal hyper-plane for separation. Since the data must be labeled prior to training, it is known as a supervised method for classification. For speech detection, the SVM can be trained on speech and noise such that a binary decision can be provided for both clean and noisy speech. Several works were identified which utilizes various features in conjunction with an SVM for speech and non speech classification.

In [31] the author proposes using MFCC, ZCR, energy and several spectral features to distinguish between speech and non-speech over a one second window. The mean and standard deviation of these features are computed over smaller windows within this time frame which is used as the final feature vector for decision. In [32] the author considers the use of Long Term Spectral Divergence which is decomposed into sub-band SNR approximations as features. In [33] the same author proposes using a Wiener filter in conjunction with sub-band power which achieves comparable results to [32]. The author in [34] considers using the raw speech samples as features for endpoint detection of clean speech when trained on clean speech, then provides results when trained and tested with additive noise. In [35] the author considers wavelet de-noising in conjunction with sub-band power, ZCR and pitch frequency. The same author also considers the AMR-WB and NB commercial features [36] and then considers modifying the AMR sub-band computation using wavelets [37]. In [38] the MFCC are again considered with the addition of the difference and double-difference coefficients. In [39] the author considers using the *a-priori* SNR, *a-posteriori* SNR and predicted SNR as features (similar to the Sohn LRT features) and in [40] considers the SNR outputs presented by Sohn in [11].

For all previous works described, agreeable detection performance is achieved for clean and low SNR environments. However, it is important to notice that when measurement of noise performance is considered, the training set was inclusive of the noise being tested. For the application presented here, the noise statistics can be produced from a variety of potentially undetermined environmental conditions. Likewise, embedded non-speech data (i.e. files, images etc.) within the speech frames can also be considered noise that can be produced in an unknown number of ways. It is obvious from these facts that training with all possible noise is not a feasible task, although one could obtain a rich dataset of noise and assume it as being “good enough”. With this in mind a different approach is taken where knowledge of clean speech only conditions are used to train a classifier such that non-speech is rejected and noise robustness is obtained through the use of noise reduction and selection of features.

2.3.2 One-Class Support Vector Machine

The downside of the two-class SVM approach to VAD is the need to have training samples which are representative of the various types of noise in many operating environments. Instead of trying to understand all possible noises, one can ask: Is it possible to detect speech in many conditions by only knowing what clean speech looks like? Leveraging the various strengths of the standard SVM, the One-Class SVM (OC-SVM) provides a promising means of such a task. One approach to the one-class classification problem using an SVM was presented in [41] where an optimal hyper-sphere is used to describe the data. Another approach was presented in [42] which finds a hyper plane that maximizes the distance of the data from the origin. In either case, the main idea behind this theory is that only a description of the desired signal is available such that when the signal is detected it is accepted while all others are considered outliers and are rejected.

One speech based application using an OC-SVM is speaker verification. In [43] the authors consider both speaker verification and identification where each speaker is trained on their own classifier using LP coefficients. Training/testing utterances are quantized using K-Means and classification is performed in a one-against-one strategy where each speaker has a classifier assigned for each other speaker. Training on both negative and positive examples is presented where the positive examples are that of the speaker and the negative is another speaker. In [44] the authors consider a speaker recognition application using LP and MFCC coefficients in a one-against-many strategy without considering other speakers for training. Instead they use the other speakers to tune each classifier so as to reduce the overall classification error. They compared their performance in relation to a Gaussian Mixture Model (GMM) and then in [45] they combined their approach with a GMM to further improve performance.

Another speech application considered is sound classification. In [46] the authors use a K-means like procedure to iteratively assign and train several OC-SVM's over a training set for speech/music discrimination. The features considered are the mean and standard deviation of MFCC, pitch, brightness and spectrum power captured over many frames for a large duration of time. The authors do not discuss effect of duration or the number of clusters used in classification. In [47] the authors classify short duration sounds where each sound is assigned a different classifier. The MFCC's are captured in several frames over the sound duration and a distance measure is formulated to efficiently assign the sound to the appropriate class. The same authors presented an identical procedure in [48] with the addition of several features to include ZCR, energy, spectral centroid and roll-off, MFCC, LPCC, PLP and wavelets.

Speaker segmentation for audio diarization was considered in which the changes in active speakers is detected in an audio stream. In [49] the authors approach the problem by considering a sliding window that is split into two halves. At each increment, one half is used to train an OC-SVM and the other half is used to test. If the prediction is true then the entire window contains the same speaker, otherwise a transition is assumed at the halfway point of the window. Overlapping frames comprise the large window and the features considered are MFCC and DWT. A similar diarization application is single/multi-speaker discrimination in an audio stream [50]. The authors consider the mean and log variance of spectral flux, spectral centroid, ZCR, and Cepstrum as well as the mean of kurtosis. The audio is analyzed over one second durations where both frames and durations are overlapping. The performance of the features was considered separately as well as together, and it was found that the combination gave the best result.

2.4 Summary

From the survey of OC-SVM applications, it is obvious that computing the statistics of the features over a long duration is desirable in addition to the use of multiple classifiers. This is likely due to a limitation of a single OC-SVM to model complex datasets such as speech. By using the statistics over a longer period of time the complexity of the model is reduced. Since operating in real time is a desirable requirement, long duration statistics is not feasible, and therefore more instantaneous decisions must be made.

Besides using noise performance as a criterion for feature selection, maintaining scale invariance against differing speaker amplitudes is also desirable. It can be shown that for a given utterance at different amplitudes, the Cepstrum coefficients (excluding 0th) are scale invariant. Likewise, any statistical representation, such as those presented in the frequency domain, is also scale invariant. Therefore both Cepstrum and frequency features were considered. For this thesis the Gammatone approach was selected due to its improved noise performance over the Mel-frequency approach.

CHAPTER 3 DESIGN & IMPLEMENTATION

3.1 Overview

This chapter is dedicated to the design details of the proposed speech detector. The technical aspects of each component are presented along with a discussion of each. A high-level diagram of the overall system design can be seen in Figure 1.

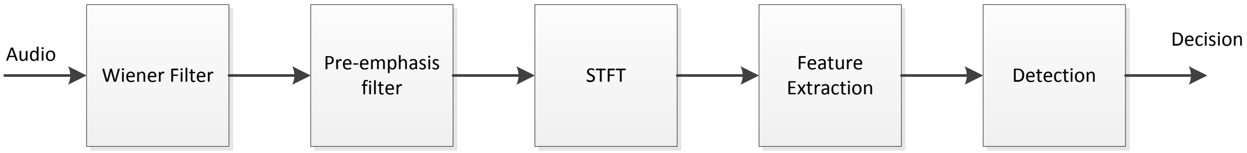


Figure 1: Proposed System Design

The figure presented is a very common scheme when performing signal detection tasks. In fact, the pre-emphasis and STFT approaches were taken straight from the literature since they are commonly used steps in recognition. Additionally, the features chosen have also been studied extensively in the literature for recognition tasks and it is their state-of-the-art performance which is the driving force behind their choice in this application. Noise reduction is also a common step in many recognition tasks and is an essential component of the system.

The main contribution of this thesis lies in the detection step where the OC-SVM is employed with other components to provide a speech detector that is robust to varying speakers and noise conditions. The method of detection provides a generic way to train a detector that is a representation of only speech such that all other non-speech noises are rejected. The selected features give a demonstration of this aspect, while not necessarily providing the best possible description of speech. In general, the noise reduction and feature components can be replaced to potentially improve upon the research proposed here.

3.2 Wiener Filter

The Weiner filter provides a noise suppression mechanism in which an estimate of the background noise is obtained and used to reduce the noise across the audio stream. Although the addition of a noise suppression step is undesirable, its inclusion serves multiple purposes. Firstly, any process which is sensitive to noise can most definitely benefit from noise suppression. From a perceptual point of view the human is able to naturally reduce background noise which improves the signal to noise ratio allowing for recognition of low-level signals. The binaural nature of human hearing (two ears) exhibits directional properties which is effective at reducing the level of interfering sounds. In the case of a monaural system, this cannot be achieved therefore simpler functionality must be employed. Second, the decrease and/or removal of noise reduces potential overlapping in the classification stage where noise patterns can intersect with portions of the speech model. This may occur due to under fitting in the speech model where the regression is fairly loose or through description of the signal where a poor or ineffective feature set has been chosen. Finally, from a steganalysis perspective the noise reduction can increase the bit error rate of potential watermarking or embedded data.

For annotation purposes, a lower case variable indicates a scalar or column vector (bold), while a bold capital letter indicates a matrix. Given a noisy speech signal *y*(n),

$$y(n) = x(n) + n(n) \tag{8}$$

where *x*(n) is the clean speech signal, *n*(n) is an additive noise and *n* it discrete time, we would like to recover *x*(n). This can be accomplished by creating a filter **g** such that the estimate of *x*(n) is

$$\widehat{x(n)} = \sum_{\mathrm{n=1}}^{\mathrm{N}} g_n y(n) = \mathbf{g^T y} \tag{9}$$

where T indicates the transpose. Using a Minimum Mean Square Estimate (MMSE) approach as in [51], it can be shown that the optimal filter is

$$\boldsymbol{g} = \mathbf{R_{yy}}^{-1}\mathbf{r_{xy}} \tag{10}$$

where $\mathbf{R_{yy}}$ is the autocorrelation of the noisy signal $\mathbf{y}$ and $\mathbf{r_{xy}}$ is the cross-correlation of the noisy signal and desired signal x(n). For signal processing applications, the autocorrelation and cross correlation can be approximated by a time-average estimate such that

$$\boldsymbol{g} = (\boldsymbol{Y^T Y})^{-\mathbf{1}}\boldsymbol{Yx} \tag{11}$$

where **Y** is a block form of shifted samples and **x** is a vector of current and previous samples. This realization provides an approximation such that

$$\lim_{n\to\infty} E[(\boldsymbol{Y^T Y})^{-\mathbf{1}}\boldsymbol{Yx}] = \mathbf{R_{yy}}^{-1}\mathbf{r_{xy}} \tag{12}$$

In practice, the autocorrelation must be computed using the clean signal *x*(n) which can be calculated by subtracting an estimate of the noise from the input signal. The noise signal can be obtained from the first few hundred milliseconds of transmission and updated during periods of non-speech.

3.3 Pre-Emphasis Filter

A pre-emphasis filter is a high pass filter which provides a gain in the upper half of the pass band and reduces low frequency interference such as AC hum caused by ground loops in a communications system. It is commonly used in speech recognition systems to remove the mean and boost the high frequency components of speech signals. For computations that require an auto or cross correlation computation, the DC component degrades performance while also creating a bias for machine learning algorithms. The frequency response of voiced signals tends to roll-off with increasing frequency which makes the information there les prominent.

For implementation, a 1st order FIR filter is typically employed with a difference equation in the form

$$y(n) = x(n) - \alpha * x(n-1) \tag{13}$$

where α is set to .97. The response of this filter can be seen in Figure 2 below.

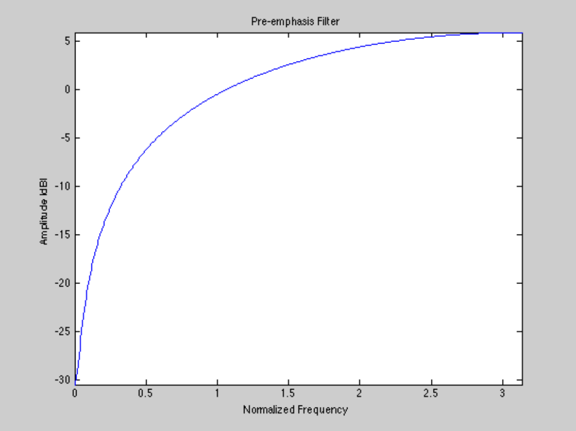


Figure 2: Pre-emphasis Filter Frequency Response

3.4 Short-Time Fourier Transform (STFT)

The STFT is a popular front-end processing stage for speech recognition and analysis problems the main steps for which can be seen in Figure 3. It consists of dividing a large or continuous amount of data into frames of finite duration, followed by the application of a time-domain window and then a Fast Fourier Transform (FFT). The choice of size for these frames is determined by the required time/frequency resolution of the signal being measured. If a small window is chosen (relative to the signal frequencies being measured) good time resolution is achieved at the cost of poor frequency resolution. In order to obtain accurate frequency measurements, the window must be long enough to measure the periodicity in the signal; hence small windows will achieve good resolution for higher frequencies and poor resolution for low frequencies. If lower frequency resolution is desired, the length of the window can be increased at the cost of poor time resolution. The time resolution has now become worse because for high frequency components many changes could have occurred over the course of the window duration and have now been combined into a single snapshot for that period of time.

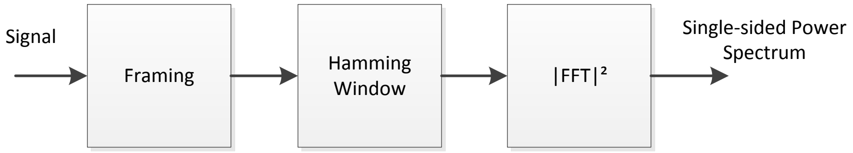


Figure 3: Short Time Fourier Transform (STFT)

For the processing of speech signals, a window size of 20 to 30 ms is typically chosen. This range of measurement achieves two things: the resolution needed to capture lower frequencies in the speech band (typically 0 to 4 kHz) and preservation of the short-sense stationarity of signal. The stationary property is important since many speech processing algorithms use this assumption to simplify computations where the statistics are assumed to be stationary. One example of an algorithm that relies on this property is Linear Predictive Coding which was mentioned in the previous chapter.

When frame based frequency domain processing is used, applying a time-domain window function to the signal is necessary in order to reduce the effects of the discontinuities at the edges of the window. The use of a rectangular window (no window) causes the spectrum of the signal to be convolved by a SINC function in the frequency domain which adds “ringing” into the signal and creates products which do not actually exist in the signal. To reduce this effect, a window which has a maximum at the center of the frame and falls to zero at the edges is applied. For speech applications a Hamming Window is a popular choice since it provides the necessary attenuation at the edges of the frame which lowers the effects of ringing. The complex spectrum X(n) can then be computed as

$$X(k) = FFT(x(n)w(n)) \tag{14}$$

where x(n) is the frames time domain signal and w(n) is the chosen window function equal to the size of the frame.

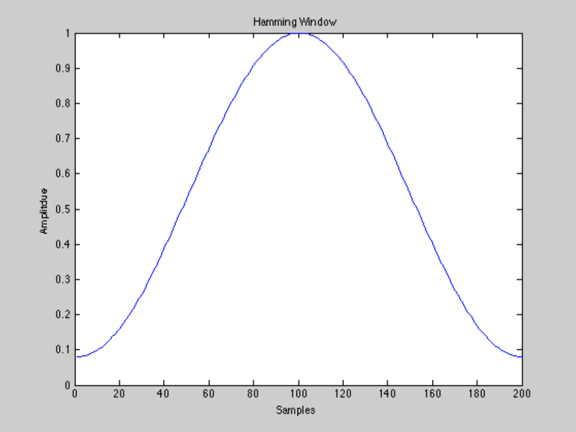


Figure 4: Hamming Window

Since the features used in this design do not incorporate phase, the amplitude or power spectrum can then be computed. The power spectrum has been chosen since its computation is commonly incorporated as part of the MFCC features which is the basis for implementing the GTCC features and is computed as

$$|X(k)|^2 = X(k)X(k)^* \tag{15}$$

where * denotes the complex conjugate.

3.5 Feature Extraction

Based on the survey of feature extraction techniques conducted in Chapter 2, the filter bank approach seemed to be the best suited. This was primarily due to the performance seen by the MFCC features in speech recognition and detections applications. The advantage of using a filter bank is seen in its noise masking properties that are related to the physiological behavior of hearing. By band pass filtering individual portions of the spectrum and then using the energy in each, added distortion and other potential masking signals are smoothed and distributed among several filters which reduces their effect.

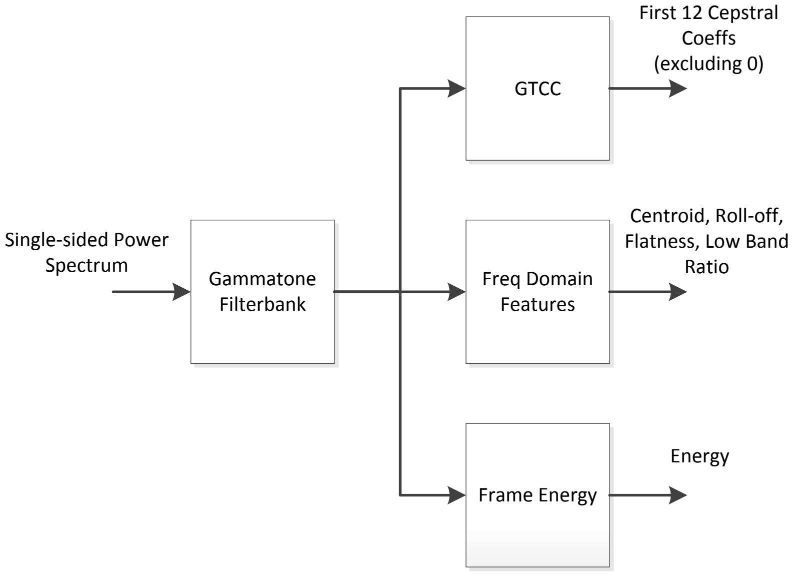

First 12 Cepstral Coeffs (excluding 0)
GTCC
Centroid, Roll-off, Flatness, Low Band Ratio
Single-sided Power Spectrum
Gammatone Filterbank
Freq Domain Features
Energy
Frame Energy

Figure 5: Feature Extraction

Recent literature showed that the Gammatone Cepstral Coefficients (GTCC) can provide equal or better performance than MFCC due to the improved characteristics of the filter responses [23], [52–54]. In order to increase the dimensionality of the feature vector while maintaining scale invariance, several frequency based features were also utilized. Rather than compute their values directly from the DFT spectra, the output of the Gammatone filter bank was used instead to approximate these values which reduced the computational complexity. Finally, a log energy feature was used (also directly computed from the filter bank output) that provided a simple mechanism to detect silence. A high level diagram of these functions can be seen in Figure 5.

3.5.1 Gammatone Filter bank

3.5.1.1 Theory

The Gammatone filter bank is a physiologically inspired representation of the auditory system front end [15]. The basilar membrane found in the cochlea is the mechanism which provides the conversion of acoustical sound into electrical impulses. As the sound wave moves along the basilar membrane, frequency selective stimulations occur as ripples with various lengths and amplitudes. Through biological investigation, it was found that these ripples can be decomposed into a multi-channel representation with impulse response envelopes similar to the gamma function. The time domain Gammatone filter defines the impulse response at different positions along the basilar membrane

$$g(t) = at^{n-1}e^{-2\pi Bt}\cos\left(2\pi f_c t + \phi\right) \tag{16}$$

where a represents the gain, n is the filter order, B is the filter bandwidth, f_c is the center frequency and ϕ is the phase.

The Equivalent Rectangular Bandwidth (ERB) is a psychoacoustic measure of the width of the auditory filter at each point along the cochlea [55] and represents the bandwidth B in equation (16). The ERB for a given filter with center frequency f_c is defined by

$$ERB = \left[\left(\frac{f_c}{EarQ}\right)^n + minBW^n\right]^{1/n} \tag{17}$$

where *EarQ* is an asymptotic filter quality at large frequencies and *minBW* is minimum bandwidth at low frequencies. Filter quality is a measure of its center frequency divided by the bandwidth. Several researchers have suggested different values for these parameters; however the most widely accepted is provided by [56] in which *EarQ* is 1000/(24.7*4.37), *minBW* is 24.7 and *n* is 1. The choice of these parameters is mainly due to the higher quality factor achieved at lower frequencies.

The filter bank design first begins by either defining the number of filters *N* or an overlap factor *v*, along with the low and high frequencies f_l and f_h which describe the analysis bandwidth. In the experiment portion of this thesis, we are interested in understanding the number of filters which gives the best detection performance. Therefore we begin our derivation assuming the number of filters is an input which results in a derived overlap factor. Given the number of desired filters, the overlap factor is defined as

$$v = \frac{EarQ}{N} ln\left(\frac{f_h + EarQ \cdot minBW}{f_l + EarQ \cdot minBW}\right) \tag{18}$$

which specifies the amount of overlap (fraction of the ERB) needed to achieve a certain number of filters over the specified band [56]. A step factor close to zero indicates almost complete overlap while a step factor close to one indicates almost no overlap within the 3dB bandwidth. Finally, the center frequencies of each filter can be determined by

$$\begin{gathered} f_c = -(EarQ \cdot minBW) + (f_h + EarQ \cdot minBW)e^{-\frac{iv}{EarQ}} \\ 1 \leq i \leq N \end{gathered} \tag{19}$$

3.5.1.2 Implementation

Efficient implementation of the filter bank has been studied extensively. In [57], Schofield showed that the 4th order Gammatone filter gave a very close fit to the human auditory filter shape. As a result this is typically the order used for most implementations in order to maintain accuracy while maximizing computational efficiency. In [58], Holdsworth et al. showed that an nth order Gammatone filter can be represented by a cascade of n 1st order recursive filters which is derived using a pole mapping technique.

In [55], Cooke argued that the pole mapping technique can lead to a poor representation of linear system response in terms of magnitude, impulse and/or phase. An extensive study of different digital filter derivations was conducted and it was shown that the impulse invariance method provided the best preservation of magnitude and phase characteristics in comparison to pole mapping and bilinear methods. The 4th order implementation presented, similar to Holdsworth et al., uses the concept of baseband filtering in which the spectrum is shifted to DC for each filter, low pass filtered and then shifted back.

In [55], Slaney argued that Cooke's method requires the use of a complex exponential to shift the signal which is equivalent to the computational complexity of an 8th order filter due to the need for both real and complex multiplication. Using Laplace analysis, Slaney showed that the 4th order Gammatone filter could be represented as an 8th order filter which was comprised of a cascade of four 2nd order biquad recursive filters. Additionally, he showed that an all-pole version (removal of the zeros) could be constructed with only minor gain reduction in the lower frequency range at the improvement of half the computational complexity.

In [59], [60] Hohmann formulated a complex version of the Gammatone impulse response using the impulse invariance method and provided analysis for its all-pole version. By adding the complex component, which represented the Hilbert transform of the impulse response, Hohmann showed that reconstruction could be performed which has applicability in areas such as hearing aids.

In [61], Ma introduced an improved computational version of the Cooke method in which the complex exponential used for frequency shifting was reformulated in terms of the previous sample rather than the current. In doing so, only one complex exponential calculation is needed for the first sample, and the remaining exponential calculations can be computed by a simple multiplication. Doing so provided in efficiency gain of 4 over Cooke's original version.

The implementations previously mentioned are typically performed in the time-domain in order to extract the envelope of the impulse response for further processing. However, we are simply interested in the energy contained in each filter band which will be used to formulate the feature vector. To aid in improving computational efficiency, [62] provided a frequency domain formulation of the Slaney implementation in which the magnitude frequency response of each filter is contained in a matrix allowing for simple and efficient computation of the filter energies given the DFT of the input signal. Given a matrix **G** which contains the rows of DFT bins for each filter and a column vector **x** which contains the DFT bins of the input signal, the vector of energies **e** can be computed as

$$\boldsymbol{e} = \boldsymbol{Gx} \tag{20}$$

or represented in its summation formulation as

$$E(l) = \sum_{k=1}^{N} |X(k)|^2 \cdot G_l(k) \quad for\ l = 1{:}L \tag{21}$$

where N is the number of DFT bins and L is the number of filters.

3.5.2 Cepstral Coefficients

A popular method for speech analysis is through the representation of speech production as a “source/filter” convolution model. The source is a stimulus which produces sounds that can be classified as either voiced or unvoiced. Voiced sounds are represented as a train of pulses which mimic the glottal excitations in the vocal tract while unvoiced sounds are represented as a white noise process. Air from the lungs is pushed through the vocal tract and mouth which represents the filter. Various lengths and shapes of the vocal tract and mouth determine the specific sounds that we make.

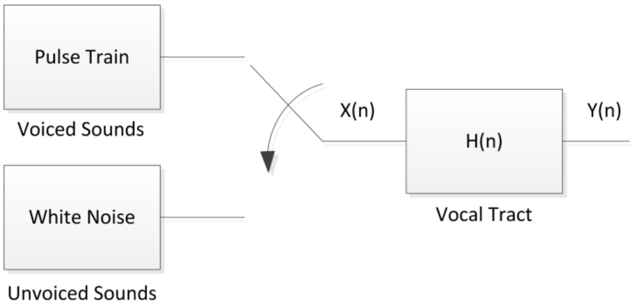


Figure 6: Speech Production Model

In speech recognition tasks, it is common to isolate the filter since it can be used to represent the fundamental sounds that makeup words. This is typically performed using LPC analysis or through cepstrum computation, the latter of which is considered here. Computing the cepstrum can be seen as a deconvolution process in which the convolution is converted to a summation such that the filter information can be easily extracted. Given the process seen in Figure 6, a sound y is represented by the source convolved with the filter

$$y(n) = x(n) * h(n) \tag{22}$$

where * denotes the convolution. If we take the log of the DFT of *y(n)*, the convolution becomes

$$\log\big(Y(k)\big) = \log\big(X(k)\big) + \log\left(H(k)\right) \tag{23}$$

where a capital letter denotes the DFT of the associated time signal. Finally, the inverse DFT is performed to compute the complex cepstrum

$$C = IDFT(\log\big(y(k)\big)) \tag{24}$$

For speech recognition tasks, maintaining phase is not a typical practice and therefore the inverse DFT is simplified by taking the DCT of the signal which handles on the real portion of the DFT.

The output cepstrum can now be viewed as a summation of the source and filter where the lower portion represents the filter and the upper portion the source. For the purpose of recognition, the first 10 to 13 coefficients have been shown to provide the most significant contribution to performance and can be filtered by simply applying a binary mask. In practice however this filtering can be performed in the DCT computation by only computing the desired number of coefficients.

3.5.3 Frequency Features

The frequency features considered here are those described in section 2.2.2 with the exception of spectral flux since temporal features are not being considered as a whole due to real-time constraints. The formulation for each is the same as those described in equations (3), (4), (5) and (7) with the exception that the DFT bins are replaced by the filter bank energies.

3.5.4 Energy Feature

The energy feature considered is defined as the log of the sum of filter energy outputs and is expressed as

$$Energy = log\left(\sum_{l=1}^{L} E(l)\right) \tag{25}$$

where *E(l)* is the energy of the l^{th} filter output.

3.6 Detection

Using the features extracted from the speech signal, several components are used for detection. First, the OC-SVM provides the main pattern recognition mechanism that determines the presence of speech. One interesting property was noticed during informal testing of clean and noisy speech. For clean speech both silence and speech were accepted indicating that it operated at what would be considered a high true and false positive rate. For noisy speech it responded like a true VAD in the sense that silence with noise was rejected while speech was accepted. This response was indicative of the outlier detection nature of the OC-SVM. In terms of traditional speech detectors this property is not desirable, yet for the application presented in this thesis it is the ideal operation.

In order to add to the design such that the response was that of a traditional VAD, a simple fixed threshold energy detector was employed. The threshold was determined during the training process in which the silence frames not used in training were instead used to compute the average energy of the pauses and silence given clean speech signal. When the outputs of both detectors are compared with a logical AND function, the result is that of a normal VAD. Finally, the weighted filter was added to combine adjacent overlapping decisions. The high level diagram can be seen in Figure 7.

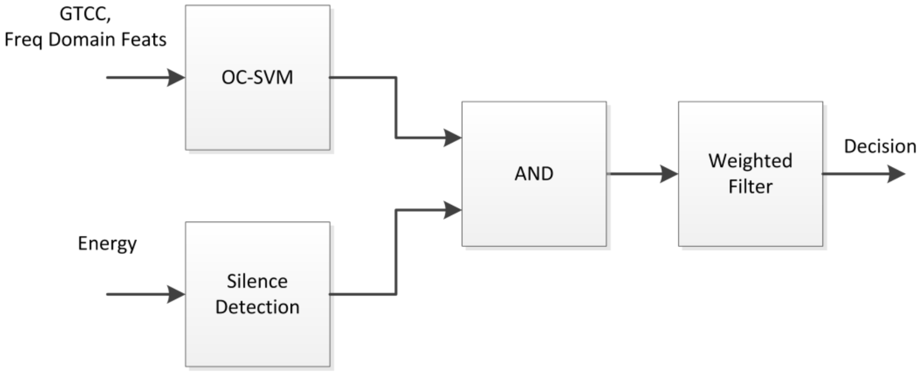


Figure 7: Detection Computation

3.6.1 Support Vector Machines

The Support Vector Machine (SVM) is a popular learning tool for binary classification which operates by producing an optimal hyper-plane between two datasets of different classes. The term "support vector" is derived from the fact that data points selected during training form the plane that best describes the data. When the data cannot be linearly separated, a "kernel" is first applied to the input which provides a transformation to a higher dimensional space where the data becomes linearly separable. The optimal selection of the kernel is based on the particular data set being considered and in the case of speech classification the Radial Basis Function (RBF) kernel has been shown to provide good performance (Section 2.3). The plane formed in the higher dimensional space now corresponds to a non-linear boundary in the original input space.

A one-class SVM (OC-SVM) provides a similar function, but instead uses the support vectors to describe only the data provided for a single known class. Several formulations for the decision boundary have been presented. In [41] Tax et al. uses the concept of hyper-spheres where the maximal sphere around the data is selected. In [42] Scholkopf et al. formulates an optimal hyper-plane which maximizes the distance between the data and the origin. In the case of non-linear separation using an RBF kernel, the result of each is nearly identical resulting in a non-linear boundary around the data. For the purposes of this thesis, only the RBF kernel is considered and the formulation provided by Tax is presented.

3.6.1.1 One-Class SVM

In order to use the two-class SVM, data for both the target class and the distribution of data outside of the class must be available. One way to handle this is to take samples of data for which the target class does not exist. Alternatively, the One-class SVM (OC-SVM) can overcome this challenge by requiring only the data for the target class. In the case of speech detection, the goal is to model the speech using only clean speech samples such that noise that does not have the same characteristics as speech is rejected. Given a training set $\{\boldsymbol{x}_i\}$, $i = 1,\ldots$N, where each represents a feature vector, we define an error function for a sphere of radius R and center $\boldsymbol{a}$ to be

$$F(R, \boldsymbol{a}, \xi_i) = R^2 + C \sum_{i}^{N} \xi_i \tag{26}$$

where C is the tradeoff between the simplicity of the model and error, and ξ is a slack variable. The role of the cost parameter will be discussed in Chapter 4. The goal is to minimize the radius of the sphere subject to the constraint

$$\|\boldsymbol{x}_i - \boldsymbol{a}\| \leq R^2 + \xi_i \tag{27}$$

The above problem is reformulated as a Lagrangian which incorporates the constraints on the sphere

$$L(R,\boldsymbol{a},\alpha_i,\gamma_i\xi_i) = R^2 + C\sum_i^N \xi_i - \sum_i^N \alpha_i\{R^2 + \xi_i - (\boldsymbol{x}_i^2 - 2\boldsymbol{a}\boldsymbol{x}_i + \boldsymbol{a}^2)\} - \sum_i^N \gamma_i\xi_i \qquad (28)$$

where α and γ are the Lagrange multipliers. We minimize L w.r.t. R, $\boldsymbol{a}$, and ξ by setting the partial derivatives to zero which gives the following constraints

$$\sum_i^N \alpha_i = 1 \qquad (29)$$

$$\boldsymbol{a} = \sum \alpha_i \boldsymbol{x}_i \qquad (30)$$

$$C - \alpha_i - \gamma_i = 0 \qquad (31)$$

Equation (29) states that the sum of all Lagrange multipliers should equal 1. Equation (30) states that the center of the sphere is a weighted sum of the features vectors and the Lagrange multipliers. Equation (31) is a constraint on the range of values that the Lagrange multipliers can take which is simplified further by removing γ and setting $0 \leq \alpha_i \leq C$. Substituting Equations (29) thru (31) into equation (28) gives

$$L = \sum_i^N \alpha_i(\boldsymbol{x}_i \cdot \boldsymbol{x}_i) - \sum_{i,j}^N \alpha_i\alpha_j(\boldsymbol{x}_i \cdot \boldsymbol{x}_i) \qquad (32)$$

Finally, L is maximized which gives the final set of α_i that is used for classification.

Values of x_i that satisfy the constraint in Equation (27) within the boundary will correspond to $\alpha_i = 0$ and those that are on the boundary will correspond to $\alpha_i > 0$. When evaluating a new feature vector, only the values of x_i on the boundary are considered and therefore they are referred to as “support vectors”. The acceptance of a new feature vector $\boldsymbol{z}$ is then determined by

$$\|\boldsymbol{z}-\boldsymbol{a}\| = (\boldsymbol{z}\cdot\boldsymbol{z}) - 2\sum_{i}^{N}\alpha_i(\boldsymbol{z}\cdot\boldsymbol{x}_i) + \sum_{i,j}^{N}\alpha_i\alpha_j(\boldsymbol{x}_i\cdot\boldsymbol{x}_i) \leq R^2 \tag{33}$$

where R can be pre-computed by

$$R^2 = (\boldsymbol{x}_k\cdot\boldsymbol{x}_k) - 2\sum_{i}^{N}\alpha_i(\boldsymbol{x}_i\cdot\boldsymbol{x}_k) + \sum_{i,j}^{N}\alpha_i\alpha_j(\boldsymbol{x}_i\cdot\boldsymbol{x}_i) \tag{34}$$

3.6.1.2 Non-linear modeling

As previously mentioned, when the data is not linearly separable in the feature space a kernel function can be applied which provides a transformation to a higher dimensional space such that it is linear separable. Many kernels have been proposed in the literature; however the selection of kernel must be determined explicitly through trial and error for a specific target data set. For speech applications using SVM’s, the RBF kernel is the most widely used since it has been shown to provide the best fit for the nonlinear boundaries of speech. A generic formulation for a kernel K is defined as

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j) = \ \phi(x_i)\cdot\phi(x_j) \tag{35}$$

which is the inner product of the input data transformed by a kernel function ϕ. The RBF kernel is defined as

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j) = \exp\left(-\|x_i - x_j\|^2/s^2\right) \tag{36}$$

where s is a free parameter that can be used to tune the fit of the model to the data and ||.|| represents the norm (distance) between two vectors.

For small values of *s*, the kernel output is approximately zero and all samples are considered as support vectors. The result is an extremely tight fit and no new sample will be accepted. When *s* is large, the kernel output approaches one and very few samples are considered as support vectors. The result is a very loose fit in which all or most samples are accepted. The optimal value of *s* can be chosen using cross-validation in which the expected error of the model for each value is iteratively computed over the entire test sequence. This approach for tuning the model is considered in this thesis and will be further discussed in chapter 4. The kernel function is integrated into the original OC-SVM formulation by replacing the dot products with *K* such that

$$K(\boldsymbol{z},\boldsymbol{z}) - 2\sum_{i}^{N}\alpha_i K(\boldsymbol{z},\boldsymbol{x}_i) + \sum_{i,j}^{N}\alpha_i\alpha_j K(\boldsymbol{x}_i,\boldsymbol{x}_i) \leq R^2 \tag{37}$$

3.6.2 Silence Detection

Since the OC-SVM provides a true decision for clean speech, including silence, a simple silence detector was added in order to gain typical VAD operation under low noise conditions. In addition to the operational performance, this also provides a way to reduce computational load, since classification is not necessary if silence exists. The threshold was determined implicitly from the training sequence using the average energy of the silence frames and can be raised or lowered at runtime if desired.

3.6.3 AND Function

When the output of the energy detector goes high, the use of an AND function is necessary to combine the decisions. The following truth table describes the input and output conditions of this function.

Table 1: AND Function Truth Table

OC-SVM	Energy	Output
0	0	0 (Silence)
0	1	0 (Noise Detected)
1	0	0 (Silence)
1	1	1 (Speech Detected)

In the case that the silence detector is low, the final decision is low. Although unlikely, this provides a means to mitigate a potential false alarm as in the 00 condition. When the silence detector is high, control is handed to the OC-SVM where the prediction determines the output of the AND function.

3.6.4 Weighted Filter

In an actual system implementation, real-time decisions can be made on the frames with any amount of overlap desired. As the purpose of the overlap is to capture as many snapshots of the speech signal as possible for training, such a step may not be necessary during the detection process. Depending on the processing capability of the system, the amount of overlap can be adjusted to provide more or less data points for decision smoothing. If there is no overlap then the final decision on each frame is merely the decision output by the classifier with the tradeoff of potentially not capturing that portion of the speech signal which was previously modeled which may lead to reduced performance.

A benefit to an increased overlap during detection is the addition of more data points which can improve detection accuracy by capturing more of the speech signal just as it did in training. In order to make the overlapped decisions useful, a weighting filter is considered. The decision made on a single frame can be considered a combination of the adjacent frames when there is an overlap. The final output decision after weighting is defined by

$$d(i) = \sum_{i-1}^{i+1} d(i)\,w(i) \tag{38}$$

where *d* is the output decision of the AND function, *i* is the frame number and *w* is the weights. For this thesis, *w* was chosen to be

$$w = [1\ \ 1\ \ 1] \tag{39}$$

The weighting values selected are based on the fact that the overlap used in the simulation section is set to 50%, which implies that frames adjacent to the current frame each contribute to their adjacent frames. If a greater overlap was chosen a longer filter would be necessary. The downside to this approach is the introduction of a single frame delay which is tolerable. The plus side is that for higher noise environments, it provides a way of slightly improving the detection rate by decreasing miss detections.

Once the weights are applied, a simple threshold can be used to determine how the current frame is affected by the adjacent frames. If a threshold of 1 is used, the filter allows the current frame to pass if any frame is 1. This will account for miss detections while letting false positives through and slightly increasing the detections around onsets and offsets of speech. A threshold of 2 will pass only if adjacent or consecutive detections are 1 which aims to mitigate false positives by requiring that speech be present in adjacent frames. Finally, a threshold of 3 will only pass when all detections are 1 which would decrease false positives and make the decisions around onsets and offsets of speech tighter. A threshold of 1 was chosen to provide the best performance at lower SNR.

3.7 Other considerations

The addition of a VAD that can accurately model speech allows the gateway to determine if speech is present and therefore can make the decision to let a network packet pass or not. In an actual communications system, a VAD alone may not provide a complete solution since the audio would be broken during speech pauses and silence. Since the goal is to only pass speech and block everything else, additional components such as a comfort noise (CN) generator would allow for silence and pauses to be filled in order to maintain a natural conversation. Additionally, improved performance at higher noise levels is typically accomplished by adding temporal smoothing. Although not considered in this thesis, temporal smoothing is a decision buffering method that typically employs a state machine which uses a priori information about speech duration to filter out false positives and miss detections.

The proposed design inadvertently provides the potential to prevent or reduce steganographic attacks. In [63], the author describes several signal processing techniques that can be implemented in VoIP systems in order to prevent hidden message passing. These methods include adding white noise and/or jitter, inducing random packet loss, resampling and frequency shifting. Based on the system design, the VAD directly creates sample loss although it is somewhat deterministic since the goal is to remove pauses and silence. The implementation of the CN generator would provide a noise addition which is a similar to adding white noise. Also, the author in [64] suggests that wiener filtering can provide an attack against embedded watermarks, which allows the wiener filter described here to serve a dual purpose.

CHAPTER 4 EXPERIMENTS

4.1 Setup

In order to test the performance of the design, custom signal processing algorithms as well as several open-source tools were used in a MATLAB environment. Multiple audio corpuses were selected to allow for analysis of the training process which helped derive an appropriate method for tuning the VAD and verify its performance. A high level diagram of the training and testing process can be seen in Figure 8. First, the TIMIT database was used to train the OC-SVM. Next, samples of several noise types from the NOISEX-92 database were used to derive selection criteria such that a minimal amount of noise would be accepted by the model. Finally, the OC-SVM was validated against speech from the NOIZEUS corpus and simulated noises for analysis. This chapter describes each of the components associated with this evaluation and presents the results and discussion.

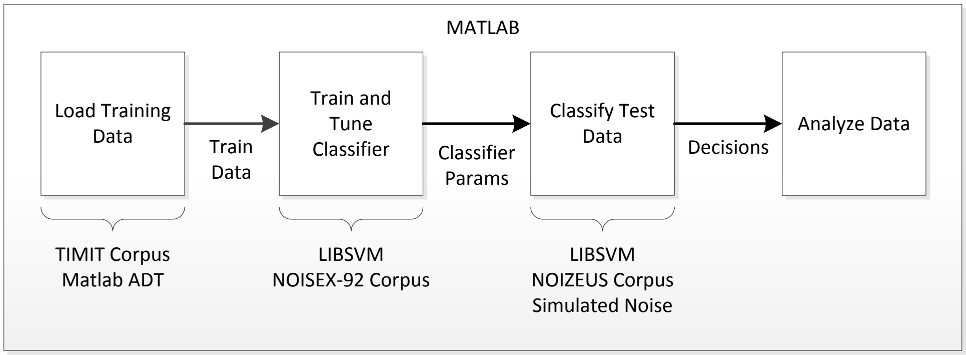


Figure 8: Simulation Setup

4.1.1 Audio Corpus

Several audio corpuses were utilized in order to validate the performance of the proposed VAD. To train the OC-SVM, the TIMIT database provided samples of clean speech without the presence of noise. The goal was to establish a well trained, clean speech model using a noise-robust feature set such that features extracted from noisy speech would be a near representation of the clean features. The NOISEX-92 database helped to provide an understanding of how well the speech model would reject noises of various types in order to establish a method for tuning the VAD. To test the model, the NOIZEUS database and several simulated noises were utilized in order to understand the out of database performance for clean and noisy speech as well as see how well the model rejects other noises not considered in the tuning process.

4.1.1.1 TIMIT Corpus

The TIMIT database is a clean speech corpus consisting of 6300 phonetically-rich sentences spoken by 630 speakers with 8 different dialects. Its original purpose was for the development and evaluation of speech recognition systems, however it can now be widely found in literature for the general purpose of developing speech processing algorithms. The database is partitioned into training and testing data and is provided at a sample rate of 16 kHz. Since the application is concerned with speech data only, the data was resampled to 8 kHz in order to match the typical audio bandwidth found on VoIP. For training purposes, one male and one female was randomly selected from each dialect region to provide a total of 16 sentences. More utterances were considered, however the added data did not contribute to performance. Filtering the data set in this fashion allowed for variability in the dialect and gender of the speaker while keeping the amount of data low to prevent over-fitting.

4.1.1.2 NOIZEUS Corpus

The NOIZEUS database is a speech corpus consisting of 30 phonetically rich clean-speech sentences produced by three male and three female speakers. Each sentence is combined with several different noise types at various noise levels making this database useful for measuring and comparing the performance of speech enhancement algorithms. The noise types include babble, car, hall, restaurant, street, airport, train station and train with SNR's of 15, 10, 5 and 0 dB for each. The data is provided at 16 kHz was resampled to 8 kHz for use in testing. Each clean speech utterance was hand labeled in order to validate the performance of the proposed VAD at various SNR levels.

4.1.1.3 NOISEX-92 Corpus

The NOISEX-92 database provides various noise recordings for different environmental sounds and includes white and pink noise; HF channel noise; speech babble; factory floor noise; jet cockpit sounds; Destroyer ship engine and operations room; tank noise; machine gun; and a Volvo car. The recordings are provided at ~20 kHz and are resampled to 8 kHz similar to the other databases. Approximately one second of audio was extracted from each noise type which was used to measure the acceptance rate under different training configurations such that and appropriate setting could be selected.

4.1.1.4 Matlab ADT

In order to manage the TIMIT data for training, a tool called MATLAB ADT (Audio Data Toolbox) [65] was utilized. It was developed by the Technion Electrical Engineering Department to provide a simple and efficient means of extracting data from various corpuses without the hassle of manually parsing the data. The use of this toolbox allowed for a significant reduction in implementation time while also allowing for iterative evaluations to be performed in an efficient manner.

4.1.2 Training & Testing Setup

4.1.2.1 Data Scaling and Normalization

Scaling and normalization are sometimes essential pre-processing steps which allow for a machine learning algorithm to function properly. In particular it is common to perform this for the SVM. To ensure that the features considered contribute equally to training, each was scaled between 0 and 1. This was performed by shifting each direction up by the minimum value in that direction and then dividing by the maximum in each direction. After scaling, the mean of each direction was subtracted and then scaled to unit variance. The values computed from scaling and normalization were then saved and applied to each of the test sequences.

4.1.2.2 Data Labeling for Supervised Training and Testing

As part of the training portion of the simulation, labeling the data was an essential step of the supervised training process. Although each of the sentences within the database is mainly comprised of speech, pauses and silence were still undesirable. Since the OC-SVM requires a supervised training set and the TIMIT database does not provide this, it was necessary to develop a method for speech extraction. Relying on the fact that the TIMIT is a clean speech corpus with a minimal amount of background noise, an energy method was employed.

This was accomplished by comparing the instantaneous energy against a long term average of the background noise energy. The decision threshold was computed as the ratio of instantaneous to background variances and a value of 1.1 was used for the TIMIT database. The selection of this value was set low enough such that silence and true pauses could be avoided while still including voiced and unvoiced segments of speech.

The implementation was as follows. An initial energy computation was made during a small period of time at the beginning of the audio file in order to initialize the noise average. The energy of each successive frame of small duration was then compared against the noise average to determine if speech was present. Using this method, each sample in the frame was assigned a 1 if speech was present and a 0 if it was not. If the frames were determined to not contain speech, the energy contribution was used to update the long term average. During frame based Gammatone feature extraction in training and testing, if 75% of the samples in a frame was labeled as a 1 the entire frame was labeled as a 1; otherwise the frame was labeled with a -1.

The testing portion of the data was hand labeled using Audacity which provides a means for adding and exporting labels to an audio sequence. These labels were then imported into the Matlab workspace to be used as the true decisions.

4.1.2.3 SVM Implementation

Several open source tools are available which provide SVM functionality in the form of software libraries that are useful for accelerating research projects. Although understanding the inner workings of the SVM is important, the goal of this thesis was not to reinvent the wheel, but rather focus on utilizing its strength as a machine learning tool in order to evaluate the proposed design. The advantage of this is reduced implementation time by utilizing code that has been developed and peer tested. Several implementations were considered and are listed as follows:

1. Native MATLAB functionality
2. SVM-Light
3. LIBSVM

Even though all implementations provide MATLAB functionality, the LIBSVM tool was chosen for several reasons. The code base offers support for several SVM algorithms to include the standard two-class implementation and as well as both OC-SVM's found in the literature. An additional benefit was the availability of extensive documentation through the LIBSVM website[1] and online support through online forums[2].

4.1.3 Performance Analysis

Identifying the measurement criteria for evaluating the performance of the VAD is critical to this research. Many methods exist for capturing this data to include individual prediction errors, accuracy & precision, Receiver Operating Characteristic (ROC) curves and various speech clipping criteria. Prediction errors are the result of inaccurate modeling that occurs when the classifier makes a wrong decision. Accuracy describes the number of correct decisions made and is directly related to the prediction errors. The ROC curve provides a way of visualizing the tradeoff between accuracy and error as certain parameters or thresholds are varied throughout their usable range. Speech detectors can induce clipping in various parts of speech, primarily those that are unvoiced and contain a weak amount of energy. To capture this type of performance, front-end, mid and back-end clipping can also be measured. For this thesis, the prediction errors are computed over a range of parameter values to tune the OC-SVM; accuracy is considered to measure performance over various noise conditions as well as verify consistency in the training model; and speech clipping is discussed within the results.

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[2] http://www. kernel-machines.org

4.1.3.1 Understanding the Errors

Analyzing the prediction performance for any classifier begins with hypothesis testing in which the null and alternative hypotheses are clearly defined. For the OC-SVM VAD, the null hypothesis (H_0) indicates the presence of noise, while the alternative hypothesis (H_1) indicates the presence of speech. The decision output by the classifier is compared against the ground truth label and measured for accuracy. Table 2 below summarizes the possible outcomes and a more detailed explanation is provided in the Appendix.

Table 2: Hypothesis Test Outcomes

	Labeled as Noise (H_0 True)	**Labeled as Speech (H_0 False)**
Speech Detected (H_0 Rejected)	False Positive (Type I Error)	True Positive
Noise Detected (H_0 Accepted)	True Negative	False Negative (Type II Error)

The prediction errors are divided into two types:

1. Type I Error – the classifier detects speech when the true label is noise
2. Type II Error – the classifier detects noise when the true label is speech

The Type I Error is concerned with the performance of the classifier in terms of how well it rejects noise while the Type II Error is concerned with how well it accepts speech. In [41], the author describes a method for estimating the Type II Error for the OC-SVM explicitly from the training data[3]. Using a leave-one-out cross validation method, it is shown that the expected value of the error can be approximated by

[3] The author aligns the definition of the null-hypothesis with the target pattern. For speech detectors, it is common to align the null-hypothesis with negative labeled data (noise) therefore the Type I and Type II Errors are reversed here as compared to the author definition.

$$E[P(error\ II)] \sim \frac{\#\mathrm{SV}}{N} \tag{40}$$

where #SV is the number of support vectors that describe the data and N is the amount of training data. As this equation is merely an approximation, it only describes the upper bound on the Type II Error. The true error will be larger since the model is not an exact representation of the data, however use of this equation allows for visual tuning of the classifier. With the Type II Error for an OC-SVM defined, the cost parameter *C* and the RBF width parameter *s* can be determined by identifying an acceptable amount of speech rejection while minimizing the noise acceptance.

As stated previously, *C* is a cost associated with classification and provides a soft mechanism for solution convergence that can take values greater than 1/N. If *C* is small, the upper bound on the Lagrange multipliers is small therefore more of the data will be considered in the boundary decision in order to satisfy Eq. (31). If *C* is large, the upper bound for the Lagrange multipliers is large; therefore the description will be achieved with fewer support vectors. When the RBF kernel is used, the value of *s* provides similar functionality and therefore it is recommended that *C* be fixed to a predetermined value. For the experiments presented here the value of *C* was set to .1 such that a tight description, where all data points can be defined as an SV, is achieved for small values of *s.*

With the RBF width as our free parameter, the Type II Error can be plotted across a range of values as seen in Figure 9. From this plot we see that as the value of *s* increases, the number of SV's decrease which decreases the speech rejection rate. By using a value of .1 for *C,* the curve converges close to 0 at which point the data is described by the minimum number of support vectors required to accept approximately 100% of the data. If the value of the *C* were reduced towards 1/N, the expected value of the error would be greater thereby limiting the minimum achievable error rate.

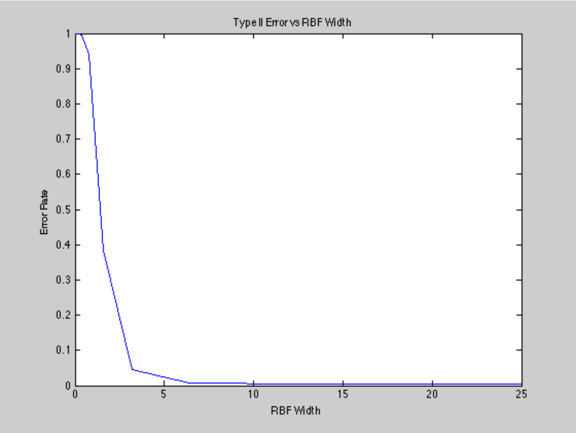


Figure 9: Type II Error vs. RBF Width

Since minimizing the Type II error is equivalent to increasing the probability of accepting speech, a high value of *s* would seem desirable. However, increasing *s* comes at the cost of increasing the Type I Error which allows for significant overlap between speech and noise. Understanding this trade-off requires knowledge of the Type I Error performance which can only be obtained through either simulating or capturing the noise distribution surrounding the training data. In order to visualize the impact of the Type I Error on the speech model, various noise types from the NOISEX-92 database were plotted with varying RBF width as seen in Figure 10.

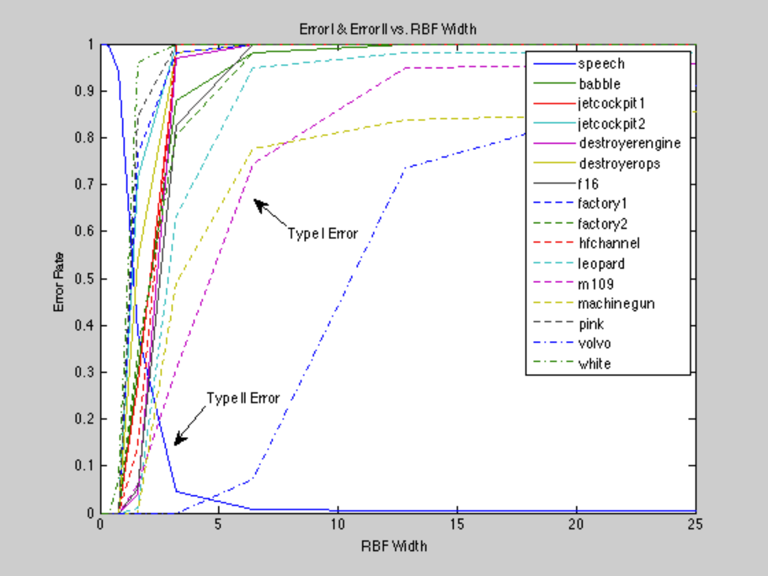


Figure 10: Type I & Type II Error vs. RBF Width

With examples of Type I Errors plotted for varying RBF width, it can be seen that the noise intersects with the speech model for low values of *s* and rises quickly to 100% error for most noises as *s* increases. Choosing a value of *s* below this intersection would reject most of the noise, but also reject most of the speech. In [66] the authors show that the Type I Error can be decreased by increasing the dimensionality of the model. In order to accomplish this for speech detection, more features must be included in the description. Alternatively, a noise reduction method such as the Wiener filter can provide a means of reducing this error as seen in Figure 11.

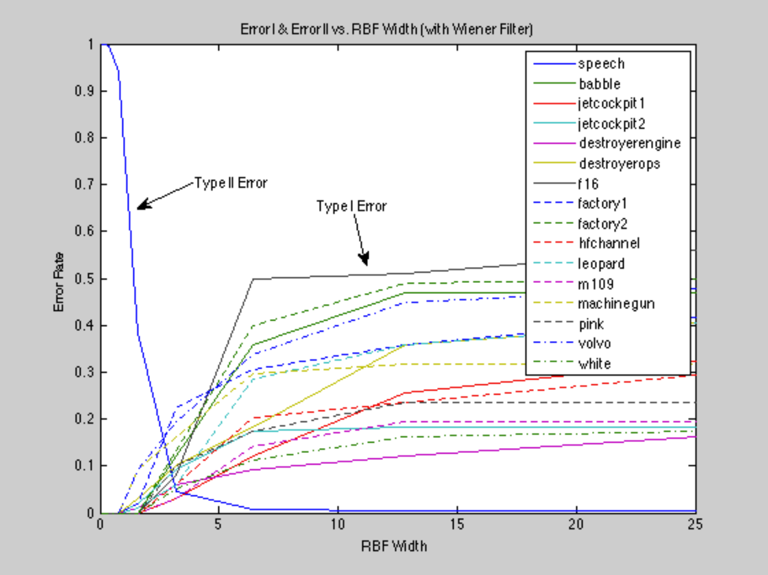


Figure 11: Type I & Type II Error vs. RBF Width (Wiener Filtered)

From this plot it becomes obvious that incorporating the Wiener filter provides a significant improvement in the Type I Error allowing an appropriate value of *s* to be chosen such that much of the noise is rejected while maintaining an accurate speech model. Even for large values of *s* many of the noise types have a low probability of error. Although the Wiener filter requires additional computational resources, its inclusion allows for a significant increase in RBF width which reduces the computational complexity of classification. It is expected that increasing the dimensionality of the model would further reduce the error rate and will be considered for future analysis to ensure meaningful features are chose so as to not negatively impact performance.

4.1.3.2 Error Measurement

The accuracy the speech model can be measured based on the results from hypothesis testing. To capture performance in the experiments, the measurements considered include the

True Positive (TP) Rate and False Positive (FP) Rate which is typically used in ROC analysis. The TP Rate is defined as

$$TP\ Rate = \frac{TP}{P} \tag{41}$$

which measures the total number of accurate speech detections out of the total number of frames labeled as speech. The FP Rate is defined as

$$FP\ Rate = \frac{FP}{N} \tag{42}$$

which measures the total number of speech detections out of the total number of frames labeled as noise. The ideal performance in any condition would have a TP Rate of 100% and an FP Rate of 0%.

Traditionally, an ROC curve is the most popular method used to evaluate a speech detector. It was not considered here for several reasons. First, the error analysis presented here provided a similar functionality by allowing for the optimal selection of the free parameter *s*. Second, the ROC curve is commonly found in literature when a comparison is being performed against other speech detection algorithms which were not considered in this thesis.

4.2 Experiments

For all experiments, the test data was sampled at 8 kHz with a 25ms window and 50% overlap. Before testing under clean and noisy speech conditions, the first experiment performed helped to provide insight into the effects of the feature set with respect to noise rejection. Once the ideal number of features was selected, the speech model was then tested under clean speech conditions in order to identify a suitable range for *s* that gives a good TP and FP rate as well as verify that the speech model provided an unbiased representation of speech. Following this, a narrower range of *s* was then chosen such that speech acceptance and noise rejection could be viewed over many noise conditions and SNR levels. Finally, for a given value of *s* the speech detector was validated against several non-speech and multiplicative noises to understand the performance under such conditions.

4.2.1 Experiment 1 - Tuning the Feature Set

Although increasing the number of features may improve performance it may also hurt performance. To understand this property, the chosen features were varied in number with a large value for *s* such that the lowest expected error could be obtained. For this experiment the number of cepstral coefficients was varied from 5 to 13 and the number of Gammatone filters was varied from the number of cepstral coefficients plus 1 to 21 for each number of cepstral coefficients. In all cases the frequency features were not removed.

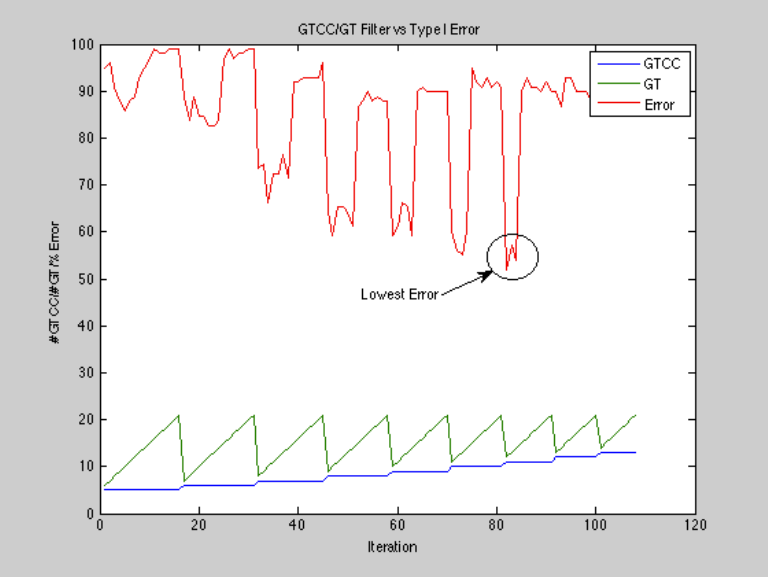


Figure 12: Feature Tuning

In Figure 12 it can be seen that then when the number of Gammatone filters is one to three more than the number of cepstral coefficients the error is lowest. Likewise, as the number of cepstral coefficients is increased the error around this point decreases overall. Based on the plot it can be seen that the optimal number of cepstral coefficients is 11 and the number of Gammatone filters is 12 to 14. Several iterations showed that 13 Gammatone filters gave a consistent result.

4.2.2 Experiment 2 - Validating the Speech Model

With the optimal number of features selected, the next step was to understand the selection for *s* which gives the best overall performance under clean speech conditions. Using the errors described in section 4.1.3.2, each of the clean speech utterances from the Noizeus database were processed and the average TP Rate and FP Rate was recorded separately for male and females at several values of *s*. The ROC was not used in this case so as to display the value of *s* over the measurements.

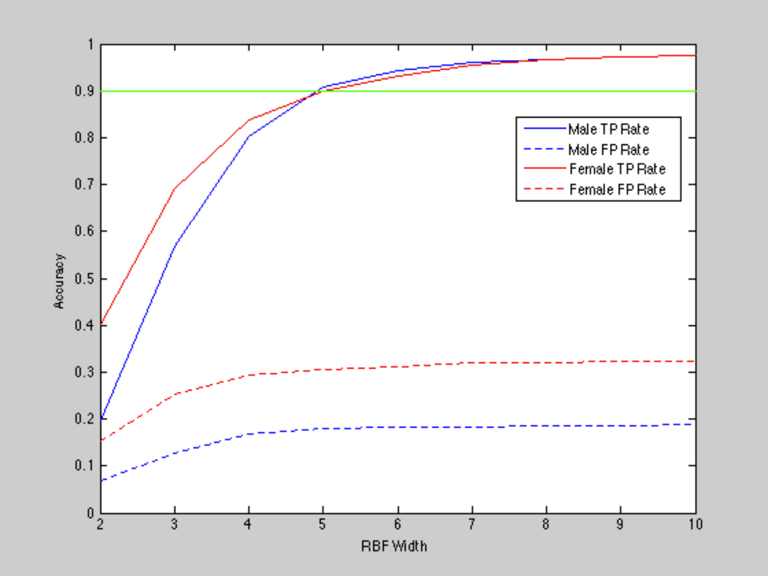


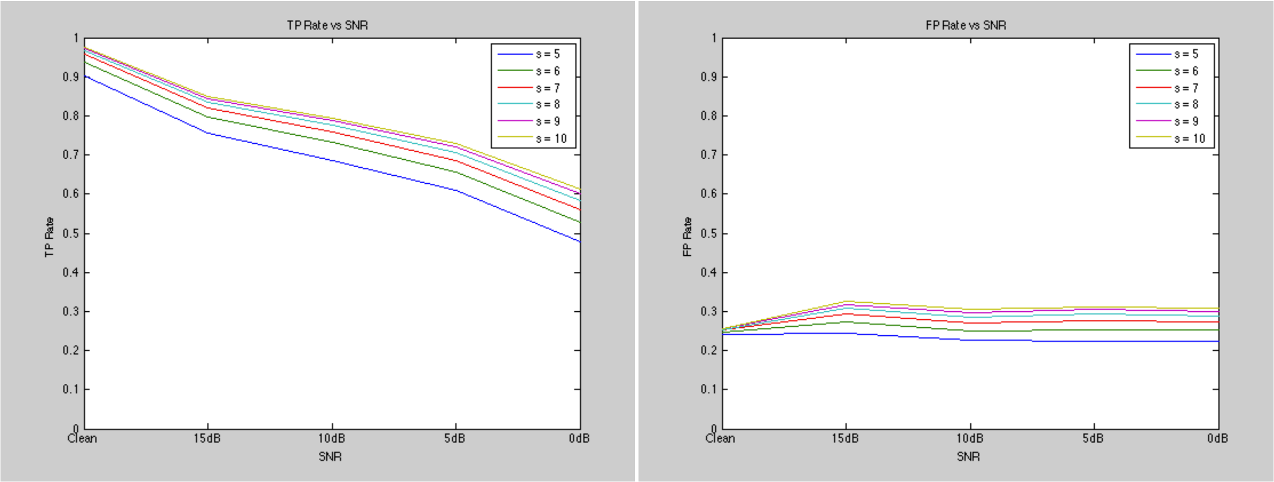
Figure 13: Clean Speech TP and FP Rate Tradeoff

Several important things can be noticed from Figure 13. First, males and females were plotted separately in order to ensure that the model equally captured both. For a value of *s* greater than 5, a TP rate greater than 90% is achieved and both plots are nearly identical while the FP Rates vary by approximately 11%. This difference in FP Rate can be partially attributed to the manual labeling in which the amplitudes of the signals were used, where the male voices may have been easier to transcribe and therefore a lower FP Rate was obtained. Inspection of the labeling showed that the predictions around the onsets and offsets of utterances had more overhang with the females as compared to males. Second, the test sequences used here were out of database indicating that there was effectively no bias on the training set from the TIMIT corpus. Lastly, it can be seen the TP Rate of the curves follow the Type II Error curve from Figure 9 quite nicely which reinforces the expected value of the error given the out of database testing.

Given the range of *s* that provides greater than 90% accuracy it can be seen that a tradeoff still exists for Type I errors. In terms of ROC analysis this indicates that for larger *s* the TP Rate and FP Rate are higher while for a lower *s* the TP Rate is slightly lower and the FP Rate for noise may be lowered by as much as 20% as seen in Figure 11. Thus a decision must be made based on the desired operating characteristics. In the case of the application presented here, a lower *s* would be desirable in order to reduce the number of Type I errors as much as possible, assuming that all communications are operating under reasonable noise conditions. In the case of a traditional speech detector, a higher TP Rate would be desired so as to allow more noise such that speech with additive noise is accepted by the detector thus increasing the acceptance rate of speech under high noise conditions.

4.2.3 Experiment 3 - Testing with Speech and Noise

In order to understand the noise performance for the range of *s* presented in the previous experiment, noise was added to the same clean speech utterances at various SNR levels for different noises. Each utterance was processed by the detector for all noises and then averaged over all utterances. In this manner, a general noise performance can be viewed irrespective of a particular type of noise.



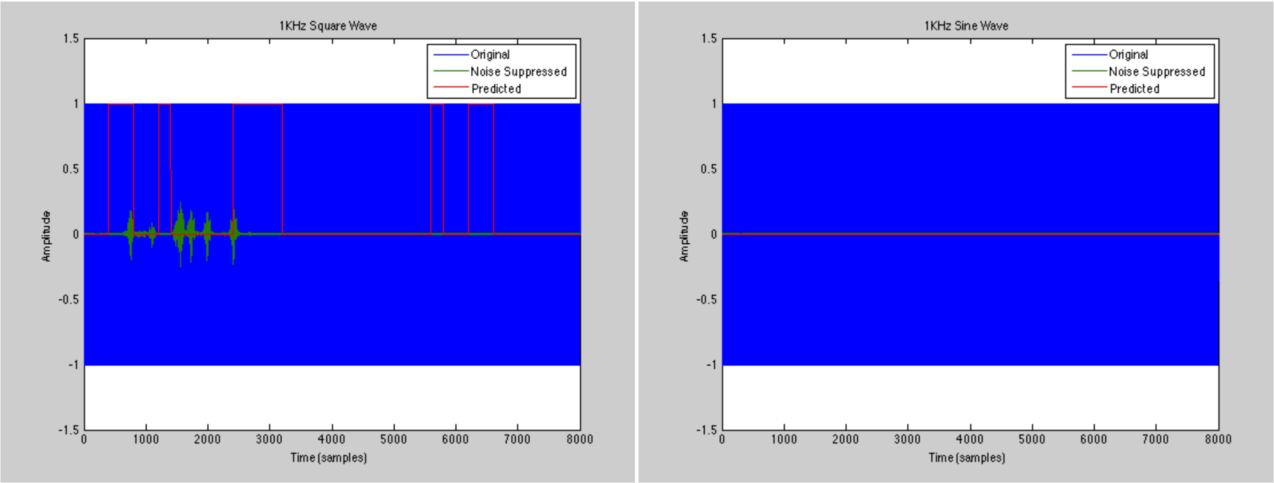
(a) (b)

Figure 14: Speech Detector performance for varying values of RBF width and SNR

In Figure 14 it can be seen that for decreasing value of SNR, the TP Rate decreases while the FP Rate stays relatively constant. This is as to be expected since adding noise to speech changes its characteristics causing the rejection rate to increase. Also, for increasing value of *s* both the TP Rate and FP Rate increase by nearly 10% over all SNR values which is quite significant. Audible hearing tests indicate that for lower SNR values the Wiener filter adds a musical distortion to the speech which is the likely reason that acceptance by the detector decreases. In terms of improvement, using a value of 10 for the RBF width would be a reasonable choice for a standard VAD since a higher noise acceptance is tolerable if it improves the speech acceptance rate. For the final experiment a value of *s* equal to 5 was chosen in order to see how well the noise is rejected when tuned to the more desirable value for this thesis.

4.2.4 Experiment 4 - Testing against Noise Only

Several noises were simulated in order to test the performance of the system under non-speech or multiplicative conditions. These noises include sine wave, square wave, speech modulated with a square wave and a jpeg image which has been decoded as speech. The results for each are provided below. In the case of the non-image signals, each was produced at a sample rate of 8 kHz and duration of 1 second.



(a) (b)

Figure 15: Speech detector examples when the input signal is (a) a 1KHz square wave and (b) a 1KHz sine wave

Example results for a square wave and sine wave can be seen in Figure 15 (a) and (b) respectively. For the square wave some false positives were present while for the sine wave no false positives were present. The output of the noise suppressor did a fairly good job of reducing the signal while in some areas it produced tonal segments which were accepted by the detector.

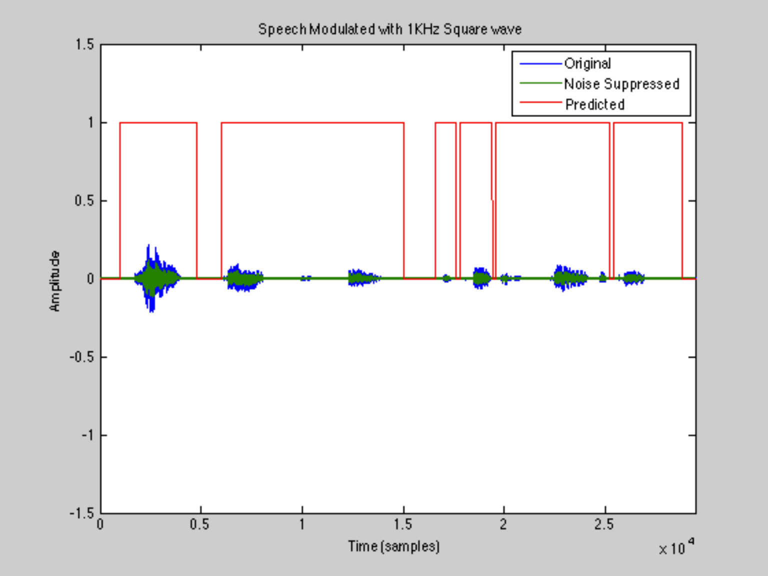


Figure 16: Speech detector output when input signal is speech modulated with a 1KHz Square wave

In Figure 16 an example of speech modulated with a 1KHz square wave can be seen before and after noise reduction. Since the square wave used was modulated by the entire speech sequence, the start of the transmission had relatively no components of the square wave and thus the Wiener filter was unable to capture the statistics of the noise. For the portions of the signal that did contain the square wave and speech, the detector was unable to differentiate it as non-speech and therefore it was accepted by the model.

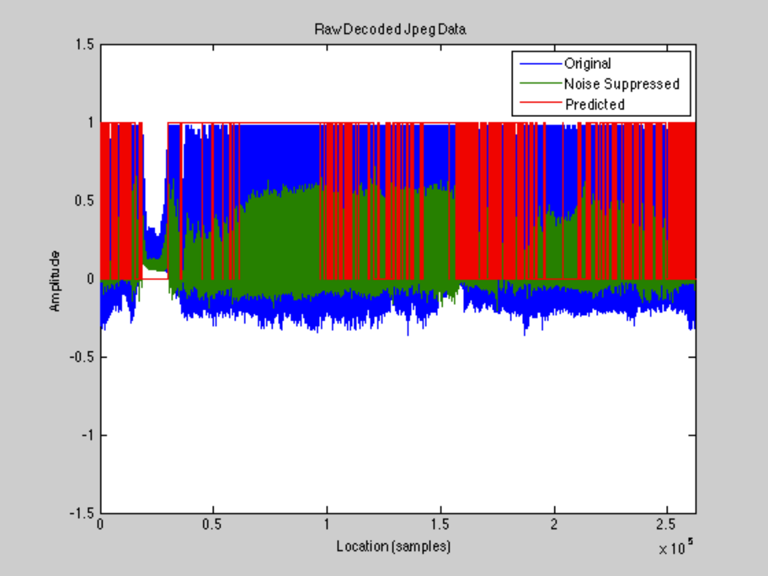
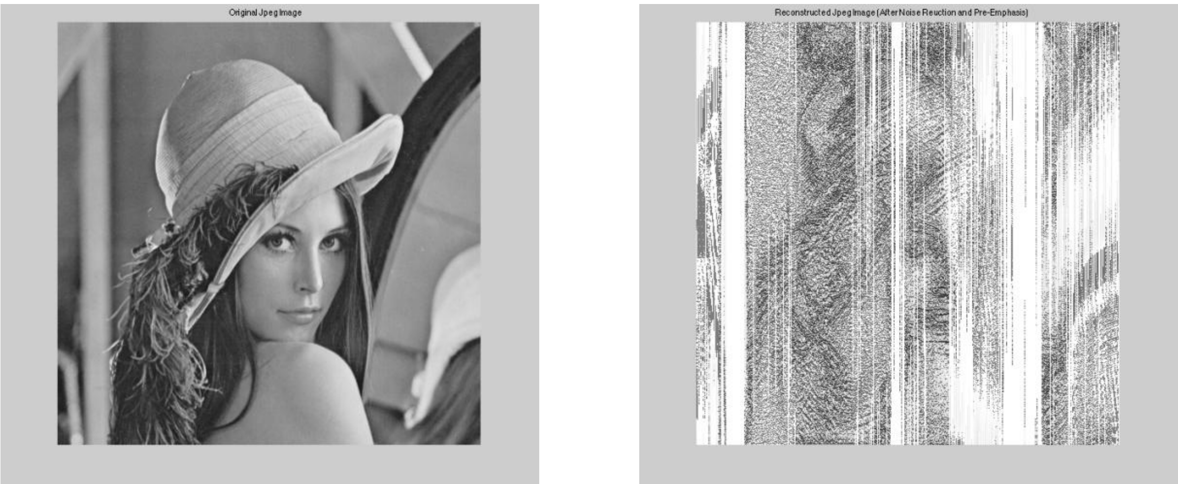


Figure 17: Speech Detector output when input signal is Raw Decoded Jpeg data.

In the event that the voice packets are used to transport a non-speech signal, it would be ideal if the detector could prevent it from passing. To test this idea, a jpeg image as seen in Figure 18 (a) was considered as a stream of bytes which were mu-law decoded and processed by the speech detector. The image, represented as an audio signal, along with detector predictions can be seen in Figure 17. It is clear that many portions of the image data passed through the speech detector although detector oscillations indicate that it was likely on the edge of the model. The Wiener filtered and pre-emphasized image signal are shown in Figure 18 (b) where it can be seen that the image has been severely degraded to the point that only a faint figure of the original image is present. In this case the speech detector might be considered in line with the communications channel such that signal processing is applied to the incoming audio before passing through. Alternatively, the speech detector could be used as a simple indicator of speech presence and would require less consideration for implementation in terms of quality of service.


Original Jpeg Image
Reconstructed Jpeg Image (After Noise Reuction and Pre-Emphasis)

(a) (b)

Figure 18: Raw Decoded Jpeg data (a) input to and (b) output of the speech detector.

False positive rates for each of the signals presented in this experiment can be seen in Figure 19. In all cases the FP Rates are less than 50% which reflects the noise analysis performed in section 4.1.3.1. Also included is the FP Rate for a compressed jpeg image which showed a much higher rejection rate than its uncompressed counterpart. The improvement on the compressed image is due to the fact that the encoded image data mimics a “white noise” signal since the DCT applied to the original image acts as a decorrelation stage.

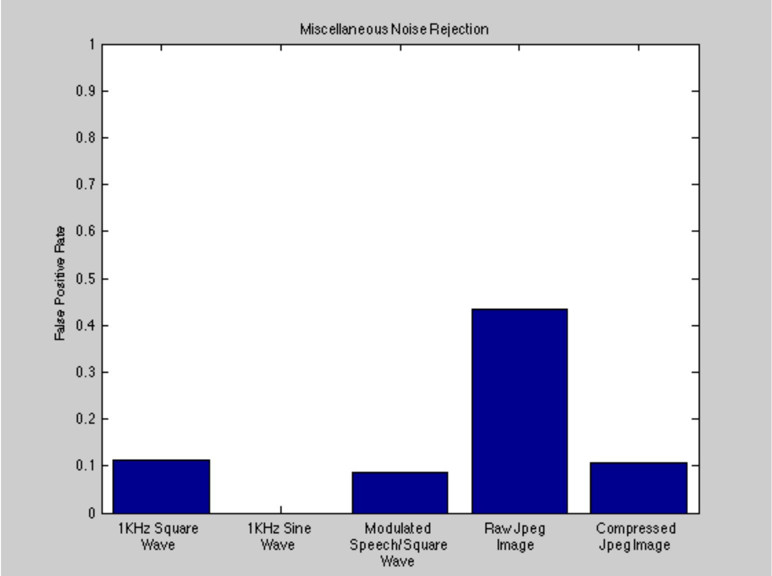


Figure 19: False Positive Rates of the Various Noises

4.3 Summary

The results show that the chosen method is a promising approach to speech detection. Although the OC-SVM must be tuned to provide a loose description of the speech data, the integration of the noise reduction stage improves the speech discrimination performance. Audible verification of the noise reduced speech showed musical distortion that made the speech not understandable yet still speech like. Use of a better noise reduction method may reduce this distortion making it more useful in an actual communications system. It is worth mentioning that the method of tuning presented can be extended further by fixing the value of *s* at the desired error and then using the radius of the sphere required to a test vector as a decision mechanism. In doing so, both *s* and the radius value could be optimized using a cross-validation procedure.

The computational complexity of the proposed design was considered in terms of time needed to process a single frame. For the implementation, optimization of the code was not considered and simulation was performed inside the MATLAB operating environment (non-

MEX function). Given a 25ms window and 8kHz sample rate, a decision on each frame can be made in approximately 2.4ms which is less than the 12.5ms frames given a 50% overlap. Optimization of this implementation would further improve computational time required per frame.

CHAPTER 5 CONCLUSIONS

5.1 Overview

The goal of this thesis was to design a speech detection system for network gateways deployed in VoIP based communication systems. The integration of such a speech detector can provide an additional measure for filtering voice data traffic between networks so as to prevent or reduce malicious transportation of non-voice data. Since current gateways can only validate data fields surrounding the speech samples, the research presented in this thesis aimed to fill this gap. Requirements were developed that helped choose the best speech feature and detection method to be used in the design. In doing so, this thesis provided a contribution by combining the noise robust Gammatone Cepstrum and frequency based features with a One-Class SVM which enabled a speech detector to be trained independent of environmental noise.

Several aspects of a speech detection system were considered to include feature extraction and detection methods. An overview of the most common feature extraction mechanisms was provided along with a brief discussion of the pros and cons of each. Through literature reviews, it was found that perceptual features based on the human auditory system exhibited the best performance in a noisy environment. Of the available perceptual features, the Gammatone Cepstral Coefficients were chosen based on improved noise performance over the widely used MFCC while maintaining equally efficient computational complexity. Additionally, frequency features were computed directly from the Gammatone filter outputs for increased dimensionality with little added cost. Exploiting the frequency domain allowed the Gammatone filters to be pre-computed for greater efficiency in a real-time environment.

The SVM was chosen from the handful of machine learning tools available due to its inherent binary decision, computational efficiency and generalization properties. The downside to the basic two-class SVM is that both the speech samples and noise data must be available in order to achieve reasonable performance in environments with varying noise types. A high degree of performance can be achieved when the noise is known such as that of a car. Obtaining noise data for training can be cumbersome for complex environments and it is not possible to capture all potential environmental noise characteristics. Therefore an alternative detection method was desired that could avoid the need to train on the noise. Moving from a two-class SVM to a one-class SVM allowed the design to train independent of the noise by modeling only the data provided in the clean-speech corpus. However, the downside to the one-class approach was a noticeable cost of decreased performance with respect to noise overlap in the speech model. By adding a Weiner filter prior to classification the detection rate increased considerably, but at the cost of an additional pre-processing step.

5.2 Future Directions

Improving the overall performance of the proposed speech detector can be achieved in several ways. The noise masking characteristics can be enhanced through the selection of a feature set which provides more accurate auditory modeling. One such feature would be the Gammachirp filter [67] which is a generalized form of the Gammatone filter that incorporates the asymmetric properties of the human auditory system providing better noise masking. The complexity would remain unchanged if frequency domain processing was used.

The addition of a dynamic nonlinear operation on the output of each filter could also increase noise performance [68] by exploiting AM-FM demodulation characteristics of speech.

Such a feature is believed to take place in the human auditory system. Inclusion of this would necessitate time domain processing in order to capture temporal and amplitude variations requiring an increase in computational cost.

Since the OC-SVM exhibited poor discrimination between speech and noise without noise suppression, using a larger feature vector by combing other speech based features could potentially enhance performance. Doing so might allow for removal of the noise suppression step while the disadvantage would be an increase in computational complexity. Utilization of the Gammatone energies could potentially be reused for such a task similar to the frequency features. With more features added, a more in-depth performance analysis could be obtained in non-stationary environments where the filtering of human voice might act as a pre-processing step for DTX communications or speech recognition. Such a study of non-stationary noise response might also include comparing performance against other popular methods or commercial standards.

When a clean speech signal passes through the OC-SVM alone, the entire signal is classified as speech (including silence). As noise is added to the audio signal, the OC-SVM is pulled down to zero. This property can be described as an anomaly or outlier detection where trained conditions give a positive decision (clean speech and silence) while untrained conditions (noise) cause outliers and therefore a negative decision. Since increasing noise amplitude can reduce unvoiced detection considerably, detection boundaries should have a larger overhang of speech onsets and offsets to account for this. The anomaly detection property could potentially be used as an indicator of high noise which can be used for online tuning of a decision smoothing algorithm that becomes loose at high SNR and tight at low SNR.

The main goal of this research was to provide a safeguard against potential misuse of the voice data that is transported in VoIP networks. During this research effort, alternative mitigation strategies were identified [63] in which the goal was to degrade the audio just enough to increase the bit error rate (BER) of a potential attack while minimizing the impact on audio quality. Use of a VAD as a prevention method was not described, most likely due to its deterministic nature that could allow for exploitation by only sending embedding data during periods of speech. However, no VAD is 100% accurate and therefore it can still provide some usefulness since some BER would be induced while also having the advantage of verifying that voice is present.

APPENDIX: HYPOTHESIS ANALYSIS

The hypothesis tests performed for speech detection are defined as follows. The null-hypothesis, H_0, argues that noise is present while the alternative hypothesis, H_1, argues that speech is present. If we hold one argument, either H_0 or H_1, to be ground truth then we can derive errors associated with each based on the classification predictions.

For analysis of speech detection performance, each audio sample in our test vector is labeled as one of two classes: speech (assigned a value of 1) or noise (assigned a value of 0). These labels provide the ground truth of our hypothesis. For each test vector, features are extracted and the processed by a trained classifier which provides a prediction.

Let's first consider performance with regards to positive labels or H_1. If the classifier predicts a sample to be noise when we not it is true, then a False Negative or Type II Error has occurred because the classifier has accepted H_0 rather than H_1. If the classifier predicts a sample to be speech then our classifier made the right decision and a True Positive has occurred because the classifier has accepted H_1. We can do the same analysis for the negative labeled data or H_1. If the classifier predicts noise, then a true negative occurs. If the classifier predicts speech than a False Positive or Type I Error has occurred.

LIST OF REFERENCES

[1] R. G. Bachu, S. Kopparthi, B. Adapa, and B. D. Barkana, “Separation of Voiced and Unvoiced using Zero crossing rate and Energy of the Speech Signal,” *American Society for Engineering Education (ASEE) Zone Conference Proceedings*, pp. 1–7, 2008.

[2] Y. Yatsuzuka, “Highly Sensitive Speech Detector and High-speed Voiceband Data Discriminator in DSI-ADPCM Systems,” *IEEE Transactions on Communications*, vol. COM-30, no. 4, pp. 739–750, 1982.

[3] M. Ito and R. Donaldson, “Zero-Crossing Measurements for analysis and recognition of speech sounds,” *IEEE Transactions on Audio and Electroacoustics*, vol. AU-19, no. 3, pp. 235–242, 1971.

[4] E. Scheirer and M. Slaney, “Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator,” *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 2, pp. 1331–1334, 1997.

[5] N. Cho and E. Kim, “Enhanced voice activity detection using acoustic event detection and classification,” *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 196–202, Feb. 2011.

[6] G. Tzanetakis, S. Member, and P. Cook, “Musical Genre Classification of Audio Signals,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[7] “Sound Event Classification Based on Feature Integration, Recursive Feature Elimination and Structured Classification,” *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pp. 177–180, 2009.

[8] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, “Audio-based context recognition,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 321–329, Jan. 2006.

[9] K. Li, M. N. S. Swamy, L. Fellow, and M. O. Ahmad, “An Improved Voice Activity Detection Using Higher Order Statistics,” *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 965–974, 2005.

[10] E. Nemer, R. Goubran, S. Mahmoud, and S. Member, “Robust Voice Activity Detection Using Higher-Order Statistics in the LPC Residual Domain,” *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 3, pp. 217–231, 2001.

[11] J. Sohn, S. Member, N. S. Kim, and W. Sung, “A Statistical Model-Based Voice Activity Detection,” *IEEE Signal Processing Letters*, vol. 6, no. 1, pp. 1–3, 1999.

[12] J. Makhoul, “Linear Prediction: A Tutorial Review,” *Proceedings of the IEEE,* vol. 63, no. 4, pp. 561–580, 1975.

[13] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. New Jersey: Prentice Hall, 2001, p. 980.

[14] H. Hermansky, “Perceptual Linear Predictive (PLP) analysis of speech,” *The Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[15] R. D. Patterson and J. Holdsworth, “A Functional Model of Neural Activity Patterns and Auditory Images,” *Advances in Speech, Hearing and Language Processing*, vol. 3, no. B, pp. 547–563, 1996.

[16] J. D. Hoyt and H. Wechsler, “Detection of human speech using hybrid recognition models,” *Pattern Recognition, 1994. Vol. 2 - Conference B: Computer Vision &amp; Image Processing., Proceedings of the 12th IAPR International. Conference on*, vol. 2, pp. 330–333, 1994.

[17] J. Stegmann, G. Schroder, D. T. Berkom, D. T. Ag, and A. Kavalleriesand, “Robust Voice-Activity Detection Based on the Wavelet Transform,” *Speech Coding For Telecommunications Proceeding, 1997, 1997 IEEE Workshop on*, pp. 99–100, 1997.

[18] R. Gandhiraj and P. S. Sathidevi, “Auditory-Based Wavelet Packet Filterbank for Speech Recognition Using Neural Network,” *Advanced Computing and Communications, 2007. ADCOM 2007. International Conference on*, pp. 666–673, Dec. 2007.

[19] B. Atal and L. Rabiner, “A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 24, no. 3, pp. 201–212, Jun. 1976.

[20] S. Kajita and F. Itakura, “Subband-Autocorrelation analysis and its application for speech recognition,” *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, vol. ii, p. II/193–II/196, 1994.

[21] J. D. Hoyt and H. Wechsler, “Detection of human speech in structured noise,” *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, vol. 2, pp. 237–240, 1994.

[22] D. Kim, A. Member, S. Lee, and R. M. Kil, “Auditory Processing of Speech Signals for Robust Speech Recognition in Real-World Noisy Environments,” *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 1, pp. 55–69, 1999.

[23] O. Cheng, W. Abdulla, Z. Salcic, and N. Zealand, “Performance Evaluation of Front-End Algorithms for Robust Speech Recognition,” *Signal Processing and Its Applications, 2005. Proceedings of the Eighth International Symposium on*, vol. 2, pp. 711–714, 2005.

[24] A. Benyassine, E. Shlomot, H. Su, D. Massaloux, C. Lamblin, and J.-P. P. Petit, “ITU-T Recommendation G.729 Annex B: a silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications,” *Communications Magazine, IEEE*, vol. 35, no. 9, pp. 64–73, 1997.

[25] P. Kabal and R. P. Ramachandran, “The computation of line spectral frequencies using Chebyshev polynomials,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, no. 6, pp. 1419–1426, Dec. 1986.

[26] J. Ramírez, J. M. Górriz, and J. C. Segura, “Voice Activity Detection. Fundamentals and Speech Recognition System Robustness,” *Robust Speech Recognition and Understanding*, pp. 1–22, 2007.

[27] F. Beritelli, S. Casale, and G. Ruggeri, “Performance Evaluation and Comparison of ITU-T/ETSI Voice Activity Detectors,” *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP ’01). 2001 IEEE International Conference on*, vol. 3, pp. 1425–1428, 2001.

[28] ETSI, “Voice Activity Detector (VAD) for Adaptive Multi-Rate (AMR) Speech Traffic Channels,” 1999.

[29] ETSI, “Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms,” *ETSI ES 202*, vol. 050, pp. 1–45, 2007.

[30] V. Vapnik, *Statistical Learning Theory*. New York: John Wiley and Sons, Inc., 1998.

[31] L. Lu, S. Z. Li, and H. Zhang, “Content-Based Audio Segmentation Using Support Vector Machines,” *Proc. ICME*, vol. 1, pp. 749–752, 2001.

[32] J. Ramírez and P. Yelamos, “SVM-based speech endpoint detection using contextual speech features,” *Electronics Letters*, vol. 42, no. 7, pp. 426–428, 2006.

[33] J. Ramírez, “SVM-enabled Voice Activity Detection,” *Advances in Neural Networks - ISNN 2006*, pp. 676–681, 2006.

[34] M. Baig, S. Masud, and M. Awais, “Support Vector Machine based Voice Activity Detection,” *Intelligent Signal Processing and Communications, 2006. ISPACS ’06. International Symposium on*, pp. 319–322, 2006.

[35] S.-H. Chen, R. C. Guido, and S.-H. Chen, “Voice Activity Detection in Car Environment Using Support Vector Machine and Wavelet Transform,” *Multimedia Workshops, 2007. ISMW ’07. Ninth IEEE International Symposium on*, pp. 252–255, Dec. 2007.

[36] S.-H. Chen, S.-H. Chen, and B. R. Chang, “A Support Vector Machine Based Voice Activity Detection Algorithm for AMR-WB Speech Codec System,” *Innovative Computing, Information and Control, 2007. ICICIC ’07. Second International Conference on*, pp. 64–67, Sep. 2007.

[37] S.-H. Chen, R. C. Guido, T.-K. Truong, and Y. Chang, “Improved voice activity detection algorithm using wavelet and support vector machine,” *Computer Speech & Language*, vol. 24, no. 3, pp. 531–543, Jul. 2010.

[38] T. Kinnunen, E. Chernenko, M. Tuononen, P. Fränti, and H. Li, “Voice Activity Detection Using MFCC Features and Support Vector Machine,” *Int. Conf. on Speech and Computer (SPECOM07)*, vol. 2, pp. 556–561, 2007.

[39] Q. Jo, Y. Park, K. Lee, and J. Chang, “A Support Vector Machine-Based Voice Activity Detection Employing Effective Feature Vectors,” *IEICE Trans. Commun.*, vol. E91-B, no. 6, pp. 2090–2093, 2008.

[40] Q.-H. Jo, J.-H. Chang, J. W. Shin, and N. S. Kim, “Statistical model-based voice activity detection using support vector machine,” *Signal Processing, IET*, vol. 3, no. 3, pp. 205–210, 2009.

[41] D. Tax and R. Duin, “Support Vector Domain Description,” *Pattern Recognition Letters*, vol. 20, no. 11, pp. 1191–1199, Nov. 1999.

[42] B. Scholkopf, J. C. Platt, J. Shawe-taylor, A. J. Smola, and R. C. Williamson, “Estimating the Support of a High-Dimensional Distribution,” *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[43] X. Dong, W. Zhaohui, and Z. Wanfeng, “Support Vector Domain Description for Speaker Recognition,” *Neural Networks for Signal Processing XI, 2001. Proceedings of the 2001 IEEE Signal Processing Society Workshop*, pp. 481–488, 2001.

[44] Y. Zhou, Y. Gong, J. Wang, J. Sun, X. Zhang, and T. Zhu, “Speaker verification based on SVDD,” *Image and Signal Processing (CISP), 2010 3rd International Congress on*, vol. 7, pp. 3168–3172, Oct. 2010.

[45] Y. Zhou, X. Zhang, J. Wang, Y. Gong, and Y. Zhou, “Speaker recognition based on the combination of GMM and SVDD,” *Przegląd Elektrotechniczny*, vol. 87, no. 3, pp. 329–332, 2011.

[46] S. Omid Sadjadi, S. M. Ahadi, and O. Hazrati, “Unsupervised Speech/Music Classification Using One-Class Support Vector Machines,” *Information, Communications & Signal Processing, 2007 6th International Conference on*, pp. 1–5, 2007.

[47] A. Rabaoui, M. Davy, S. Rossignol, Z. Lachiri, and N. Ellouze, “Improved One-Class SVM Classifier for Sounds Classification,” *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pp. 117–122, 2007.

[48] A. Rabaoui, M. Davy, S. Rossignol, and N. Ellouze, “Using One-Class SVMs and Wavelets for Audio Surveillance,” *Information Forensics and Security, IEEE Transactions on*, vol. 3, no. 4, pp. 763–775, Dec. 2008.

[49] H. Kadri, M. Davy, A. Rabaoui, Z. Lachiri, N. Ellouze, and C. Universitaire, “Robust Audio Speaker Segmentation Using One Class SVMs,” *Proceedings of the EURASIP EUSIPCO’08*, pp. 1–5, 2008.

[50] M. Campus and F.- Metz, “Single-speaker/multi-speaker co-channel speech classification,” *Proc INTERSPEECH*, pp. 2322–2325, 2010.

[51] S. V. Vaseghi, “Wiener Filters,” in *Advanced Digital Signal Processing and Noise Reduction*, vol. 9, John Wiley & Sons, Ltd., 2000, pp. 178–204.

[52] R. Schl, I. Bezrukov, H. Wagner, and H. Ney, “Gammatone Features and Feature Combination for Large Vocabulary Speech Recognition,” *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4, pp. 649–652, 2007.

[53] W. H. Abdulla, “Auditory Based Feature Vectors for Speech Recognition Systems,” *Advances in Communications and Software Technologies*, pp. 231–236, 2002.

[54] X. Valero, S. Member, and F. Alías, “Gammatone Cepstral Coefficients: Biologically Inspired Features for Non-Speech Audio Classification,” *Multimedia, IEEE Transactions on*, vol. 14, no. 6, pp. 1684–1689, 2012.

[55] M. Slaney, “An Efficient Implementation of the Auditory Filter Bank,” *Apple Computer, Perception Group, Tech. Rep*, 1993.

[56] B. R. Glasberg and B. C. Moore, “Derivation of auditory filter shapes from notched-noise data,” *Hearing research*, vol. 47, no. 1, pp. 103–138, Aug. 1990.

[57] D. Schofield, “Visualisations of speech based on the model of the peripheral auditory system,” *NASA STI/Recon Technical Report*, vol. 86, 1985.

[58] J. Holdsworth, I. Nimmo-smith, R. Patterson, and P. Rice, “Implementing a GammaTone Filter Bank,” *Annex C of the SVOS Final Report: Part A: The Auditory Filterbank*, 1988.

[59] V. Hohmann, “Frequency analysis and synthesis using a Gammatone filterbank,” *Acta Acustica United With Acustica*, vol. 88, no. 3, pp. 433–442, 2002.

[60] V. Hohmann and T. Herzke, “Improved Numerical Methods for Gammatone Filterbank Analysis and Synthesis,” *Acta Acustica United With Acustica*, vol. 93, no. 3, pp. 498–500, 2007.

[61] N. Ma, P. Green, J. Barker, and A. Coy, “Exploiting correlogram structure for robust speech recognition with multiple speech sources,” *Speech Communication*, vol. 49, no. 12, pp. 874–891, Dec. 2007.

[62] D. Ellis, “Gammatone-like spectrograms,” 2009. [Online]. Available: http://www.ee.columbia.edu/~dpwe/resources/matlab/.

[63] M. Nutzinger, “Real-time Attacks on Audio Steganography,” *Information Hiding and Multimedia Signal Processing*, vol. 3, no. 1, pp. 47–65, 2012.

[64] H. Ito, “A local Wiener attack for additive watermarks,” *Consumer Electronics, 2009. ISCE ’09. IEEE 13th International Symposium on*, pp. 507–510, 2009.

[65] Technion Electrical Engineering Dept., “MATLAB Audio Database Toolbox User Manual,” no. July, pp. 1–12, 2008.

[66] D. Tax and R. Duin, “Support Vector Data Description,” *Machine Learning*, vol. 54, no. 1, pp. 45–66, 2004.

[67] T. Irino, R. D. Patterson, and I. Method, “A time-domain, level-dependent auditory filter: The gammachirp,” *The Journal of the Acoustical Society of America*, vol. 101, no. 1, pp. 412–419, 1997.

[68] O. Gauci, S. Member, C. J. Debono, S. Member, and P. Micallef, “A Nonlinear Feature Extraction Method for Phoneme Recognition,” *Electrotechnical Conference, 2008. MELECON 2008. The 14th IEEE Mediterranean*, pp. 811–815, 2008.