Table of Contents

1 Introduction 1

2 Literature Review 3

2.1 Speech Processing 3

2.1.1 Speaker dependence 3

2.1.2 Speech type 3

2.1.3 Recognition type 3

2.2 Pre-Processing 5

2.3 Feature Extraction 5

2.3.1 Mel-Frequency Cepstral Coefficients 5

2.3.2 Linear Predictive Coding 6

2.4 Classification Techniques 8

2.4.1 Dynamic Time Warping 8

2.4.2 Artificial Neural Networks 10

2.4.3 Hidden Markov Models 11

2.5 Auditory data 12

2.6 Robust speech recognition 12

2.6.1 Non-Negative Matrix Factorization 12

2.6.2 Discrete Wavelet Packet Transforms 13

3 Methodology 14

3.1 Justifications 14

3.1.1 System Specifications 14

3.1.2 Test Data 14

3.1.3 Classifier Choice 14

3.1.4 Feature Extractor Choice 15

3.2 Work to be done 15

3.2.1 Benchmarking 15

3.2.2 Combinatorial Extraction 16

3.2.3 Extension research 16

4 Project Management Plan 17

5 References 18

# Introduction

Research into the field of speech recognition began in the 1940’s when Bell Laboratories developed a basic speech recognition device. Further progress was made in the early 1950’s when Bell Labs developed a system that was differentiate between different spoken numerical digits [1]. Since then continuous progress has been made in the speech recognition field.

In recent years technology has become ubiquitous throughout modern society and as a result the need for automated speech recognition (ASR) systems has skyrocketed due to many fields currently employing them, such as military, telephony, and medicine.

With the rise of ASR systems one major problem has been encountered: Systems are unable to perform to a reliable standard when the conditions of the speech are outside those in which the system was trained.

As a result one of the prominent fields of research has been robust speech recognition, which attempts to combat the unreliability inherent with modern day ASR systems. This research has been focused into three main categories: pre-processing speech signals to normalize and de-noise them, increasing the performance of existing recognition methods [2, 3]; improved classifier construction, endeavoring to build machine learning paradigms that compensate for noise induced signal variance [4, 5]; and improved feature extraction methods, which attempts to define noise immune features of signals such that additive noise causes little to no variance in the resultant feature vector [6-9].

This thesis aims to establish whether a combination of existing feature extraction techniques can be employed to yield greater results than each one individually, and determine whether it is possible to combine the strengths of each into a single extraction method. As such, the following research questions will be addressed:

1. Using a combination of feature extraction methods is it possible to achieve a higher phonetic recognition rate than using each extractor individually?
2. Using noise independent features is it possible to increase phonetic recognition rates of speech systems?
3. Is there a correlation between feature vector variance and recognition rates?

The scope of this thesis will be limited to the research, implementation, and optimization of feature extraction algorithms and does not involve the construction or implementation of a complete ASR system.

# Literature Review

## Speech Processing

Automated speech recognition (ASR) systems can be partitioned into numerous categories based on the various criteria they meet. The criteria for systems usually consist of: speaker dependence, speech type, and recognition type.

### Speaker dependence

Speaker dependent systems are trained for use by a single speaker, building classification models based on the speaker’s unique acoustic-phonetic model. In contrast, speaker independent systems are designed for use by numerous speakers, including those who were not involved in the system training process.

Speaker independence is hard to achieve due to the feature parameterization becoming tuned to the training speaker(s), causing a speaker-specific bias in the classification [4]. Error rates for speaker independent systems tend to be 3 to 5 times larger than speaker dependent systems [10].

### Speech type

Speech can be broken into three distinct types: isolated, discontinuous, and continuous [4]. Isolated speech consists of singular words and is often considered word recognition. Discontinuous speech involves the speaker being purposefully articulate and inserting artificial pauses between consecutive words. In continuous speech (natural speech) the speaker makes no effort to alter their speech patterns.

Recognition for isolated and discontinuous speech is simpler due to the clearly defined word boundaries and distinct pronunciation. Continuous speech is much harder to process due to the undefined word boundaries in addition to corrupted pronunciation introduced by co-articulation, the slurring of speech sounds, which can cause phrases like “could you” to sound like “could jou” [4]. During a standard evaluation, isolated and continuous speech achieved error rates of 3.1% and 8.7% respectively [11].

### Recognition type

There are two main classifications for ASR systems: word recognition and phonetic recognition. Word recognition is desirable due to the human interpretable aspect of the transcription, that is, all results are complete words. However, the main disadvantage of word based ASR is that the error rates of the system are proportional to the vocabulary size. For small sets of words it is possible to obtain <1% error rates [12], whereas vocabulary sizes of 200, 5000, or 100000 could have approximate error rates of 3%, 7%, or 45% respectively [13-15]. Additionally, the processing time for a word-based system is also proportional to the vocabulary size, making it impractical for most real-world applications.

Conversely, phonetic recognition is able to avoid both error rate and processing time inflation by breaking words down into their base components, phonemes. A phoneme can be defined as “the smallest contrastive linguistic unit which may bring about a change of meaning [of a word]” [16], which results in a classification base of approximately 42 phonemes for the English language (see Table 2‑1), with the drawback being that the transcription isn’t directly human readable (see Table 2‑2).

Table ‑: List of English Phonemes and Manners of Articulation [17]

|  |  |  |  |
| --- | --- | --- | --- |
| **Phoneme** | **Manner of Articulation** | **Phoneme** | **Manner of Articulation** |
| iy | vowel | l | liquid |
| ih | vowel | r | liquid |
| ia | vowel | m | nasal |
| ey | vowel | n | nasal |
| eh | vowel | ng | nasal |
| ae | vowel | f | fricative |
| ea | vowel | v | fricative |
| aa | vowel | th | fricative |
| ao | vowel | dh | fricative |
| ow | vowel | s | fricative |
| uh | vowel | z | fricative |
| uw | vowel | sh | fricative |
| ua | vowel | zh | fricative |
| ah | vowel | hh | fricative |
| er | vowel | p | stop |
| ax | vowel | b | stop |
| ay | diphthong | t | stop |
| oy | diphthong | d | stop |
| oh | diphthong | k | stop |
| aw | diphthong | g | stop |
| y | glide | ch | affricate |
| w | glide | jh | affricate |

Table ‑: Example of the Phonetic Decomposition of Various Words/Phrases [17]

|  |  |
| --- | --- |
| **Word/Phrase** | **Phonetic Decomposition** |
| call | k ao l |
| dial | d ay ax l |
| seven | s eh v n |
| recognise speech | r eh k ao g n ay z s p iy ch |
| wreck a nice beach | r eh k ay n ay s b iy ch |

## Pre-Processing

Pre-processing is a means of cleaning and normalizing a speech signal to allow for easier and more reliable feature extraction, as most speaker independent ASR systems perform very poorly when tested in environments different from the one in which they were trained [2]. Two major factors that contribute to signal distortion are additive noise and signal convolution with an unknown linear system [18]. As the primary goal of pre-processing techniques is to reduce the total noise in a signal, none will be considered in this thesis.

## Feature Extraction

Feature extraction is the process of parsing an input signal and parameterizing it into a vector, known as a feature vector, which contains the important characteristics of the signal as deemed by the chosen parameterization process. It is important that any feature vectors produced contain enough information to accurately distinguish between all possible outcomes of a system, while attempting to limit the total number of features to a computationally efficient range. The two most important sections of speech characteristics are those contained in the spectral envelope (vocal tract characteristics) and those contained in the supra-segmental features (voice source characteristics) [19].

Most common ASR extraction approaches use cepstral analysis, the power cepstrum in particular, as the cepstral domain contains more information than the spectral domain, which is more commonly used for signal analysis.

The pitch and formant information contained within the signal are additive in the cepstral domain, making them easily separable [20].

### Mel-Frequency Cepstral Coefficients

The most commonly used cepstral analysis technique is known as Mel-Frequency Cepstral Coefficients (MFCC). The MFCCs collectively represent the Mel-Frequency Cepstrum (MFC), which is the linear cosine transform of the log power spectrum but scaled to the non-linear Mel Scale.

The Mel Scale was proposed in 1937 and was determined by listeners as the tone steps approximately ‘half as high’ in pitch compared to the reference tones [21]. This experiment helped conclude that the human auditory model was not linear, and was in fact approximately logarithmic. As a result, the Mel Scale is often used in speech processing as it more closely approximates the human auditory model than the linearly spaced Hertz scale.

The equation for the conversion of Hertz to Mels is given by [22]:

|  |  |
| --- | --- |
|  | (‑) |

The process for extracting MFCCs from an input signal is similar to most other cepstrum based methods, with the main difference being the incorporation of the Mel Frequency Filter Bank, as shown in Figure 2‑1.

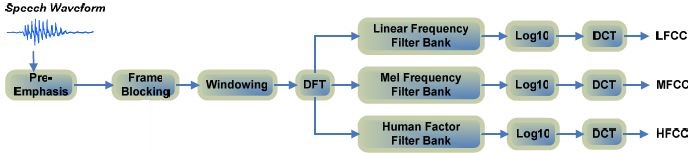


Figure ‑: Example of Various Cepstrum based Extractors [8]

### Linear Predictive Coding

Linear Predictive Coding (LPC) is a method of feature extraction that attempts to predict future values of an input signal based on the past values of that signal [23]. The feature vectors represent the coefficients of a linear filter that would reproduce the signal. LPC can be broken into two distinct segments, analysis/encoding and synthesis/decoding. During the encoding stage the speech signal is broken into blocks or frames, which are then processed to determine the corresponding filter coefficients that would be capable of reproducing that frame of speech. The decoding stage involves rebuilding the speech signal from the received filter coefficients. For the purposes of this thesis, only the encoding stage is considered, as speech reconstruction is not necessary for classification.

The encoding phase of LPC feature extraction involves two main steps: determining whether a block is voiced or unvoiced and pitch period estimation.

In order to separate voiced and unvoiced speech components, the assumption is made that voiced sounds have a high average energy (thus large amplitudes) as well as having distinct formant frequencies (see Figure 2‑2), while unvoiced sounds have more “random” waveforms of lower amplitude but higher frequency (see Figure 2‑3). Due to their higher frequencies, unvoiced speech crosses the x-axis more times than voiced speech, providing another deterministic feature.

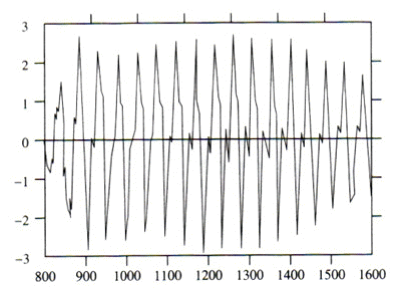


Figure ‑: Voiced sound - Letter 'e' in the word 'test' [23]

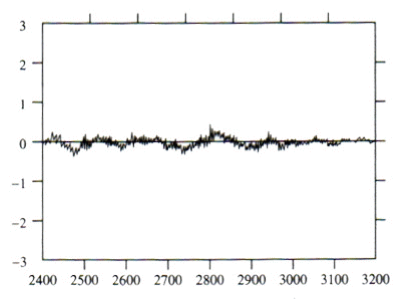


Figure ‑: Unvoiced sound - Letter 's' in the word 'test' [23]

Neighbouring segments are also taken into consideration when determining the vocal nature of the current segment, given the likely correlations due to the nature of human speech, e.g. it’s improbable to have unvoiced segments contained in a group of voiced segments and vice versa.

Pitch period estimation is used to determine the period of the vocal cord vibration required to produce the speech segment, however this only applies to voiced speech segments as the unvoiced segments are not produced by vocal cord vibrations. While there are numerous algorithms to estimate pitch period, they are all considered computationally expensive.

LPC isn’t a very robust feature extraction method as its features are based on assumptions that are easily interfered with by additive noise. Background noise can increase the average energy of an unvoiced segment enough to misclassify, as much as it can increase the frequency of a voiced segment.

## Classification Techniques

Classification is the final stage of the speech recognition process, taking the calculated feature vectors and determining their corresponding word/phoneme. However, the choice of classifier is much less important that the quality of the feature extraction method chosen, as an unreliable input will cause the classifier to operate much more poorly than expected [24].

The most common classification techniques can be broken into three main categories: Template-based approaches, Knowledge-based approaches, and Statistical-based approaches [4].

Template-based approaches involve comparing input speech to pre-recorded word/phoneme templates. While this has the advantage of using ‘perfectly’ accurate word models, the templates are fixed, making it hard to account for any variance in the input speech. Dynamic Time Warping is an example of template-based classification.

Knowledge-based approaches involve hand-coding speech variances into a system based on obtained ‘expert’ knowledge. The downside to this approach is that it is often impractical to obtain and incorporate ‘expert’ knowledge into a system.

Statistical-based approaches attempt to model speech variations statistically through automatic learning procedures. The main disadvantage of statistical based systems is that they often require extensive training or come with pre-defined modeling assumptions, which can be inaccurate. Hidden Markov Models are an example of statistical-based classification.

### Dynamic Time Warping

Dynamic Time Warping (DTW) is a classification technique that has fallen out of favor since the rise of statistical classifiers like Hidden Markov Models.

The DTW classification method uses template patterns of known results and attempts to non-linearly warp the input signal to match these templates, as shown in Figure 2‑4.



Figure ‑: Example of Matching an Input (Y) to a Template (X) [25]

To achieve this the input signal and template are broken into a discrete number of frames and the optimal alignment path calculated. A *cost measure* is determined for each frame pair, with the combination of all frame pairs resulting in a *cost matrix*. The optimal alignment path for the cost matrix is considered to be shortest path from to the opposite corner of the matrix, as shown in Figure 2‑5. Each variant of DTW has different path constraints for the optimal alignment path, with the ‘classic’ being Monotonicity, i.e. the path must never travel backwards, and Step Size being limited to adjacent frames.



Figure ‑: Dynamic Time Warping. (a) Optimal Alignment Path. (b) Path Constraint Example [4]

Mathematically, the cumulative word score can be described as

|  |  |
| --- | --- |
|  | (‑) |

where is the Euclidean distance between frame of the input signal and frame of the reference template. Ultimately, the reference template with the lowest cumulative word score is considered to be the best match for the given input.

### Artificial Neural Networks

Artificial Neural Networks (ANN) attempt to simulate the neural networks within the human brain. ANNs consist of numerous simple processing elements (neurons) connected together through a weighted network of interconnects. Each neuron computes the non-linear weighted sum of its inputs and transmits the result along its outputs [4]. ANNs can usually be broken into layers: an input layer, zero or more ‘hidden’ layers, and an output layer. The input layer handles receipt of data and may perform initial computations, the hidden layer(s) perform any additional computations required by the system, and the output layer performs data aggregation and outputs the result from the system.

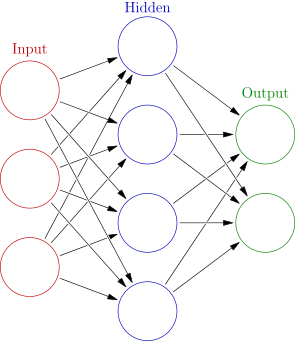


Figure ‑: Example of an Artificial Neural Network

The simplest ANNs consist solely of forward propagating neurons, each making calculations independent of past calculations, which is useful for continuous signals such as speech.

There are three main types of machine learning: supervised, semi-supervised, and unsupervised. In supervised and semi-supervised learning, the ANN is provided with input data and its performance monitored by a “teacher”. The results of the system are judged as explicitly correct/incorrect for supervised learning, allowing for an influenced solution, and good/bad for semi-supervised learning, which aims for the system to determine an optimal solution. Unsupervised learning requires the system distinguish patterns within the data with no guidance and is usually applicable where little information is available about the data.

During supervised learning, the ANN is fed known sets of data and the weighting scheme for each neuron is adjusted to compensate for any incongruities between the calculated and expected results. An ANN is also capable of learning dynamically after the dedicated learning phase.

The biggest disadvantage of ANNs is the need for very large and varied training sets to build a variance tolerant classifier. However, given a robust training set, ANNs can be very adaptive to speech variance, making them very useful for speech processing.

### Hidden Markov Models

A Markov model is considered a probabilistic model that assumes the Markov property, meaning that the system is memory-less or that the future states of the system are independent of its past states. A Hidden Markov Model (HMM) is a form of Markov model in which the states are unobserved (hidden). A HMM can be expressed as a Finite State Machine where transitions are based on a probabilistic model.

HMMs consist of three fundamental components: an transition probability matrix , where represents the probability that a transition will occur from state at time to state at time for all states; an observation probability matrix , which contains observations where represents the probability of observation at time while being in state ; and a initial probability matrix , which defines the probability of the model starting at state and time . Thus the model can be defined in compact notation as [26, 27].



Figure ‑: Example Markov Generation Model [5]

There are three basic algorithms associated with HMMs: the *Forward algorithm*, used for isolated word recognition; the *Viterbi algorithm,* used for continuous speech recognition; and the *forward-backward algorithm,* used for HMM training [4].

For the purposes of speech recognition a left-right topology is implemented, which constrains the state transitions to the current node or nodes to the right, similar to a forward propagating ANN. A single HMM is generated for each phoneme per extraction method, with each model being trained via the aforementioned *forward-backward algorithm*. The *Viterbi algorithm* is used to determine the sequence path which would produce the final observed state sequence [5].

## Auditory data

## Robust speech recognition

### Non-Negative Matrix Factorization

Non-negative Matrix Factorization (NMF) has been successfully used in multiple areas of speech recognition, including noise reduction and speaker separation [6, 7]. Basic NMF approaches assume a linear signal model, i.e. the signal is a combination of several distinct linear spectral components. Conventionally NMF has been used as a preprocessing step for speech recognition, with recent research into their use for robust feature extraction.

The basic principle of NMF is that a speech signal can be represented as the spectra of events occurring within the signal and the time-varying gain of those events. Mathematically, given an matrix , there exists two matrices, and **,** such that

|  |  |
| --- | --- |
|  | (‑) |

where and represent the spectral events and time-varying gain respectively.

For a continuous speech recognition task it is possible to predefine as the spectra of all phonemes, which is kept constant throughout the process while is updated iteratively, saving computation time.

As NMF is performed on a short-time magnitude spectrum of the signal, additive noise has little overall effect on the gain of the underlying speech spectra, as shown in Figure 2‑8.



Figure ‑: NMF gains of the Spoken Letter Sequence “u r o”. Clean (top) and Noisy (bottom) [6]

### Discrete Wavelet Packet Transforms

# Methodology

The following section outlines the suggested implementations and methodologies for developing a robust ASR system, including justification for the choices made regarding various aspects of the system.

## Justifications

### System Specifications

As speaker dependent systems offer a limited usage scope, as well as requiring extensive training for each speaker, a speaker independent system was opted for. Continuous speech was chosen due to the larger number of feature extraction techniques already researched in addition to it being a real-world representation of speech signals. Phoneme recognition was chosen due to its significantly smaller model training time, which is a major factor given the large number of extraction methods needed. Phonemes also contain definitive structural characteristics, providing a smaller comparative set than whole word recognition, helping limit misclassification introduced by speaker variance.

### Test Data

Due to the requirement of producing a robust ASR system, the Wall Street Journal Cambridge Read News (WSJCAM0) corpus was the chosen speech repository. It contains a very broad speaker base in relation to gender and age, allowing for superior speaker independence, as well as providing a very large number of continuous training/testing sentences. Budget and accessibility were also applicable factors, making the WSJCAM0 a superior option due to its immediate availability as the electrical engineering department already had a copy.

### Classifier Choice

A Hidden Markov Model classification scheme was chosen for baseline classification due to the readily available open source software (HTK), the software’s ability to provide statistical feedback about the extractor misclassification, and the extensive documentation available for the software. HMMs are also a superior to ANNs due to their continuous probabilistic nature, which more closely models a continuous speech signal than a similar ANN.

An Artificial Neural Network is the choice of classifier for the combinatorial machine learning stage of the implementation as it allows for an easily adaptive learning process, due ANNs being able to perform dynamic learning more effectively than a HMM.

### Feature Extractor Choice

The three primary feature extraction methods chosen perform in different signal domains and as such provide a much larger feature space that contains minimal feature overlap. For extraction methods in which the signal domains are the same, the analysis technique differs to a large enough degree that the feature overlap remains minimal.

Mel-Frequency Cepstral Coefficients are the chosen baseline feature extraction method as they are widely considered to be the standard by which other extraction methods are measured. MFCCs are also supported by the HTK, reducing development time required for the benchmarking stage of the system.

Non-negative Matrix Factorization has been successfully employed to extract speech features from various speech signals with minimal noise induced variance.

Discrete wavelet transforms allow the

* Non-Negative Matrix
* Discrete wavelet

## Work to be done

The numerical laboratory software MATLAB® will be used for the generation of additive signal noise, the filtering of the signals, and the feature extraction processes (excluding the MFCC extraction). MATLAB will also be used to perform the ANN learning for the final stages of classification. The Hidden Markov Model Toolkit (HTK) will be used to perform the baseline machine learning and classification, the generation of the MFCC feature vectors, and the generation of confusion matrices for each of the extraction methods.

The use of BASH scripts will be employed to enable the automation of most of these processes.

### Benchmarking

A baseline result will be generated using an MFCC extraction process, as it is widely recognized as the standard feature space for speech recognition systems, allowing for a comparative point with other research. The extraction process will be done on signals over a range of Signal-to-Noise ratios (SNR) to provide the guideline performance measure.

Each of the chosen feature extractors will also be benchmarked at the same SNRs, to provide their independent classification metrics.

### Combinatorial Extraction

During the preliminary benchmarking process, the HTK will generate confusion matrices showing the statistical breakdown of the phoneme misclassification for each extraction method. These confusion matrices will be used to determine the strengths and weaknesses of each extraction method, as well as which extraction methods excel in each classification area. Using ANN learning, a weighted classification scheme will be generated in an attempt to use the strengths of each extraction method to provide a more robust overall classification result.

### Extension research

Performing numerous feature extractions of a single signal in order to achieve combinatorial classification is both computationally expensive and time consuming. If preliminary results are positive, an attempt will be made to compile a signal extraction method based on a combination of the best extraction methods for each classification area. This will be achieved using the confusion matrices generated for each extraction method to determine area based effectiveness, and employing the underlying knowledge of how the extraction is performed to determine *why* that extractor is effective and how to utilize it.

# Project Management Plan

# References

[1] K. Davis, R. Biddulph, and S. Balashek, "Automatic recognition of spoken digits," *The Journal of the Acoustical Society of America,* vol. 24, pp. 637-642, 2005.

[2] R. M. Stern, F.-H. Liu, Y. Ohshima, T. M. Sullivan, and A. Acero, "Multiple approaches to robust speech recognition," in *Proceedings of the workshop on Speech and Natural Language*, 1992, pp. 274-279.

[3] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 27, pp. 113-120, 1979.

[4] J. Tebelskis, "Speech recognition using neural networks," Carnegie Mellon University, 1995.

[5] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason*, et al.*, *The HTK book* vol. 3.2: Entropic Cambridge Research Laboratory Cambridge, 1997.

[6] B. Schuller, F. Weninger, M. Wollmer, Y. Sun, and G. Rigoll, "Non-negative matrix factorization as noise-robust feature extractor for speech recognition," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 2010, pp. 4562-4565.

[7] M. Schmidt and R. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," 2006.

[8] I. Mporas, T. Ganchev, M. Siafarikas, and N. Fakotakis, "Comparison of Speech Features on the Speech Recognition Task," *Journal of Computer Science,* vol. 3, pp. 608-616, 2007.

[9] B. Kotnik and Z. Kačič, "A noise robust feature extraction algorithm using joint wavelet packet subband decomposition and AR modeling of speech signals," *Signal Processing,* vol. 87, pp. 1202-1223, 6// 2007.

[10] K.-F. Lee, "On large-vocabulary speaker-independent continuous speech recognition," *Speech Communication,* vol. 7, pp. 375-379, June 21 1988.

[11] L. Bahl, R. Bakis, P. Cohen, A. Cole, F. Jelinek, B. Lewis*, et al.*, "Speech recognition of a natural text read as isolated words," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '81.*, 1981, pp. 1168-1171.

[12] G. R. Doddington, "Phonetically sensitive discriminants for improved speech recognition," in *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, 1989, pp. 556-559.

[13] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 23, pp. 67-72, 1975.

[14] M. Miyatake, H. Sawai, Y. Minami, and K. Shikano, "Integrated training for spotting japanese phonemes using large phonemic time-delay neural networks," in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, 1990, pp. 449-452.

[15] S. Kimura, "100000-word recognition using acoustic-segment networks," in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, 1990, pp. 61-64.

[16] A. Cruttenden, "Gimson's pronunciation of English," 7th ed: Routledge, 2013, p. 41.

[17] A. Bath, "Robust Speech Recognition via Human Auditory Modeling," Bachelor of Engineering (Electrical) / Bachelor of Science (Computer Science), School of Engineering and Physical Sciences, James Cook University, 2010.

[18] A. Acero and R. Stern, "Environmental robustness in automatic speech recognition," in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, 1990, pp. 849-852 vol.2.

[19] S. Furui, "Recent advances in speaker recognition," in *Audio- and Video-based Biometric Person Authentication*. vol. 1206, J. Bigün, G. Chollet, and G. Borgefors, Eds., 1st ed: Springer Berlin Heidelberg, 1997, pp. 235-252.

[20] A. M. Noll, "Cepstrum Pitch Determination," *The Journal of the Acoustical Society of America,* vol. 41, pp. 293-309, 1967.

[21] S. S. Stevens, J. Volkmann, and E. B. Newman, "A Scale for the Measurement of the Psychological Magnitude Pitch," *The Journal of the Acoustical Society of America,* vol. 8, pp. 185-190, 1937.

[22] D. O'shaughnessy, *Speech communication: human and machine*: Universities press, 1987.

[23] J. Bradbury, "Linear predictive coding," *Mc G. Hill,* 2000.

[24] E. Avci, "A new optimum feature extraction and classification method for speaker recognition: GWPNN," *Expert Systems with Applications,* vol. 32, pp. 485-498, 2// 2007.

[25] M. Müller, "Dynamic time warping," *Information retrieval for music and motion,* pp. 69-84, 2007.

[26] L. Rabiner and B.-H. Juang, "An introduction to hidden Markov models," *ASSP Magazine, IEEE,* vol. 3, pp. 4-16, 1986.

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