Image processing methods to segment speech spectrograms for word level recognition

Declaration

I hereby declare that this thesis has not been and will not be, submitted in whole or in part to another University for the award of any other degree.

Signature

Mohammed Al-Darkazali

Dated: 7th April 2017

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisors Prof Chris Chatwin and Dr Rupert Young, at the University of Sussex for their incessant support, encouragement and guidance throughout my PhD study. I owe this thesis to their insightful comments and hard questions which incited me to widen my research from various perspectives.

I would like to mention my special thanks to Dr Rupert Young, whose office doors were always open whenever I ran into confusion. He is also instrumental in editing my work and honing my language.

Besides them, I am grateful to my dear friend, Dr Abbas Ali for being there for me through thick and thin. His friendship has been a huge emotional support during my time in the UK.

I would also like to acknowledge my Sponsors at the University of Mustansiriya for funding my PhD in Sussex and giving me the opportunity to fulfil my dream.

This thesis would not have been possible without my brothers and sisters who have been my pillars of support during times of stress and burnout. I thank them for their patience and unwavering love.

Finally and most importantly, I would like to dedicate this work to my parents who were my first teachers. They inculcated the spirit of hard work, perseverance and honesty in me. I miss them both immensely and forever seek their blessings from Heaven.

Abstract

The ultimate goal of automatic speech recognition (ASR) research is to allow a computer to recognize speech in real-time, with full accuracy, independent of vocabulary size, noise, speaker characteristics or accent. Today, systems are trained to learn an individual speaker's voice and larger vocabularies statistically, but accuracy is not ideal. A small gap between actual speech and acoustic speech representation in the statistical mapping causes a failure to produce a match of the acoustic speech signals by Hidden Markov Model (HMM) methods and consequently leads to classification errors. Certainly, these errors in the low level recognition stage of ASR produce unavoidable errors at the higher levels. Therefore, it seems that ASR additional research ideas to be incorporated within current speech recognition systems. This study seeks new perspective on speech recognition. It incorporates a new approach for speech recognition, supporting it with wider previous research, validating it with a lexicon of 533 words and integrating it with a current speech recognition method to overcome the existing limitations. The study focusses on applying image processing to speech spectrogram images (SSI). We, thus develop a new writing system, which we call the Speech-Image Recogniser Code (SIR-CODE). The SIR-CODE refers to the transposition of the speech signal to an artificial domain (the SSI) that allows the classification of the speech signal into segments. The SIR-CODE allows the matching of all speech features (formants, power spectrum, duration, cues of articulation places, etc.) in one process. This was made possible by adding a Realization Layer (RL) on top of the traditional speech recognition layer (based on HMM) to check all sequential phones of a word in single step matching process. The study shows that the method gives better recognition results than HMMs alone, leading to accurate and reliable ASR in noisy environments. Therefore, the addition of the RL for SSI matching is a highly promising solution to compensate for the failure of HMMs in low level recognition. In addition, the same concept of employing SSIs can be used for whole sentences to reduce classification errors in HMM based high level recognition. The SIR-CODE bridges the gap between theory and practice of phoneme recognition by matching the SSI patterns at the word level. Thus, it can be adapted for dynamic time warping on the SIR-CODE segments, which can help to achieve ASR, based on SSI matching alone.

Table of Contents

List of Publications

The publications resulting from this thesis are presented:

1. Al-Darkazali M., Young R., Chatwin C., Defining properties of speech spectrogram images to allow effective pre-processing prior to pattern recognition. SPIE 8748, Optical Pattern Recognition XXIV, 11 pages, published in SPIE Proceedings, Vol. 8748, 29/04 (2013).

2. Al-Darkazali M., Young, Rupert, Chatwin, Chris and Birch, Philip (2016) Integration of phoneme pattern recognition with hidden Markov models to enhance performance of low level speech recognition. Asian Journal of Physics, 25 (6). ISSN 0971-3093

List of Acronyms

A	Represents a word consisting of patterns of SSIFS pattern
AB	Represents a word consisting of patterns of SSIFS and SSIRS respectively
ABA	Represents a word consisting of patterns SSIFS, SSIRS and SSIFS, respectively
ABAB	Is represented a word consist of patterns SSIFS, SSIRS, SSIFS and SSIRS respectively
ABM	Area based matching
AFP	Absent Full Formant Pattern
ASR	Automatic Speech Recognition
B	Represents a word consisting of patterns of SSIRS pattern
BA	Represents a word consisting of patterns SSIRS and SSIFS, respectively
BAB	Represents a word consisting of patterns SSIRS, SSIFS and SSIRS, respectively
BABA	Represents a word consist of patterns SSIRS, SSIFS, SSIRS and SSIFS, respectively
MCE	Minimum classification error metric
DFT	Discrete Fourier Transform
FBM	Feature Based Matching
FD	Fourier descriptors
FFP	Full Formant Pattern
FFT	Fast Fourier Transform
FT	Fourier Transform
G	Gap Pattern
HMM	Hidden Markov Model
IF	Instantaneous Frequency

IPA	International Phonetic Alphabet
ISTFT	Inverse STFT
IWSR	Isolated Word Speech Recognition
LAFP	Long duration Absent Formats Pattern
LCP	Linear Code Prediction
MACH	Maximum Average Correlation Height
MCC	Maximum Cross Correlation
MFCC	Mel Frequency Cepstral Coefficient
NCC	Normalised Cross-Correlation
OT	Optimal -Trade-off
PLP	Perceptual Linear Predictive Coefficients
PR	Pattern Recognition
PSD	Power Spectral Density
TFR	Time Frequency Representation
QTFR	Quadratic TFR
RGAML	Right Guess is After the Maximum Likelihood
RL	Realization Layer
SAFP	Short length Absent Formant Pattern
SBSSI	Semi Binary SSI
SFFP	Semi- Full Formant Pattern
SIR-CODE	Speech-Image Recogniser Code
SSI	Speech Spectrogram Image
SSIFS	SSI Format Structures
SSIRS	SSI Random Structures
SSIPR	SSI Pattern Recognition
STFT	Short-Time Fourier Transform
WT	Wavelet Transforms

List of Figures

List of tables

List of Symbols

$\langle u\|v\rangle$	The inner product of u and v, where u and v are members of an inner product space.
arg	Function operating on complex numbers gives the angle between the positive real axis to the line joining the point to the origin.
u	Scale domain variable
θ	Frequency domain variable
τ	Time domain variable
ω	Spectra in frequency domain $\omega \in \theta$
t	The instant of a signal in the time domain, $t \in \tau$
S	A signal in frequency domain
s	A signal in time domain
$\mathrm{C}(\mathrm{t}, \omega)$	Spectrum of the signal (s)
$\emptyset(\theta, \tau)$	Function kernel
$A\ (\theta, \tau)$	Symmetrical ambiguity function
$*$	Complex conjugate
$\|\quad\|$	Euclidean norm
$P\left(O\|[\underline{w} = w]\right)$	Conditioning on a random variable, Let O be an event. The conditional probability of O, given W is defined as the random variable, whenever (W=w).

CHAPTER ONE

1 CHAPTER ONE

1.1 Introduction

The aim of speech recognition technology, in a general sense, is to build machines that can receive spoken information and act appropriately upon that information, and is part of the quest for “artificially intelligent” machines.

There are three categories of speech recognition systems with acceptable performance:

a. System with small vocabularies (~10- 100 words).
b. Systems in which words are purposely spoken in isolation from one another (vocabularies can exceed 10,000 words).
c. Those that accept continuous speech but are concerned with relatively constrained “task domains”, for example, messages likely to occur in office correspondence at a particular company (vocabularies typically ~1000-5000 words).

Most systems active in practical application are of the small-vocabulary or isolated-word type. Existing systems for more natural like human – machine communication remain primarily experimental. There is no system, even of those being used in practical applications, that is highly robust to environmental noise (office noise, factory noise, airport noise, etc.). All perform significantly better if required to recognize only a single speaker who trains the system. Even if the system is used to recognize multiple speakers, performance is generally improved if the users are also the trainers. Although some existing systems take advantage of the grammatical structure of the language, only experimental systems have more abstract cognitive abilities like discerning meaning or learning from mistakes [1].

The first speech recognition systems could understand only digits (given the complexity of human language, it makes sense that engineers first focused on numbers). The first paper presenting the idea of speech recognition was published in 1952 and described the Bell Labs spoken digit recognizer Audrey [2]. The system relied on measuring spectral resonances during the vowel region of each digit. In the 1960s several fundamental ideas, such as filter bank spectrum analysis, zero crossing analysis and time-normalization methods in speech recognition were published [3]. In the 1970s isolated word recognition became an advanced technology due to fundamental studies [4]; also pattern recognition, dynamic programming, and linear predictive coding (LPC) ideas were applied to speech recognition. Speech recognition systems were the made truly speaker independent [5]. In the 1980s a focus of research was the problem of connected word recognition. Speech research was shifted from template based approaches to statistical modelling methods, i.e. the hidden Markov model (HMM) approach and neural network methods [6]. In the 1990s the main focus of research was large vocabulary continuous speech recognition and robust speech recognition, which included syntax, semantics, and pragmatics into speech recognition higher level processing [1, 7]. Speech recognition systems have been developed for a wide variety of applications, ranging from small vocabulary word recognition to large vocabulary speech dictation.

1.2 Goal of automatic speech recognition (ASR) systems

A distinction is generally made in ASR between recognition of utterances from a speaker who has previously enrolled his voice (speaker dependent recognition) and a speaker whose voice the recogniser heard previously (speaker independent recognition).

Generally speaking, attempts at ASR fall into two categories: a knowledge-based approach, in which knowledge about the domains of linguistics and phonetics is used to construct a set of rules which is in turn used to interpret the acoustic input signal; and a pattern-matching approach in which a priori knowledge about speech is mainly ignored and techniques of pattern classification are applied to the input signal [8]. In particular, there are two levels to the recognition of speech which are summarised in Figure 1.

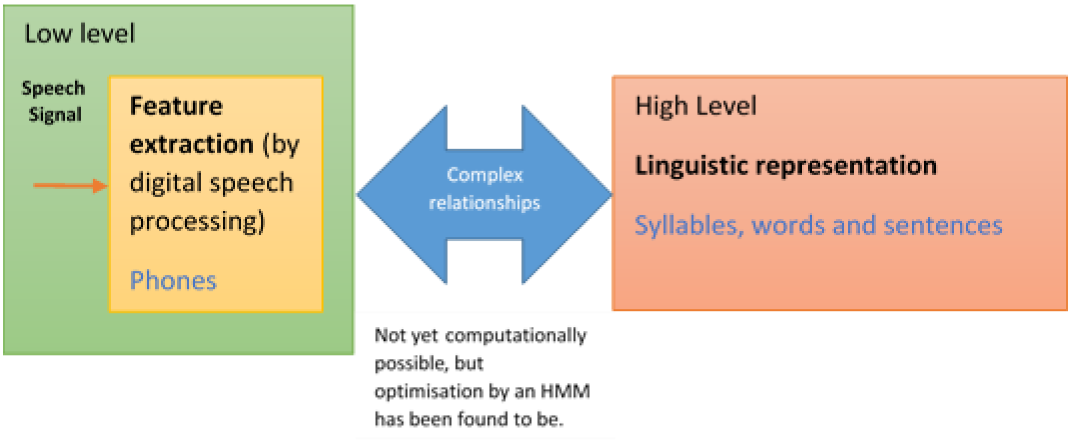

Low level
Speech Signal
Feature extraction (by digital speech processing)
Phones
Complex relationships
Not yet computationally possible, but optimisation by an HMM has been found to be.
High Level
Linguistic representation
Syllables, words and sentences

Figure 1 the recognition levels in ASR.

ASR systems are basically pattern classification systems [3]. Any utterance of speech is modelled as a sequence of sounds. These sounds are either of the phonemes in a language, words in that language, or larger units, depending on the vocabulary of the system and the task being achieved by it. The complete set of sounds that the ASR system has to recognize forms the classes modelled by it. The ASR system then orders segments of speech so as to place it into one of these classes.

Classification is not performed using the speech signal directly. Instead, the speech signal is parameterized into a sequence of feature vectors, or parameter vectors, and classification is performed using these feature vectors. The feature vectors used are usually cepstral coefficients [9] or variants of the cepstral [10] derived from the power spectral density (PSD) of short windowed segments, or frames of speech. Thus, a sequence of speech samples is transformed into a sequence of feature vectors each representing a single frame of speech, which is used to perform recognition.

1.3 Can the speech wave signal of words which are uttered by different persons form a unique spectrogram word pattern?

Spectrograms have demonstrated that human speech utterances can be analysed by expert spectrogram readers. Spectrogram reading requires a combination of different sources of knowledge such as articulatory movement, phonetics, linguistics and acoustic phonetics [11, 12]. The wave of separate utterances of the same word may be very different, but there are likely to be more similarities between spectrograms because they better illustrate the vocal tract resonances, which are closely related to the positions of the articulators [13]. The spectrograms will differ in detail due to the above differences, but different timescale variations will be particularly obvious.

Movement of the vocal tract can be well represented using a wideband spectrogram. The wideband spectrogram is generated using a relatively short time window that gives good time resolution but less specified frequency resolution. In general none of the those spectrogram word patterns will be matched perfectly, but in some sense of the word, the speech pattern of a correct word is likely to be a better match than a wrong word, because it is generated by more similar articulatory movements. Exploiting this similarity is, however, critically dependent on how the word patterns are compared. The slow change of the spectrogram word patterns, and the accompanying transitions within individual words produce the characteristic contours and shapes that are used to identify the sounds [14].

Pinkowski [15], treats the spectrogram objects as two-dimensional binary image objects, in which case shape or contour features such as Fourier descriptors (FD) are appropriate. For classifying spectrogram objects, the shape descriptors alone are limited, but they are enhanced when they are combined with orientation features (relative to some principal axes). Pinkowski has used 17 FDs features to characterize shape or contour which were obtained from each binary spectrogram image and used in the analysis. To classify the features, cluster analysis was used. Since such measures often contain redundant information [16], principal component analysis is used to reduce the size of a large feature set. The weak points of this study are, however, that (1) principal component analysis does not necessarily select important features for separating pattern classes [17], which means there is no perfect extraction of features of the shape pattern; (2) cluster analysis for a large number of features (17 FDs), and extracting them after converting the wave signal to a spectrogram image, can involve long delays which means it is difficult to apply in online speech applications given the time efficiency of this algorithm; (3) in this approach the problem of parameter signal extraction is converted to a problem of pattern extraction from an acoustic image generated by a binarized speech signal i.e. a binary image.

Steinberg and O'Shaughnessy [18], regard spectrograms as image patterns and perform segmentation in order to capture the energy associated with each formant using Mathematical Morphology operators, based mainly on the watershed transform operators as used in the watershed transform. Two levels of threshold segmentation are used and as a result this study has claimed better segmentation results than previous algorithms. However, segmentation remains a big challenge.

There is no existing study using images of speech representation as a basis for recognising uttered words rather than verifying extracted features from speech image representations based on phone information contained in the speech wave signal. In addition, all the studies cited above have described the spectrogram objects as a two dimensional binary image, in which case there are a major problems which are: firstly, threshold level segmentation estimation; secondly, misclassification when objects in the spectrograms are mirror images of one another (e.g. the words eel and lee), because the method is dependent on shape or contour features (number of pixels, x and y coordinates, maximum frequency, etc.) which give the same features for mirror image objects; thirdly, methods are not realisable for real time speech applications (due to long delays in processing).

Spectrogram-reading experiments have shown that the acoustic signal is rich in phonetic information [19]. Without knowing anything about the words that are present, an expert spectrogram reader can produce a broad phonetic transcription that agrees with a panel of phoneticians from 80% to up to more than 90 % phonemic accuracy, depending on the scoring method used [20]. The spectrogram provides a rough picture of the energy distribution over time and frequency, but there is no representation of harmonic structure, and pitch pulses and onsets are smeared out by the modulation filtering. The gross distribution of energy overtime and frequency, however, is the information that is best preserved in the presence of acoustic interference.

Consequently, we consider the speech spectrogram image as a written text in some unknown language and match pattern transitions of words associated with each formant. The result can be later used for ASR. The major focus of this thesis is suggested a new approach to speech recognition by using image processing on spectrogram speech images (SSI).

The thesis contains four main areas which are illustrated in Figure 2. In the thesis a combination of algorithms are used solve the problems occurring in these four areas. The performance of algorithms was assessed by calculating different test parameters; for more reliable results, a lexicon of 533 words was employed in this study for testing.

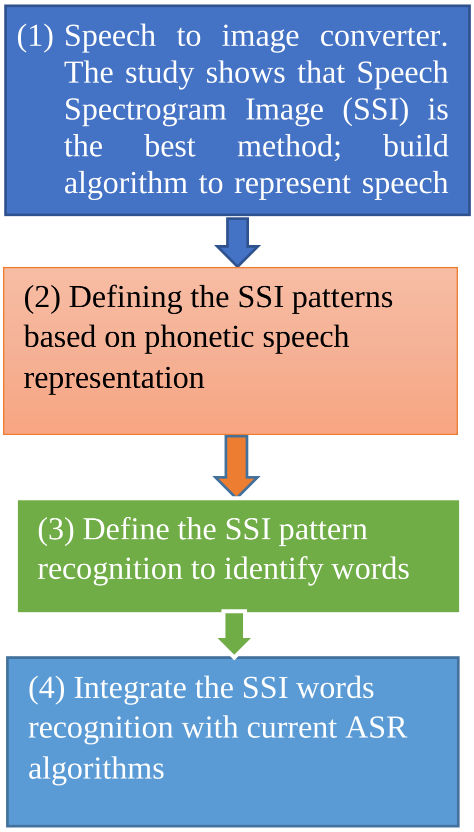

(1) Speech to image converter. The study shows that Speech Spectrogram Image (SSI) is the best method; build algorithm to represent speech
(2) Defining the SSI patterns based on phonetic speech representation
(3) Define the SSI pattern recognition to identify words
(4) Integrate the SSI words recognition with current ASR algorithms

Figure 2 Areas of study in the thesis.

Chapter 2 focuses on the methods of speech wave to image conversion and gives a comprehensive background of the time-frequency representation (TFR). In this chapter it is been discussed why the SSI is selected among the various TFR methods.

Also, considered are discussed what parameters can be used to control conversion of a speech wave to the SSI and what the optimum values of these parameters are.

In Chapter 3, the individual units of the SSI pattern speech representations based on the Phonetics and Phonology speech representation knowledge are described. These units are used to create a version of a Speech-Image Recogniser Code (basic SSI Patterns). The units of SSI patterns are clear start-end points which can help to solve major problem of the speech recognition.

In Chapter 4 the image processing methods for SSI Pattern Recognition are introduced, including elements of SSI analysis, Common tasks in SSI recognition are used suggest a general algorithm for SSI pattern recognition. The general algorithm is applied to a lexicon of 553 sample words of different genders and speakers and various utterances, for word level recognition.

In Chapter 5, the general algorithm for SSI recognition has been integrated with isolated word recognition by using a statistical approach to achieve for higher performance recognition.

Chapter 6 provides a discussion of research of thesis and describes possible future work.

CHAPTER TWO

2 CHAPTER TWO

Use of the spectrogram in speech analysis

2.1 Introduction

The speech waveform is a non-stationary signal the frequency of which changes over time. The analysis of non-stationary signals has been developed to improve the description of their frequency domain content. Each of these techniques has their own particular domain of focus which addresses certain, but not all, problems encountered in the analysis of non-stationary signals. A comparison of these techniques is presented below, including some practical examples illustrating how they can be used to assist in the analysis of a speech signal.

2.2 Time-frequency transforms

2.2.1 Preface

Mathematically, a signal can be represented in a variety of alternative ways which are appropriate for a given applications. It is well understood that in engineering applications, signals are usually a function of time but in the analysis signals and in designing systems, frequency domain representations of the signals are often used.

2.2.2 Primary method of analysis: the Fourier transform (FT)

It is well known that e^{x} is an elementary function[1], which can be used to express a function as a weighted summation of basic elements. The signal is real in nature, but writing it as its signal complex counterpart signal has advantages to overcome difficulties that arise when considering only a real signal (i.e. the energy

[1] Elementary Function: a function built up of a finite combination of constants, with the combination using the four elementary operations $(+,\ -, \times, \div)$.

density spectrum of a real signal is always symmetric about the origin, so the average frequency will always come out to be zero). So, a signal can be written in complex form as $s(t) = A(t)e^{j\varphi\,(t)}$, which is called a quadrature model. The $A(t)$ depicts the signal behaviour in the phase domain. This gives an intuition that there are an infinite number of waysto express the signal. Each new signal expression can be obtained by translating a signal through a linearity map[2] operator. This operator is called the shift operator, which translates a signal from a certain position to another which is displaced by a constant, a , within the domain x ; $e^{a\,x}\,f(x) = f(x+\ a)$. The derivative of the sum is the sum of the derivatives due to the linearity $\mathcal{A}(f+g) = \mathcal{A}f + \mathcal{A}g$. Also differentiation is a linear operator. Thus the operator can be represented as an eigenfunction with an eigenvalue: the operator d/dx operating on e^{ax} returns ae^{ax} , and hence e^{ax} is an eigenfunction with eigenvalue a. In this case there are an infinite number of eigenfunctions because it can be taken using any number of values a. The a values can be called a kernel of the transform.

The a matrix should be selected as a Hermitian, a self adjoint, symmetric, a complex square matrix. The mathematical interpretation of self adjoint requires the matrix to satisfy two properties: first the matrix a is bounded, which means the elements of the matrix a are real numbers that are not infinite. Secondly, when the matrix a is self adjoint then $\mathcal{A} = a\,e^{ax}$ and is linear self adjoint operator and $\int g^*(t)\mathcal{A}f(t)dt = \int f(t)[\mathcal{A}g(t)]^*\,dt$ or as vector expression $\langle \mathcal{A}x|y\rangle = \langle x|\mathcal{A}^*y\rangle$, so energy is conserved within the system. This makes it possible to calculate the dual function (dual space) of the analysed function, where $f(t)$ and $g(t)$ are paired functions [21]. The Fourier transform (FT) is a Hermitian transformer, which transforms a signal $s(t)$ in the time domain into a spectrum signal $S(\omega)$ in the frequency domain, where $s(t)$ and $S(\omega)$ are paired functions, i.e. FT pairs. In fact, the FT is an analysis of the density of the energy signal $P(t)$ to give the characteristic function[3] $M(a)$ of the signal where $M(a) = \int e^{jat}\,P(t)\,dt$. On the converse, the characteristic function is a summation of the

[2] Linearity map is the superposition principle, i.e., additively $f(x+y) = f(x) + f(y)$ with homogeneity $f(ax) = af(x)$.

[3] The term characteristic function is used in a different way in probability, is a function defined on a set X that indicates membership of an element in a subset *M* of X, having the value 1 for all elements of *M* and the value 0 for all elements of X not in *M*.

elementary functions of a signal $P(t) = \frac{1}{2\pi} \int M(a)\, e^{-jat}\ da$. This process is a synthesis linear combination of associated probabilities of the phase distribution. It is thus called a synthesis problem.

Thus a primary analytical tool is that the FT decomposes a signal as the sum of weighted sinusoidal functions. The FT is common in many applications and is suitable for stationary signals but it does not provide the best method to analyse finite signals (non-stationary signals) which occur in many real-life signals such as seismic signals, audio signals (including speech and music signals), transition signals, Radar signals and FM signals in broadcasting, to list some common examples. Joint time-frequency transforms were developed for the purpose of characterizing the time-varying frequency content of a signal. The well-known time-frequency representation of a time signal is the Gabor transform [22] and is known as the short–time Fourier transform (STFT).

The time-frequency signal representations (TFRs) analyse signals in a time-frequency plane which is a 2D time-frequency distribution and so the joint time-frequency distribution. TFRs give indications as to where the spectral components are present at which time. The TFRs are broadly classified into two categories: linear time transforms and bilinear transforms (quadratic transforms).

2.2.3 Linear time frequency representations

The linear transforms can be divided into two classes: STFT [23, 24] and time-scale representation i.e. various wavelet transforms (WT) [25, 26]. The WT uses a windowing process as in the STFT but the basic difference between the WT and the STFT is that the window width can be changed in the WT as a function of the analysing frequency whilst the STFT uses a fixed time-frequency resolution. The WT uses short windows at high frequencies and long windows at low frequencies. This capability has made multi-resolution analysis more useful in many practical applications. This allows low frequencies to last for the entire duration of the signal, whereas high frequencies appear localised in time as short bursts [26].

The TFR linear transform of non-stationary signals is basically a transformation between the time domain and frequency domain; conversely, the link between the time domain and the frequency domain is the instantaneous frequency[4] (IF). Essentially, the linear transform is a very useful description of the energy density or intensity of a signal simultaneously in both the time domain, which is called the instantaneous energy, and the frequency domain, which is called the energy density spectrum, so providing a powerful tool for the construction of signals with desirable properties. It allows the decomposition of a signal into individual frequency components and establishes the relative intensity of each component. In addition, it is very convenient for reconstructing the decomposed signals, which is one of the distinguishing features of a linear system. However, the TFR linear transform has both theoretical and practical shortcomings.

The practical shortcomings can be summarized in two points. Firstly, the linear transform is represented by the spectrum and so cannot be used to ascertain or define whether a signal is mono-component or not, although in some cases it may give an indication that components are present [27]. A good example of signals whose frequency content changes rapidly in a complex manner is human speech. Indeed, it was the motivation to analyse speech that led to the invention of the sound spectrogram[28]. In 1951, the Kay Electric Co. produced the first commercially available machine for audio spectrographic analysis. The spectrogram is a quadratic representation that performs the mapping of signals into a time-frequency space, showing the spectral component of a signal as a function of time. Secondly, windowing aims at assuring a local stationarity. However, we need different kinds of trade-off related to time-frequency localisation in the case of chirp signals i.e. a stationary signal. Thus, there exist natural and man-made signals whose spectral content is changing so rapidly that finding an appropriate short-time window is problematic since there may not be any time interval for which the signal is more or less stationary. Also, decreasing the time window so that one may locate events in time reduces the frequency resolution. Hence

[4] Instantaneous frequency is one of the basic signal parameters which provide important information about the time-varying spectral changes in non-stationary signals.

there is an inherent trade-off between time and frequency resolution [23, 29-31], which is unavoidable.

The shortcomings are thus related to how the linear TFR represents the frequency spectrum in a certain local time interval. This will be so, as long as a window function is used as the tool of the linear TFR to capture the shape of a signal and then analyse it. As previously mentioned, the time domain signal is divided into shorter data sequences, which usually overlap, and are then Fourier transformed to calculate the magnitude of the frequency spectrum for each sequence. The selection of window function is governed by the need to reduce the spectral leakage with the windowing function, which in turn leads to the trade-off in resolution between time and frequency domain localization. The normal trade-off variance inherent to any estimation procedure is amplified when analysing non-stationary stochastic processes [32]. Furthermore, the trade-off resolution must be consistent with the Heisenberg Uncertainty Principle[5] (windowing functions that are localized in the time domain have Fourier transforms that are spread out across the frequency domain and vice versa, a phenomenon known as the Uncertainty Principle). This has motivated many studies to address the problem of devising an adaptive window. The Heisenberg Uncertainty Principle is a fundamental limitation of mathematical transformation that must be considered when mapping the analysed signal into its TFR, since it forbids any precise temporal localization of frequency. Therefore, it has been argued that proper joint distributions cannot exist because of the Uncertainty Principle [12, 13]. In other words the Uncertainty Principle precludes the existence of proper joint TFRs.

The statistical consideration can be used as a way of introducing and interpreting the TFR. From this point of view, the trade-off in resolution is reflected by the variance of the window distribution. Moreover, the proper TFRs could be called nonnegative TFRs, because they have a correct marginal[6] (which is the probability of a

[5] The uncertainty principle has implications in two main areas: quantum physics and signal analysis. As developed by W. Heisenberg, it is a statement of the effects of wave-particle duality on the properties of subatomic objects. Consider the concept of momentum in the wave-like microscopic world. The momentum of a wave is given by its wavelength. A wave packet like a photon or electron is a composite of many waves. Therefore, it must be made of many momenta. But how can an object have many momenta? In signal processing is a fundamental statement regarding Fourier transform pairs.

[6] The word “marginal” is used by probability theory to indicate the individual distribution. The marginals are derived from the joint distribution by integrating out the other variables. The term is dubbed

single event happening and it is not conditional on any other event occurring). The positive joint distributions are easy to construct [6, 13], which can be interpreted as true energy densities. The positive result reflects the physical situation of having composed the signal from accumulating the signal's spectrum from zero to infinity (the real signal is always a symmetrical spectrum about the origin and the average frequency of the spectrum will thus be zero [26, 33]).

The doubt about the existence of positive TFRs is because of a paradox: the Uncertainty Principle depends only on the marginal integrating value, but the marginal carries no information about covariance. The question here is if it is possible that the Uncertainty Principle involves covariance [28, 29]. By this it is meant that any joint distribution that has these marginals should satisfy the Uncertainty Principle. Summarising, the theoretical shortcomings of the linear TFR are because it provides biased estimators of the signal IF and the group delay[7].

2.2.4 The bilinear time-frequency transform

The difficulties of the linear TFR have been recognized for some time. This was the main motivation [33] for an approach to improve upon the spectrogram, given a fundamental analysis and thus clarification of the physical and mathematical ideas needed to understand what a time-varying spectrum is. There have been alternative approaches, but researchers have put together a unified approach to define the TFR. The idea initiated as a formatting of the TFR by the multiplicative comparison of a signal with itself, providing a means for finding repeating patterns, expanded in different directions about each point in time. A means for deriving time-dependent spectra is by generalizing the relationship between the PSD and the autocorrelation. Such

"marginal" because they used to be found by summing values in a table along rows or columns, and writing the sum in the margins of the table. Hence we can say that $|s(t)|^2$and $|S(w)|^2$are the marginals of $P(t,w)$, as in each case the instantaneous energy $\int P(t,w)\ dw = |s(t)|^2$ or energy density spectrum (the power spectrum of the signal) $\int P(t,w)\ dt = |S(w)|^2$.

[7] The group delay describes the time lags among different frequencies, which measure the propagation time through a system as a function of frequency. Thus, this quantity measures the average time arrival of the frequency ω , $\tau_x(\omega) = -\frac{1}{2\pi}\cdot\frac{d(\arg X_a(\omega))}{d\omega}$, $X_a(\omega)$ signal spectrum. X_a is an analytic signal. As a linear system block with frequency domain transform $H(j\omega) = e^{-j\omega T}$, group delay is the ideal element that delays a signal by time T.

formulations are known as quadratic TFRs (QTFRs) because the representation is quadratic in the signal. This formulation was first described by Wigner in quantum mechanics [34] and introduced in signal analysis by Ville to form what is now known as the Wigner-Ville distribution (WVD) [23, 24, 33]. The WVD is the prototype of distributions that are qualitatively different from the spectrogram, and produces the ideal energy concentration along the IF for linear frequency modulated signals.

The WVD, like other bilinear transforms has a shortcoming in that the zero frequencies, which should be zero amplitude, have a false component introduced. Those false terms are known as “interference cross terms “, that can case difficulties in the projections in the time-frequency space and in the reconstruction of the signal, hence distributions cannot be considered full correct. A large area of research has been devoted to reduction of these cross-terms, using different time-frequency kernels.

In 1966 a formulation was made by Cohen, applied to quantum mechanics [35], which included these and an infinite number of other methods as kernel functions. The formulation by Cohen was restricted with a constraint on the kernels so the marginals could be satisfied. This establishes what is known as Cohen’s class. A large number of bilinear TFR’s have been proposed, each differing only in the choice of a kernel function [23, 26].

A unified approach can be formulated in a simple manner with the advantage that all distributions can be studied together in a consistent way, the general form being written [23]:

$$C(t,\omega) = \frac{1}{4\pi^2}\int\int\int\left(s^*\left(u-\frac{1}{2}\tau\right) * s\left(u+\frac{1}{2}\tau\right)\right) \emptyset(\theta,\tau)\, e^{-j\theta t - j\tau w + j\theta u}\, du\ d\tau\ d\theta \qquad \text{Equation 1}$$

where $\mathrm{C(t,\omega)}$ is the spectrum of the signal s. $\emptyset(\theta,\tau)$ is called the kernel.

Historically, the kernel identified by Claasen and Mecklenbrauker [36]. The kernel may depend explicitly on time and frequency and in addition may also be a functional of the signal. If the kernel is independent of the signal, then the distributions are said to be bilinear because the signal enters only twice. The approach characterizes the TFR by an auxiliary function kernel $\emptyset(\theta, \tau)$, allowing the selection of a kernel that produces a distribution with prescribed and desirable properties. This approach provides a simple way of examining different kernels since their effects are reflected in the properties of the signal distribution. In addition, since the kernel designs treat time and frequency on an equal footing, the two constraints, one for each domain, collapse into one. This means the Heisenberg Uncertainty Principle is satisfied by the time and frequency marginals, $\emptyset\ (0, \tau) = 1$ and $\emptyset\ (\theta, 0) = 1$. Thus, a kernel design is a way to overcome practical and theoretical shortcomings of the linear TFR.

Hence, the kernel should be independent of time and frequency. $e^{j\tau w}f(t) = f(t+\tau)$ is a translation of the signal by an amount τ in the distribution. That is, if the spectrum is a fixed frequency at the shifting moment, a shift in the time domain of the signal produces a corresponding shift in the frequency domain $e^{j\theta\tau}S(\omega) = S(\omega+\theta)$ by a constant frequency $\theta.$ Both of these cases can be handled together.

There are two advantages for seeking decomposition of a signal. The first advantage is a classification by doing a singular value decomposition of each class to get a set of singular values and eigenfunctions that are all different from each other. The second is extraction a signal in a noisy environment. Actually, decomposing the source of signals into sinusoidal components in the frequency-domain does not allow extraction of the whole signal. Rather, the sinusoidal signal decomposition cannot be applied to extract small duration of the signal since these are small magnitude and pattern-less, and may be just random noise perturbations [37]. Thus researchers have tried to solve that shortcoming by adding a new domain, the scale domain (du) to the two traditional domains of time $d\tau$ and frequency $d\theta.$ A scale-domain description of a signal breaks it into similarly shaped signal fragments of varying sizes. Actually, the concepts developed for multicomponent signals in the time-frequency plane can be generalized to generate the concept of a time scale plane [27]. The scale domain can be the instantaneous scale and the spread of instantaneous scale. The scale domain analysis is approached using two concepts, the correlation method and the Doppler effect[8] of electromagnetic waves between a source and a receiver. One of the three scenarios is when the distance between the source and receiver of electromagnetic waves remains constant, in which case the frequency waves is the same at both the source and the receiver. Those frequencies are estimated by the correlation method which is used to determine the of time delay between replicas of an unknown continuous waveforms when these replicas are contaminated by additive noise [38]. Extension of this method requires estimation a joint expression of the differential time offsets τ and differential frequency offsets θ. This joint is the complex ambiguity function, which is a joint of the time and Doppler frequency which is thus the natural generalization of the correlation process [39]. It can be used to handle sources of nonstationary waveforms that are highly localized in the TFR and, provides extraction of the signal from background additive noise. The Equation 1 can then be rewritten as:

$$C(t,\omega) = \frac{1}{2\pi} \int \int e^{-j\theta t - j\tau\omega} \cdot A\,(\theta,\tau) \cdot \emptyset(\theta,\tau)\; d\tau\; d\theta \qquad \text{Equation 2}$$

where,

$$A\,(\theta,\tau) = \frac{1}{2\pi} \int e^{j\theta u} \cdot S^{*}\left(u - \frac{1}{2}\tau\right) S\left(u + \frac{1}{2}\tau\right) du \qquad \text{Equation 3}$$

$A\,(\theta,\tau)$ is the symmetrical ambiguity function, $\omega = j\frac{d}{dt}$ is the frequency operator in the time domain, ω is the spectra of a signal, with representation in the frequency domain θ, $\omega \in\ \theta$,$\tau = j\frac{d}{d\omega}$ is the time operator in the frequency domain, t is the moment of a signal and represents the signal in the time domain, $t\ \in\ \tau$.

By way of example and by no means exhaustively, some well-known bilinear TFRs are: Choi-Williams, which is one of the members of Cohen's class of transform functions [40]; Zhao-Atlas-Marks transforms [41]; second-order quadratic

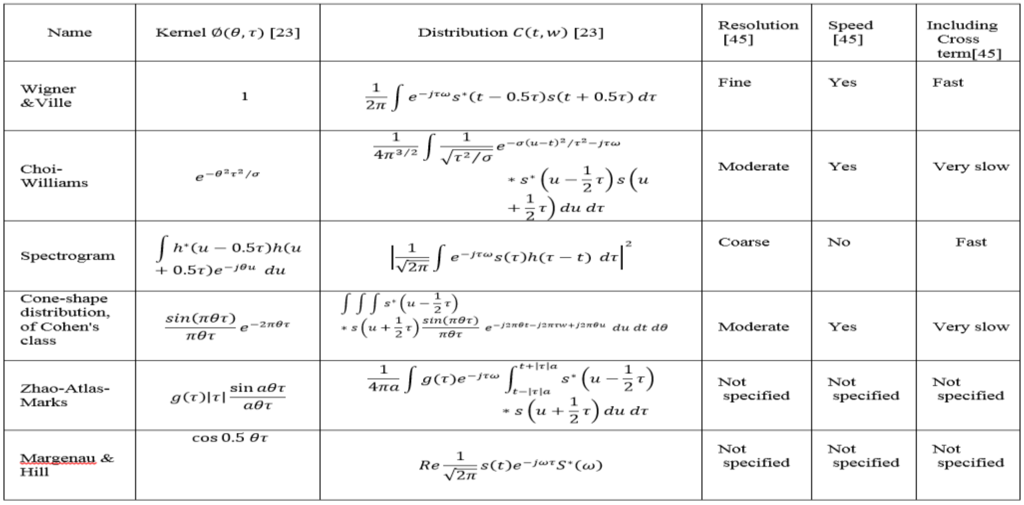
[8] The Doppler effect is the change in frequency of a wave for an observer moving relative to its source, which was proposed by Austrian physicist Christian Doppler in 1842.

representations which are spectrograms and the so-called scalograms. The scalogram is a visual method of displaying a WT and is the squared modulus of the wavelet transform [25, 42]. The quadratic representations are the squared magnitudes of the linear TFRs.

All bilinear TFDs belong to of Cohen's class of transforms. The WVD is a prototype TFR. In fact, the WVD is a short-time Fourier transform with a window function that is perfectly matched to the signal. The WVD is highly concentrated in time and frequency and is a quasi-probability distribution, but it is also highly nonlinear and non-local. The Cohen's class is a kind of "smoothed" WVD, employing a smoothing kernel that can reduce sensitivity of the distribution to noise and restrains cross-components [43], at the expense of smearing the distribution in the TFR. This smearing causes the distribution to be non-zero in regions where the true WVD shows no energy.

The quadratic representations are 2D matrices. Interpretation quantitatively might not be straightforward but should allow the extraction of useful 1D information such as the computation of the mean instantaneous frequency, the mean instantaneous bandwidth[9], the group delay and the marginal integration[10][24, 44]. Table 1 shows a list of some quadratic distributions and their corresponding kernels together with their application properties.

Table 1 listing of some well-known quadratic distributions, their corresponding kernels and their application properties.



Name	Kernel $\emptyset(\theta,\tau)$ [23]	Distribution $C(t,w)$ [23]	Resolution [45]	Speed [45]	Including Cross term[45]
Wigner &Ville	1	$\frac{1}{2\pi}\int e^{-j\tau\omega}s^*(t-0.5\tau)s(t+0.5\tau)\,d\tau$	Fine	Yes	Fast
Choi-Williams	$e^{-\theta^2\tau^2/\sigma}$	$\frac{1}{4\pi^{3/2}}\int\frac{1}{\sqrt{\tau^2/\sigma}}e^{-\sigma(u-t)^2/\tau^2-j\tau\omega}*s^*\left(u-\frac{1}{2}\tau\right)s\left(u+\frac{1}{2}\tau\right)du\,d\tau$	Moderate	Yes	Very slow
Spectrogram	$\int h^*(u-0.5\tau)h(u+0.5\tau)e^{-j\theta u}\,du$	$\left\lvert\frac{1}{\sqrt{2\pi}}\int e^{-j\tau\omega}s(\tau)h(\tau-t)\,d\tau\right\rvert^2$	Coarse	No	Fast
Cone-shape distribution, of Cohen's class	$\frac{sin(\pi\theta\tau)}{\pi\theta\tau}e^{-2\pi\theta\tau}$	$\int\int\int s^*\left(u-\frac{1}{2}\tau\right)*s\left(u+\frac{1}{2}\tau\right)\frac{sin(\pi\theta\tau)}{\pi\theta\tau}\,e^{-j2\pi\theta t-j2\pi\tau w+j2\pi\theta u}\,du\,dt\,d\theta$	Moderate	Yes	Very slow
Zhao-Atlas-Marks	$g(\tau)\lvert\tau\rvert\frac{\sin a\theta\tau}{a\theta\tau}$	$\frac{1}{4\pi a}\int g(\tau)e^{-j\tau\omega}\int_{t-\lvert\tau\rvert a}^{t+\lvert\tau\rvert a}s^*\left(u-\frac{1}{2}\tau\right)*s\left(u+\frac{1}{2}\tau\right)du\,d\tau$	Not specified	Not specified	Not specified
Margenau & Hill	$\cos 0.5\ \theta\tau$	$Re\frac{1}{\sqrt{2\pi}}s(t)e^{-j\omega\tau}S^*(\omega)$	Not specified	Not specified	Not specified

[9] The instantaneous bandwidth is an indication of the frequency spread at a given time.

[10] The marginal Integration is integration of the spectrogram along the time axis which equals the power spectrum of the signal.

2.3 Why the STFT is an effective method for speech signal analysis

Cohen's class quadratic (which is QTFR) was designed for continuous signals in quantum mechanics, whereas the applications in signal analysis are to discrete time signals. The extension of TFR from a continuous plan to a discrete TFR plane is not straightforward. Unfortunately, all the properties of the continuous QTFR are not preserved by discretization, due to effects aliasing in the discrete version. This means after the limitation imposed by the Uncertainty Principle on the small time intervals is addressed by using the Cohen's class quadratic, there is new limiting factor which is the Uncertainty Principle applied to the whole bandwidth of the signal. The short duration signals have inherently large bandwidth. There are many studies that have examined applying WVD to discrete date [45-47]. The limitation of this technique is that it estimates the spectrum of short duration signals [21, 28]. But, in practical applications, it could be that there occurs a signal that is a long transient of a nonstationary process. Thus, the most common approach is to divide the time domain signal up in to short transient signals by a sliding window approach before analysing the contained signal by the QTFD. However, there may be another problem that would arise of choosing the time length of the window that matches the interval in which the signal undergoes significant spectral variation, that is, the interval in which the signal is considered "essentially stationary" with respect to the window. The question then arises as to how to deal with infinite sequences of arbitrary forms such as the nonstationary signals contained within a speech pattern.

Added to these are problems of nonlinear TFR. Historically, the STFT was developed before other QTFD methods during the 1940's century as mentioned in Section 2.2.3. The STFT was developed for speech analysis to display and estimate the multiple component of a speech signal where the components are called formants (read visually as of speech spectrogram images [48]), with some subsequent developments being applied to the analysis of nonstationary signals in general. Different TFRs roughly give the same results in regard to the existence of the various components with maybe somewhat different representations [27]. That made the STFT became a standard powerful tool for the analysis of speech signals and other nonstationary signals [23, 48-54]. The concept behind it is simple and can be implemented by using the fast Fourier transform (FFT). It provides powerful estimate of the spectral density of a signal with a simple interpretation of how the signal frequency spectrum varies as a function of time when the signal is stationery.

Also, the STFT is invertible after taking in to consideration overlap of the frames causing artefacts at the boundary. The inverse STFT (ISTFT) allows the original signal to be recovered from the transformed signal and is thus an important and versatile signal processing method [55]. In contrast, the summation for computing the dual function could be unbounded, which is not suitable for numerical implementation. This has motivated many researchers to seek and prove existence of the discrete version of the STFT and ISTFT [56-60].

2.4 Spectrogram

The spectrogram is used to display the magnitude variations of the spectral signal versus time as a three dimensional plot (i.e. time vs. frequency as amplitude). As mentioned in Section 2.2.4, the spectrogram has been developed for representing sound data and is similar to the process employed in human hearing which is based on forming a real-time spectrogram encoded by the cochlea of the inner ear and used by the brain to classify and recognise patterns of sound samples. The spectrogram of signal s is estimated by computing the squared magnitude STFT of the signal. It is important to note that the phase of the signal is not retained din this process. Thus:

$$Spectrogram\ (\tau, \omega) = \ |STFT(\tau, \omega)|^2 \qquad \text{Equation 4}$$

Actually, the STFT splits the time domain signal into many frames of length L by the windowing processing, and then take the FT of each frame as shown in Figure 3.

If the window $h(k)$ has the constant hop size M (the hop size is the number of samples between the start-times of adjacent frames), then the STFT is a matrix of size (m, n).

$$C_{m,n} = STFT[m\,M\,, n\,] = \sum_{k=0} s[k]\;\; h^{*}[k - m\,M]\, W_N^{-nk} \qquad \text{Equation 5}$$

where M is the hop size, m is frame index, and n is the size of the discrete Fourier transform (DFT) which is equal to the hop size. When, the length of the hop size is less than n, the segment of the hop size is padded with trailing zero to length n.

2.4.1 The windowing process

For a given input signal $s(t)$ of arbitrary duration equal to L_s , the windowing process extracts data segments at regular intervals using a window limited $h(k)$; these signal segments or frames can be expressed as $s_m(k) = h(k)s(k + mM), 0 \leq k \leq L_{WD}$, where L_{WD} is the window length, m is a frame index, and M is the hop size (i.e. the number of samples between the star-times of adjacent frames) and the index k is the local time index , i.e. an index (not length) relative to the start of the sliding window. The $s_m(k)$ is the modified signal, which is a function of two times; the fixed time (local time) upon which the window is centred on at k, and the running time,n. So, we can say $f_s(k; n)$ is a frame of signal $s(k)$ of length L_{WD} (window duration) ending at n, i.e. $f_s(k; n) \;= s(k)h(n - k)$. This period then covers the small duration of signal where the signal is time invariant and gets smeared out as vertical striations on the spectrogram due the frequency content in this duration. The number of striations is equal to the number of frame, $m = \left\|\frac{L_s}{M}\right\|$ which is the ratio of fixed time to running time.

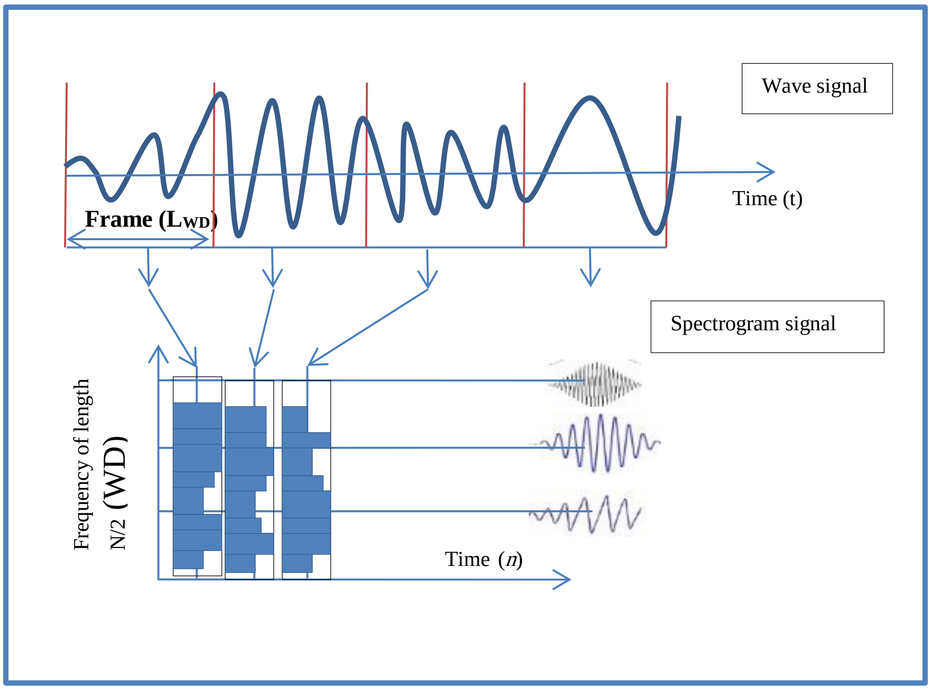


Figure 3 illustration of the Short-Time Fourier Transform (STFT).

The simple window is a rectangular window. The power spectrum of a rectangular window (low pass filter) is shown in Figure 4.

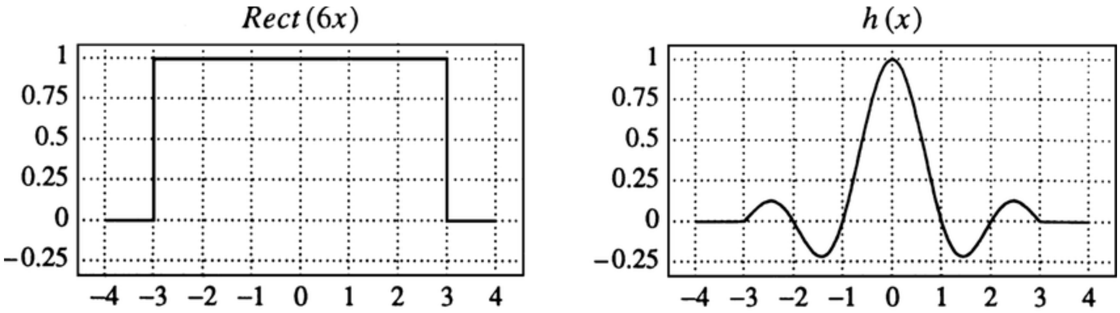


Figure 4 shows the rectangular window and its magnitude spectrum. Note that the normal bandwidth is $\mathbf{2\pi/n}$.

Therefore desirable features of a window are a narrow bandwidth main lobe and large attenuation in the sidelobes. The window function length L_{WD} is inversely proportional to the bandwidth of main lobe (i.e. the width of the lobe is decreases with L_{WD}) and approximately constant with attenuation of the sidelobes. Therefore, a large value for L_{WD} begins to defeat the purpose of windowing. The difficulty caused by abrupt truncation is solved by use of the smoother truncation windows such as the Kaiser, Hamming, Hanning, and Blackman windows. Actually, smoother windows tend to distort the temporal waveform on the range of L_{WD}, but with the benefit of less abrupt truncations at the boundaries (i.e. 10-60 dB better than a rectangular window for example the sidelobe attenuation of the popular Hamming window being -30dB) [1]. In addition, the smoother windows have a wider main lobe than the rectangular window for a given L_{WD}. As a consequence of that, the windowing process is mainly defined by setting window type, length of the window and the percentage of windows overlap. The overlap of the windows is to compensate for the loss of signal energy, which is drops because of abrupt truncation at the boundaries of the window and causes spectral leakage in the frequency domain. The overlap of M samples of the rectangular window is 50% of L_{WD}, while the overlap of smoother windows, such as the Hamming, is 75% [61].

2.4.2 Window length in speech processing

Generally in speech processing, smoother windows such as the Hamming window are used, so the window type is fixed. To improve the spectral resolution, L_{WD} must be increased to get more time domain information. However, L_{WD} is bounded by two limitations, which are the stationary duration of the signal and the Uncertainty Principle. Thus, the maximum length of L_{WD} is limited by the stationary duration of the signal. The speech signal is a slowly time varying signal over short periods of time between $5\ and\ 100\ msec.$ As a rule of thumb, the window length in speech can be assumed to remain stationary for frames on the order of $20\ msec$ [3].

The Uncertainty Principle involvement in the waveform analysis is not concerned with measurement of the time energy density $|s(t)|^2$and frequency energy density $|S(\omega)|^2$; instead it states that the effective duration of a signal cannot be less than the inverse of the effective bandwidth of the signal WD [21]. Typical human speech communication is limited to a bandwidth of $7-8\ kHz$, with effective speech possible with a bandwidth $WD = 3.5-4\ kHz$, so the estimated minimum duration of the window is $1/WD = 0.25\ msec$.

2.4.3 Speech spectrogram types

The Uncertainty Principle is often written: $\Delta t \Delta \omega \geq \frac{1}{2}$, (Δ represents the standard deviation). It is important to mention that using the word ‘uncertainty’ is a misnomer when applied to single processing. The Uncertainty principle is simply states that is that a narrow waveform Δt yields a wide spectrum $\Delta \omega$, and vice versa, and each cannot be made small simultaneously. Thus consideration of the spectrum to select the length of window is preferred. The trade-off of processing between Δt and $\Delta \omega$ forces concentration on one variable, by defining the stationary time period and allows the frequency domain is stable during the design process leading to both a narrow-band spectrogram and a wide-band spectrogram.

In narrow-band spectrogram analysis (long time window length), the bandwidth is appreciably less than the fundamental frequency of phonation. It is useful for determining the intonation (tone) of an utterance by showing the harmonics structure clearly, but blurs the rapid changes. The narrow- band window analysis is used to separate the individual harmonics (phones) of the voiced excitation source [13, 62]. Figure 5 shows a narrow-band spectrogram of the sentence “she had your dark suit in greasy wash water all year” created with a window length equal to 35 msec.

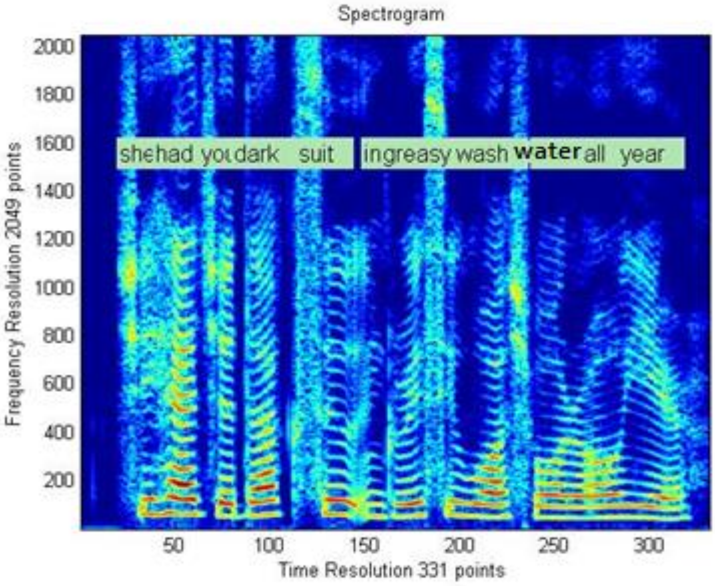


Figure 5 narrow-band spectrogram of “she had your dark suit in greasy wash water all year’’.

In wide-band spectrogram analysis (short time window length), the bandwidth contains at most the response of the fundamental frequency of phonation and thus cannot display the harmonic structure because, the bandwidth of the equivalent filter is wider than the fundamental frequency and so the harmonics will not be separated. However, it is very accurate in the time dimension, showing each vibration of the vocal cords as a separate vertical line and indicating the precise moment of a stop burst with a vertical spike. It is thus suitable for the separation of the phonemes [13, 62]. Figure 6 shows a wide-band spectrogram of the sentence “she had your dark suit in greasy wash water all year’’ created with window length equal to 5 msec.

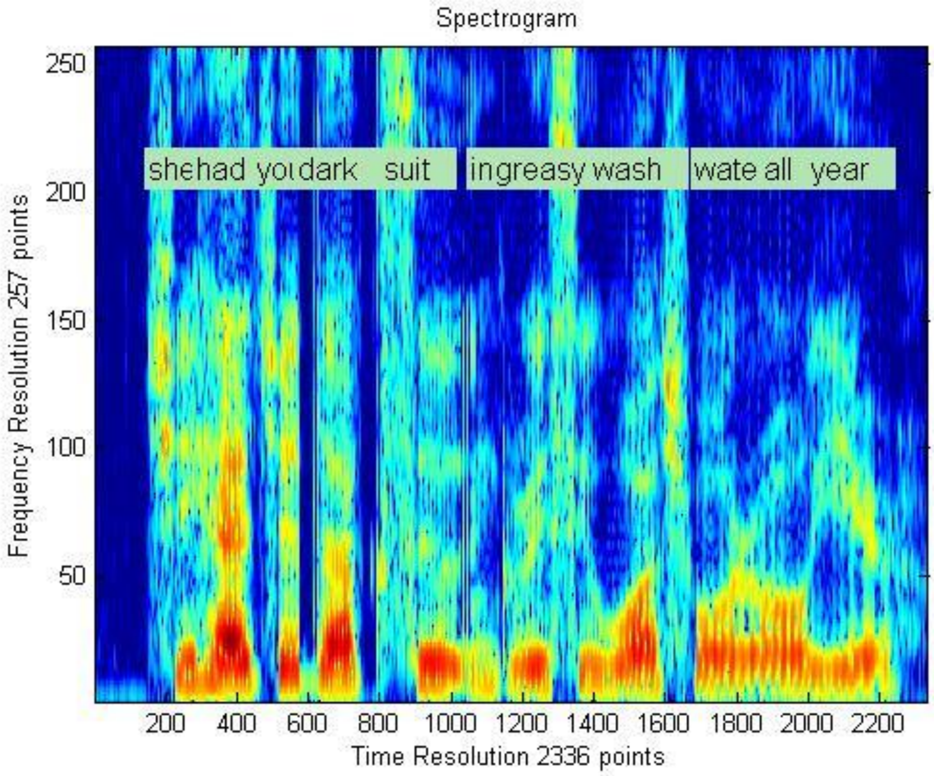


Figure 6 wide-band spectrogram of “she had your dark suit in greasy wash water all year’’.

2.4.4 Window size versus data length for FFT

The window function is usually described in $msec$ to make it independent of the frequency sampling. The length of window in sample points is $L_{WD} = T_{WD}/T_{fs} = T_{WD} \cdot f_s$. as shown in Figure 7.

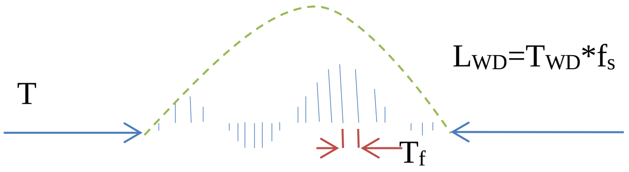


Figure 7 relation between length of window and sampling frequency.

Increasing the length of the FFT increase the resolution of the sampling frequency of the spectrum, which makes the SSI smoother. As an example, a narrow band spectrogram, obtained using a time domain window of a duration of 64 $msec$ is sampled at 16 kHz (equivalent to length of 1024 sample points) may be directly frequency analysed by an FFT of length 1024. Then, the FFT gives a resolution of 15.6 Hz which is more than reasonable for display of a speech signal, the smaller frequency differences between the first three formats being around 300Hz [63]. The display resolution will increase when the window is decreased to get a wide-band spectrogram whilst employed fixed length of FFT of 1024 sample points.

2.5 Time localisation of the signal on Spectrogram (Running time)

The need for inspection of a local time (i.e. a certain event) in a signal on its spectrogram and vice versa visa might be necessary to extract information from the signal. As mentioned in Section 2.4.1, the number of frames is equal to the ratio of the fixed time to the running time. That means each frame of the signal contains M points to be displayed as a point in running time of the spectrogram. As shown in Figure 8, a duration of fixed time of length M $\approx$ 20 points are displayed as stripe on spectrogram. Thus, the location of stripe on the spectrogram, i.e., localisation of fixed frequencies, corresponds a period of length M in the fixed time of the signal. That is an indication of an error equal to the time resolution Δt of the spectrogram. This explains why the wide-band spectrogram is suitable for standing phonemes.

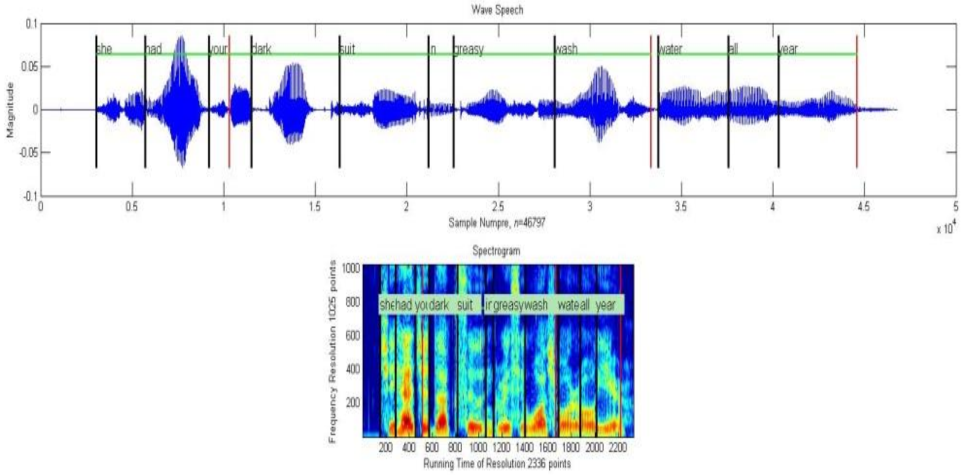


Figure 8 illustration of the ratio of the fixed line to running time.

There are some speech databases that have comprehensive details like the exact duration of the phones (start and end of words and phones). The time location in SSI can help to verify such information above.

2.6 Spectrograms information superiority waveform representation

As mentioned in Section 2.4.1 the speech signal is stationary within the short term. Spectrograms are better than waveforms of segments to provide reliable measurements since the differences among vowels, nasals and laterals can be seen on spectrograms, whereas it may be impossible to see these differences in the waveforms. Those differences could be related to linguistic properties (i.e. places of articulation) and voice quality aspects of a speaker's speech habits.

2.6.1 The spectrogram as linguistic

Linguistic spectrograms show display the following qualities:

a) Vowel quality is indicated by the spectrogram. The vowel quality is described by the International Phonetic Alphabet chart (schematic IPA vowel diagram) which describe vowels in terms of three common articulatory features: height (vertical dimension), tongue backness (horizontal dimension) and roundedness (lip articulation). The first formant certainly show relative vowel height quite accurately and the distance between the first and second formants reflect the degree of blankness quite well, but there may be misperceptions due to differences in the degree of lips rounding.

b) Voice from voiceless sound can be separated by the spectrogram.

c) Consonants begin as stops, because affrication of a stop can be seen on most occasions. For example one can usually see whether a stop has been weakened to a fricative, or even to an approximant.

d) Consonant trill sounds can be separated from taps since one can also observe the relative rates of movement of different articulations.

However, Spectrograms cannot be used to: measure degrees of nasalization or differentiate between adjacent places of articulation.

In person's voice, there are individual characteristics recorded in the spectrogram which are indicative of the speaker's voice quality rather than the linguistic aspects of the sounds. These are: the position of the fourth and higher formant of vowels; and the rate of transition of the formants after voiced stops that give individuals shape related to the speaker's voice quality.

2.7 Conclusion

The aim of this chapter was to represent speech waves as an image and interpret their patterns. We have used the SSI for the purpose of analysing the speech signals. This has been done for the following reasons: first, the SSI is easy to implement and is suitable for speech online application; second, the SSI effectively represents most speech wave fluctuations. This is one of the reasons that it is popularly used in many studies. Thus, it is realistic to use the SSI patterns for creating an initial image representation and a classification model. The study employs narrow band spectrogram analysis to create the SSI because of its high frequency resolution. This also paves the way for other types of speech wave to be represented as images to allow for improvements in SSI patterns classification.

CHAPTER THREE

3 CHAPTER THREE

Phonetics, Phonology and SSI patterns for speech representation.

3.1 Introduction

A speech segment is a unit that can be identified by either physical (place of articulations) or auditory (consonants and vowels) characteristics. Speech segments are considered as a linear sequence to give meaningful field analysis, such as a mora or a syllable, or a morpheme in morphology [64]. Computational linguistics is a combination of knowledge in the linguistics, computer and electrical engineering fields to develop methodologies and technologies to enable recognition and translation of a spoken language into text by computer. Phonemes show significantly lower redundancy than letters, but redundancy has an effect of increasing the task difficulty of first- or second-order phonemic probability guessing (first- or second order HMM) [65]. In this chapter, we work to define a written transcription that can help either in enhancing speech recognition or can establish a new approach for speech recognition.

3.2 Phonetics and Phonology speech representations

Speech interpretation can be classified, from the engineering point of view, into two levels: an acoustic processing (which is the low level) and a language processing (which is the high level). The two processing stages are guided by phonetics and phonology, respectively. Phonetics and phonology are integrated in order to understand the speech of any language. Phonetics is concerned with the physical acoustic production, the transition and perception of speech sounds by using phoneme units. Whereas phonology is concerned with the way the sounds are gathered across a language to encode meaning by using phoneme units. Phonetics is descriptive linguistics, whilst phonology is theoretical linguistics. In American English, a set of 62 alphabetises may form a code which can be encoded and decoded for understanding speech in the English language. The well-known phoneme classification (41 phonemes) is based continuant/ noncontinuant properties [1]. The continuant (stationary) are vowels and consonants. The continuant consonants are: fricatives, whisper, affricates, and nasals. The non-continuant are: diphthongs, semivowels and stops.

From the listener's point of view, speech sounds are linear combinations the units of phonemes, syllables and words. Unfortunately, speech sounds physically overlap. The speech sound unit is influenced by both surrounding sounds, the preceding sound and the following sound. On the other hand, the speech sound signal has some acoustics cues (e.g. voice onset, places of articulation, stress, etc.) that are used to differentiate speech sounds in phonetic categories. The speech recognition problem can be defined as finding an accurate written transcription of spoken utterances in units (phonemes, syllables words or other units). The accuracy of transcription is measured in terms of the distance (the smaller the distance the more accurate) between a reference transcription and a sequence output in a continuous speech model.

3.3 Phoneme classification

The phonemes can be classified based on different properties. The history of developing and revising them are basic to the understanding and applications of the speech field:

1. Time waveform: gives a primary interest in waveforms and what they reveal about the physiological and acoustic aspects of speech.
2. Frequency waveform: gives significant information about the physical phenomena from frequency domain plots derived from acoustic waveforms.
3. Manner of articulation: gives the configuration and interaction of the articulators (speech organs such as the tongue, lips, and palate) when making a speech sound. It describes how closely the speech organs approach one another.
4. Place of articulation: describes the place of contact where an occlusion occurs in the vocal tract between an articulatory gesture, an active articulator (the degree of narrowing in the oral tract) and a passive location which gives a consonant. This gives a consonant its distinctive sound.

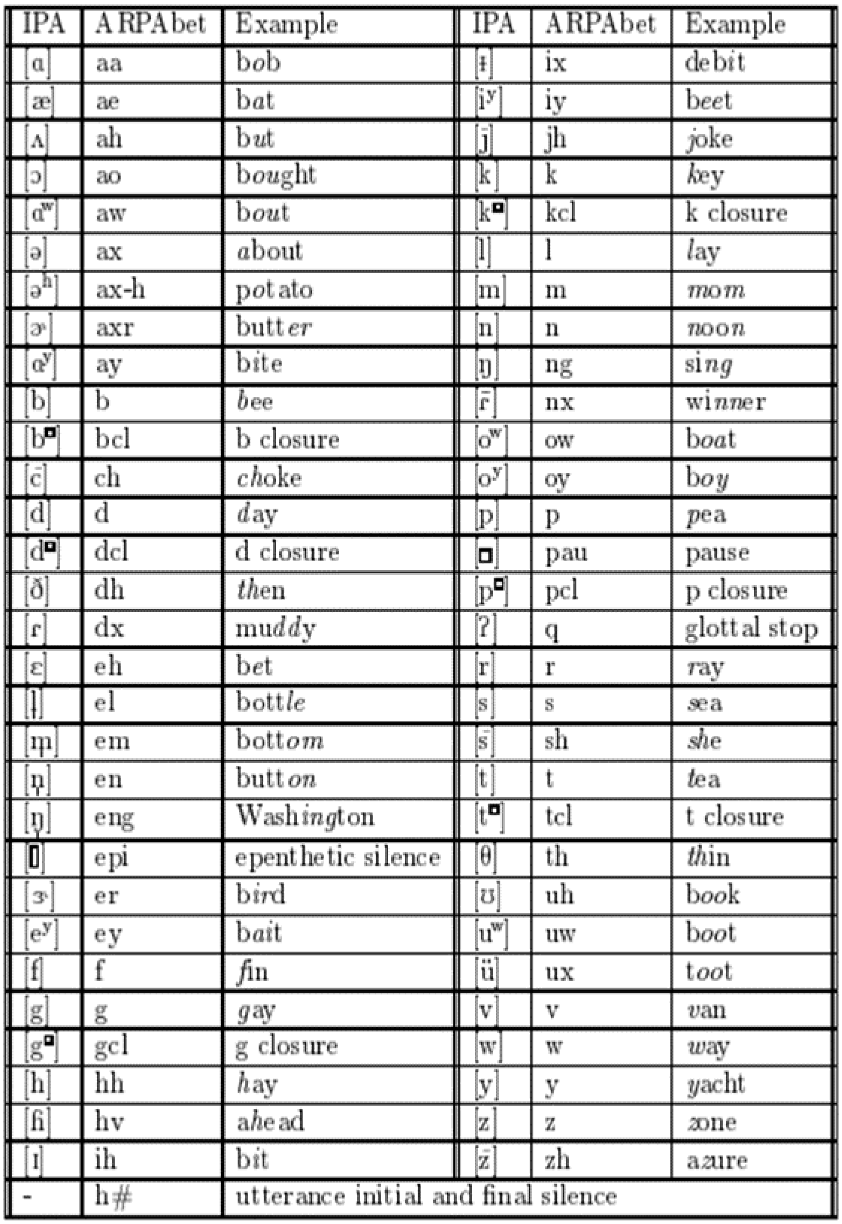
5. Type of excitation: the speech sound is produced in two elemental excitation manners: voiced and unvoiced excitation. The combinations of voiced and unvoiced and silence are usually outlined for modelling and classification purposes (mixed, plosive, whisper and silence).
6. Stationary phonemes: describe if the speech sound is produced by a steady-state vocal tract configuration. A phoneme is a non-continuant if a change in the vocal-tract configuration is required during production of the speech sound.

So, speaking can be described as trying to associate a symbol to each sound in all of the known languages in the world. The most known set of symbols for phonetic transcriptions is the International Phonetic Alphabet (IPA). In Paris in 1886 a small group of teachers and linguists from France, Germany, Britain and Denmark (recently called the International Phonic Association) formalized a meeting. This group created a standardized format for expressing the phonetic sounds used in the various spoken human languages. At the time of the IPA, classification started based on subjective methods (Auditory phonetics: the study of the reception and perception of speech sounds by the listener). A phoneme is the basic theoretical unit of a language for describing how speech conveys linguistic meaning. It is a distinctive unit of sound because the whole of a phoneme must be substituted to make a different word. The IPA has been developed and revised several times since the end of the 18th century based on objective methods (Acoustic phonetics: the study of the physical transmission of speech sounds from the speaker to the listener using phones which are small units of sounds produced in speaking).

3.4 The International Phonetic Alphabet (IPA)

This phonetic notation which is based on the Latin alphabet, tries to represent only those qualities of speech that are part of oral language: phonemes, intonation, and the separation of words and syllables. For representing additional qualities of speech (i.e. teeth gnashing, lisping and sounds made with a cleft palate) an extension to the IPA may be used in a narrow transcription. Among the symbols of the

English IPA, 20 letters represent vowels and 24 letters represent consonants, and 5 additional signs indicate suprasegmentally qualities such as length, tone, stress, and intonation [1], with the overall total being 49 phonemes. The IPA is not usually available on computers because it makes extensive use of letters. So, the ARPABET were proposed as mappings from IPA to "computer-friendly" ASCII symbols. The ARPABET of the English language are based on 62 phones, the table below showing the IPA and ARPABET symbols of phonemes [66].



IPA	ARPAbet	Example	IPA	ARPAbet	Example
[ɑ]	aa	b*o*b	[ɨ]	ix	deb*i*t
[æ]	ae	b*a*t	[iʸ]	iy	b*ee*t
[ʌ]	ah	b*u*t	[ǰ]	jh	*j*oke
[ɔ]	ao	b*ough*t	[k]	k	*k*ey
[ɑʷ]	aw	b*ou*t	[k□]	kcl	k closure
[ə]	ax	*a*bout	[l]	l	*l*ay
[əʰ]	ax-h	p*o*tato	[m]	m	*mom*
[ɚ]	axr	butt*er*	[n]	n	*noon*
[ɑʸ]	ay	b*i*te	[ŋ]	ng	si*ng*
[b]	b	*b*ee	[ɾ̃]	nx	wi*nn*er
[b□]	bcl	b closure	[oʷ]	ow	b*oa*t
[č]	ch	*ch*oke	[oʸ]	oy	b*oy*
[d]	d	*d*ay	[p]	p	*p*ea
[d□]	dcl	d closure	□	pau	pause
[ð]	dh	*th*en	[p□]	pcl	p closure
[ɾ]	dx	mu*dd*y	[ʔ]	q	glottal stop
[ɛ]	eh	b*e*t	[r]	r	*r*ay
[l̩]	el	bott*le*	[s]	s	*s*ea
[m̩]	em	bott*om*	[š]	sh	*sh*e
[n̩]	en	butt*on*	[t]	t	*t*ea
[ŋ̍]	eng	Wash*ing*ton	[t□]	tcl	t closure
[▯]	epi	epenthetic silence	[θ]	th	*th*in
[ɝ]	er	b*ir*d	[ʊ]	uh	b*oo*k
[eʸ]	ey	b*ai*t	[uʷ]	uw	b*oo*t
[f]	f	*f*in	[ü]	ux	t*oo*t
[g]	g	*g*ay	[v]	v	*v*an
[g□]	gcl	g closure	[w]	w	*w*ay
[h]	hh	*h*ay	[y]	y	*y*acht
[ɦ]	hv	a*h*ead	[z]	z	*z*one
[ɪ]	ih	b*i*t	[ž]	zh	a*z*ure
-	h#	utterance initial and final silence			

Figure 9 the IPA and ARPABET symbols of phonemes.

3.5 SSI pattern speech representations

Our approach to understand a speech signal can be realised by processing at two levels. The first level is word recognition by image matching of SSI patterns; the word level paves the way for writing word-level speech down in a sentence of SSIs patterns. The second level is sentence interpretation which could be achieved by SSI pattern matching also.

The speech signal can be represented in different ways (e.g. FFT and linear code prediction LCP spectra, spectrograms, fundamental frequency). Technically, the representations are derived from an FFT (which can be a Mel-scale filter bank) with various weighting schemes applied to the coefficients and measurement of their rates of change with time which augment the recognition of speech sounds. Actually, each method of acoustic speech representation has some capability to identify certain phonemes more than others, e.g. the LCP spectra are more accurate in recognising the frequency components of vowels and semivowels. In Chapter 2, we mentioned the spectrogram can be used for speech representation as an image. It is important to consider the strengths and weaknesses of features of the spectrogram's ability to represent the speech signal as SSI patterns. The spectrogram is a comprehensive method that allows examination of the dynamic changes in a speech spectrum [67]. The spectrograms can demonstrate precisely the acoustic cues and changes of a speech signal. For example, it can display accurately: stop burst in the changes of consonants, vowel frequency fluctuations, and the change between vowels to consonants etc. Dickinson *et.al.* [68] believe that the spectrogram helps us in automatically determining phonemes, because the SSI is powerful for the segmentation of speech, its labelling into categories and provides the clearest visual cues to the boundaries between phonemes. On the other hand, the SSIs do not provide precise measurements of formants of vowels that is due to the trade-off representation among time-frequency resolutions, as discussed in Chapter 2.

Historically, in the middle of the 20th century, phonetics researchers used the SSI sounds to study and identify individual phonemes, allowing a relatively simple way to analyse speech sounds without significant mathematical analysis and computer modelling [62]. Then later, Zue and Cole [69] used visual examination of an unknown utterance to label their spectrograms phonetically. The spectrogram reading experiments revealed the spectrogram as a rich source of phonetic information that can be extracted by applying clear rules.

Dennis *et al.* [20] tested 19 sentences representing data from 5 talkers as a visual examination of spectrograms. In total, the experimenters scanned a 200 word lexicon. They reported that 10% of phones were omitted, 40% of phones were transcribed only partially in terms of phonetic features and that 17% of phones were incorrectly transcribed which could be because they were beyond the spectrogram resolution.

Recognising semivowels is a challenging problem because semivowels are similar to the vowels [70]. Pinkowski [14], has shown that the spectrogram (SSI) word patterns show changes and that the accompanying transitions within individual words produce the characteristic contours and shapes that are used to identify the English semivowels (phonemes):/w y l r/. Pinkowski has used a binary image with strong contours and shapes of words and applied Fourier descriptors for characterizing the boundary of the segmented words. Steinberg *et al.* [71] have performed segmentation of voiced phonemes in order to capture the energy in phonemes associated with each formant (between f0 to f4 formant frequencies) as spots to create speech spectrograms that can be read visually by trained experts. Khunarsal *et al.* [72], recognise the word in a singing signal with background music by using the concept of spectrogram pattern matching. Dey *et al.* [73] used a spectrogram to analyse a set of five different speakers to generate a dataset, which is used in recognising speech by an artificial neural network and speaker recognition by a hidden Markov model.

All this research has in common the use of the traditional pattern matching techniques of SSI for the phonetic level, where they focus on vowels phone patterns as an approach for verifying coefficients of the speech signal. In contrast, in our work we do not look up individual SSI phones but rather we are interested in classifying the speech sound into patterns of the SSIs. These patterns are shaped by frequency transitions and fluctuations of sound groups. We believe that the embedded sound information in the patterns of the SSIs can be represented as a code, where its parameters can be defined based on image processing, and we claim that is no one has done such an approach before.

3.6 The SSI patterns can be a more useful phoneme representation

We use in this study English language phonemes for mapping into SSI patterns; however, the methods developed can be applied in other languages with the same procedures. As we mentioned, the SSI patterns form a kind of writing system. So, the method produces a more or less permanent record that can be used to represent speech. The SSI pattern system involves a mapping between SSI patterns and sounds. The expectation is that the SSI pattern is a larger duration length than the sound units, since the SSI pattern is a unit of combination of speech sounds. The finite set of SSI patterns form words, which potentially form an infinite set in any language. However, a system based on large units can be more successful than one that is phoneme based. This is because the end point deduction of a phone is still a challenging process and large units are over limiting in their influence on coarticulation. However, the large unit models could need less computation than the phone (small unit) feature models and it is easy to count them. Kirchhoff [74] concludes that such co-articulatory modelling (syllable templates) is more effective than carrying out feature based recognition. King and *et al.* [75, 76] propose a syllable model rather than phone level model by recognising the phonetic features and decoding at a syllable level by using a neural network. The aim is to allow modelling of coarticulation effects to get better recognition.

It is expected that the SSI patterns are slightly similar to the syllable structures, but the number of SSI patterns is not the same as the number syllable structures. This is because the syllable structures are not the same for different languages, but languages can share the same types of SSI patterns. Another reason is to overcome the difficulty of syllable distinction (this is critical for deciding between syllables). In contrast, the SSI patterns are easily visually distinguished.

The syllable “is a phonological units of organisation containing one or more segments” [77], and the, syllable usually contains a vowel. A word can consist of: a single, two, three or more syllables. They are of different forms: the nucleus, onset, and coda as shown in Figure 10.

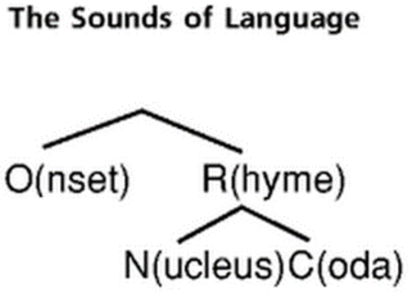


Figure 10 the syllable structure.

Typically, the nucleus of a syllable is the vowel and any following semivowels (diphthongs consist of a vowel and glide (semivowel) together); the onset is the preceding consonant, and the coda is the consonant after the vowel. Together, the nucleus and coda form the rhyme. The stress is a syllable more prominent with increased loudness and vowel length than its surroundings [77].

The stop gap (oral stop) helps distinguish the presence of a stop consonant. The sonority hierarchy is a ranking of speech sounds by amplitudes and helps in analysing syllable structures. The vowel sounds are more likely to be in the middle of words while oral stops and voiceless fricatives occur near the edges [68].

Traditional speech recognition is based on formant recognition. However, there is an inherent speech problem with formant recognition since it is not always possible to define correctly formants of fricatives or nasalised sounds and the amplitude is needed to distinguish certain phone types such as nasalised sounds and voiced vowels. That is not the full story of the speech recognition problem as many of the speech feature cues that are required for precise recognition are not able well defined. Tactically, the solution is by a sequential training process of extracted formants features (e.g. using HMMs), which also has some limitations [6].

In contrast, the spatial matching of features by SSI patterns allows the matching of whole actual speech features (formants, power spectrum, duration, cues of place articulation etc.) in one process. The spatial matching of features by using SSIs map the whole speech features but it scales down the time signal as shown in Figure 11. The time axis of the spectrogram is 41 times less than the wave time. Since the SSI patterns are a large units, they are less effected by the time resolution scale down. This could be useful for the prediction of continuous SSI patterns where determination of the time scaling is important in some ASR applications.

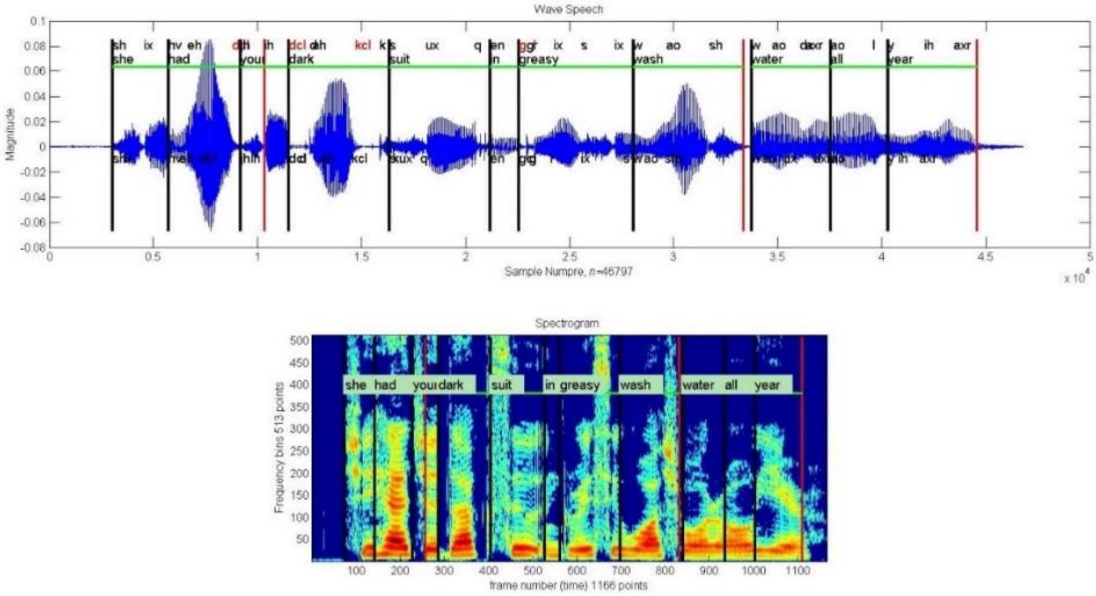


Figure 11 the time in the spectrogram is 41 times less than the wave time.

English is similar to other languages in having ambiguity (e.g. words have multiple naming, unintended meaning, and different contexts). Adding to the ambiguity, we do not have an idea of exactly how many of the SSI patterns exist, and they need to be found using a visual process initially. A question arises, do the SSI patterns have more or less permanents structures so that they can be implemented as image matching filters? Is it possible to add extra marks to SSI patterns using image processing methods to make SSI patterns richer? Not all these questions will be answered completely; rather we are going to establish essential steps for a Speech-Image Recogniser Code (SIR-CODE).

3.7 The Speech-Image Recogniser Code (SIR-CODE)

To look at our problem, it is helpful to ask the following: Is it possible to define the smallest units (segment) of the SSIs as decoding speech units to build a new speech recogniser by using image pattern recognition rather than phonemes units? Indeed, what is the entropy[11] to build this SIR-CODE? Which is kind of lossy data compression code can represent the speech wave as units?

The code parameters (symbols) can be defined by researching how the SIR-CODE parameters match the information in the speech signal to allow for recognition. The SIR-CODE entropy can be defined by counting all stabile parameters (not be affected by spectrogram resolution and speaker variations) in SSIs using image processing and classifying them into groups of patterns depending on phoneme properties. This can be estimated by using a limited number of words (small lexicon) based on the notes of expert spectrogram readers, which is done in this study. However, optimising the SIR-CODE entropy needs comparison results of SSI recognition to correlate with the-state-of-the-art techniques of HMM for a huge lexicon. Maybe, it needs to add extra symbols (indications) on some SSI patterns during image processing that can help to make the SIR-CODE a more robust code.

In other words, the SIR-CODE can be designed for different recognition purposes. In general, these may be either a word recognition by the SSI, which is a spatial matching of features of a word by SSI patterns, or prediction of continuous SSI patterns. The prediction of continuous SSI patterns can be the entirely of a speech recognition system by image processing. Therefore, the SSI patterns predictions can be achieved when optimisation of the SIR-CODE entropy is defined perfectly. The optimising of the SIR-CODE entropy aims to find as close as possible the minimum entropy of the SSI patterns that can determine how much information in the SSI patterns is needed to represent a model of speech recognition (lexicon) perfectly. We expect the

[11] Entropy is a measure of how much information observations (unpredictable information about the SSI patterns) can be used to give a certain order to a random process (a word is an ordered sequence of the SSI patterns).

code would not only work for ASR, but it could also be a new way for deaf people to learn and read, since they can then perceive speech of other people. Since children with hearing impairment learn to read words as a whole, they can recognize printed words by recoding them into a visual representation.

3.8 Defining a basic SSI pattern

Using an approach based on syllables, phonemes will be classified into SSI patterns starting from the full formant pattern structures (periodic segments of voiced utterances) to the full noise pattern structures (unvoiced). Usually, formant frequencies of a speech signal appear as dark peaks making horizontal bands in the spectrogram; the darkness reflects the reduced amplitude of a speech signal in certain frequency bands. The vowel speech sounds appear as vertical striations in the spectrogram due to the periodic nature of the glottal excitation since the majority of vowel sounds in speech are voiced, whereas the unvoiced speech sounds appear as rectangular dark patterns due to their noise-like excitation [62].

It could be premature to discuss the image processing on SSI patterns, but it is important to draw attention to the fact that all the images that have been used in this study have a noise redacted background (the noise background reduction algorithm is employed). It will be explained in detail later in section 3.12.1 of this chapter.

Nearly all lexical words have vowel sounds. The SSI (spectrogram) can resolve a vowel and diphthong (consisting of two vowels) sounds clearly into patterns containing 4 formants (or possibly 5 formants depending on the loudness and SSI resolution). These formants are strong enough to be maintained after applying an image threshold prior to application of the noise background reduction algorithm. Therefore, we call this type of SSI pattern the Full Formant Pattern (FFP), which has vertical striations due to the periodic nature of the glottal excitation [voiced speech]. Those lines are very close together and form this shape of consecutive line patterns at the lowest frequencies. A type of FFP, which is representative of certain compound sounds, has the same locations and transition of the lower three formants to form a uniquely shaped pattern. The higher formant varies slightly from speaker to speaker and can be is used for a speaker identification. Peak formant frequencies appear as dark horizontal bands in the SSI pattern. The FFP has a more stable pattern richer in features that results in each FFP pattern consisting of a unique structure. Figure 12 shows the FFP patterns in the word “need”. This has two different kinds of FFP pattern. Based on this image, it is easy to make a distinction using image processing based on the FFP, which means automatic image matching is possible too.

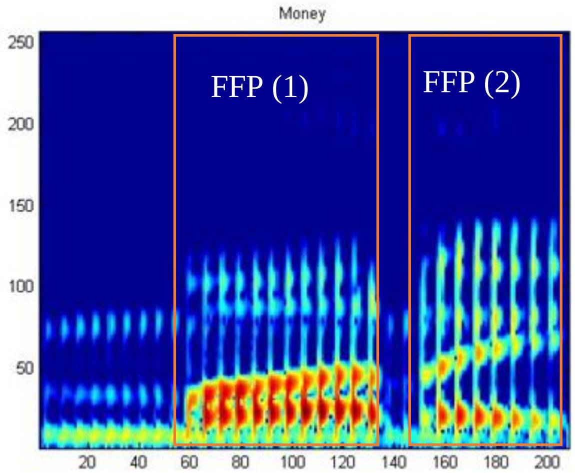


Figure 12 demonstration of the different FFP patterns in the SSI of the word “need”.

Furthermore, the formant locations, and distance between the two of them, result in the FFP having variable start and end points. Within the formant, the maximum amplitudes can be increasing or decreasing or have a flat transition as shown in Figure 13.

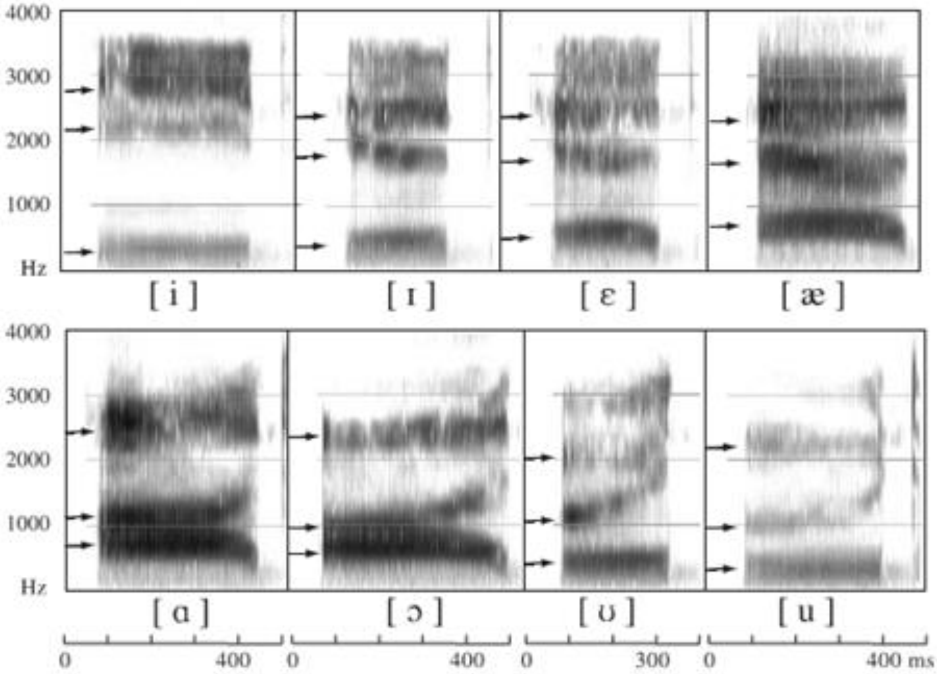


Figure 13 an SSI of the words ‘heed’, ‘hid’, ‘head’, ‘had’, ‘hood’, ‘hawed’, ‘hood’, and ‘who’d’ as spoken by a male speaker of American English. The locations of the first three formants are shown by arrows [78].

Semivowels (glides and liquids) have the same characteristics of vowels but they are much shorter in duration than vowels and the glides and liquid formants are weaker than vowel sounds [1, 78]. Nasal sounds are normally weaker in energy than vowels [78] but they are the same as vowels too. Therefore, after applying the image threshold for the noise background reduction algorithm on those pattern types which are created by Semivowels and Nasals, the patterns look like the FFPs but more faint. We call them Semi- Full Formant Patterns (SFFP). The SFFP has a formant structure similar to the FFP type, except that the bands of the 3 formants are somewhat fainter. Since the nature of the SFFP is similar to the FFP, the expectation is that the kinds of SFFP type are unique too, unless a kind of SFFP type is affected by the noise background reduction algorithm that can then generate some suspicious patterns. Figure 14 shows two kinds of SFFP. Since they are not totally extinguished and they can be recognised easily.

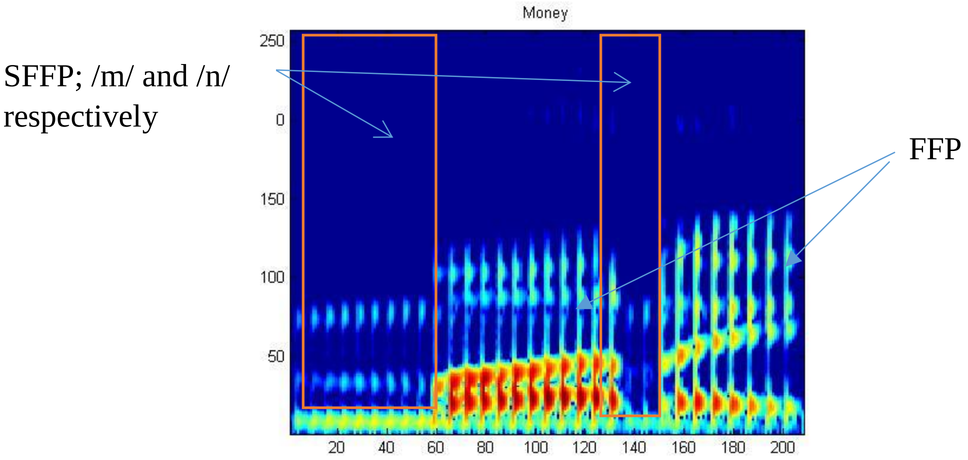


Figure 14 patterns of the word “money”, showing SFFP and FFP patterns.

The FFP and SFFP can be called SSI Format structures (SSIFS). The SSIFS shares a common parameter which is a high PSD strength at the first formants. The first formant is located in a small dynamic range location, the average first formant locations for vowels sounds (/IY/, /IH/, /EH/, /AE/, /AA/, /AO/, /UH/, /UW/, /AH/, ER/) being: 270, 390, 530, 660, 730, 570, 440, 300, 640 and 490 Hz, respectively [1]. The average first formant location for vowels sounds is 502 Hz. Phonetically, the FFP and SFFP are under a nucleus syllable structure. The nucleus syllables occur more frequently than other syllables. Usually, an SSIFS contains either of an FFP or combination of FFP and SFFP; thus the SSIF is inevitably a unique form as it contains a unique kind of FFP.

Consonants (expect Nasals sound) are responsible for random noise pattern SSIs. The random patterns are spread in different ways; starting from an upper limit and extending down to different locations in the middle of the SSI, or scattered from the middle to the lower part of the SSI, centred on the middle or scattered in the whole pattern, depending on the unvoiced sound type. In Fricative and Affricates, the random noise pattern is in the higher frequencies (6 kHz), and extending to different regions of the lower frequencies, depending on the place of articulation for the fricative sound (e.g. in /SH/ this extends to about 2.5 kHz and /Z/ and /F/ extends to a lower part of the frequency scale less than 0.5 kHz) [18]. With stops in sound, the random noise patterns are in the lower frequencies and extend to different locations of higher frequencies depending on the place of articulation. Usually, the duration of stops are shorter sound patterns (even a sharp pattern) and weaker (with a fainter PSD) as compared to both Fricative and Affricate patterns. Therefore, stop patterns are more likely to be affected by the image threshold of the noise background reduction algorithm. Some patterns are common which are clearly absent of formant patterns. Therefore, we call them Absent Full Formant Patterns (AFP). They appear as rectangular dark patterns (scattered spots appearing as a noise structure) with wide or narrow durations, centred and extended in different locations. There are also random occurrences of weak oscillations due to sudden variations in energy (unvoiced speech). In general, the AFP patterns can be classified into two types of AFP: Short duration Absent Formant Patterns (SAFP), as shown in Figure 15, and Long duration Absent Formats Pattern (LAFP), as shown in Figure 16.

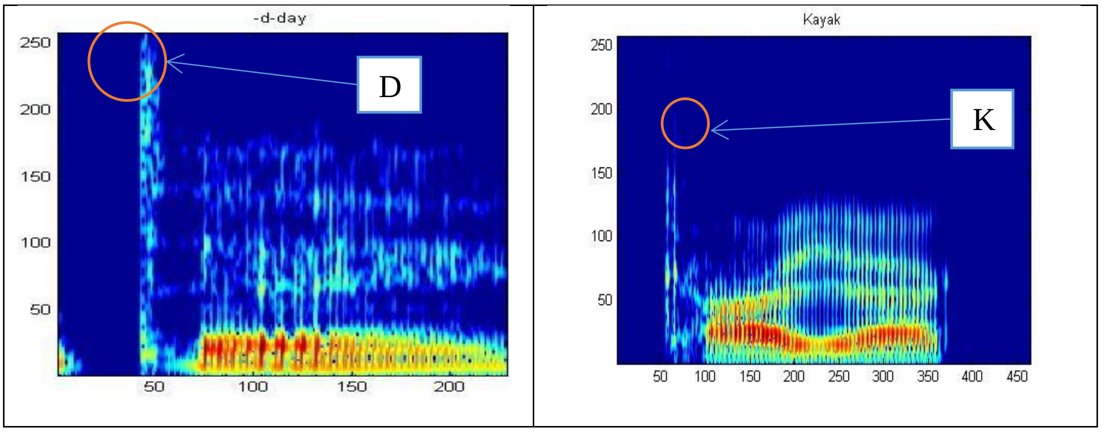


Figure 15 examples of the SAFP.

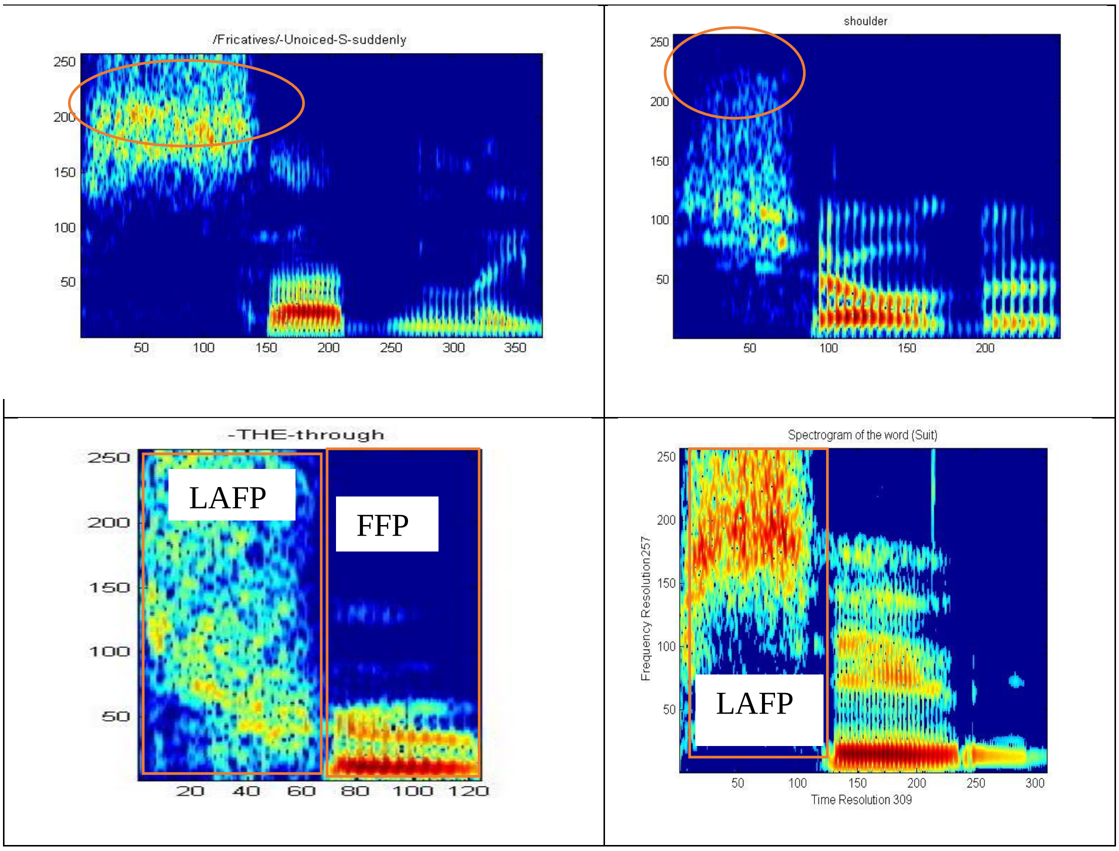


Figure 16 example of the LAFP.

The high intensity power spectra of the random patterns (spots) are less impacted by threshold processing of the noise background reduction algorithm applied to the SSIs. Therefore, the remainder of the spots are scattered around different centroids, mostly concentrated on the top, centre, bottom or whole rectangular AFP that can be used to classify the AFP types into subtypes for both the SAFP and LAFP forms.

Moving from systematic patterns to noisy patterns, we pass through of a hybrid of systematic and noisy patterns which we call a nested pattern. The nested pattern is created mostly by the Whisper sound /HH/; some textbooks count it as a consonant sound rather than a vowel [1], since its characteristics fluctuate between a formant structure and a random structure depending on its position in a word. As an example, Figure 17 shows the SSI of the word ‘Higher’, which contains two sounds of /HH/ at the start and the middle. The sound /HH/ at the start is like an AFP pattern while in middle more like an FFP pattern.

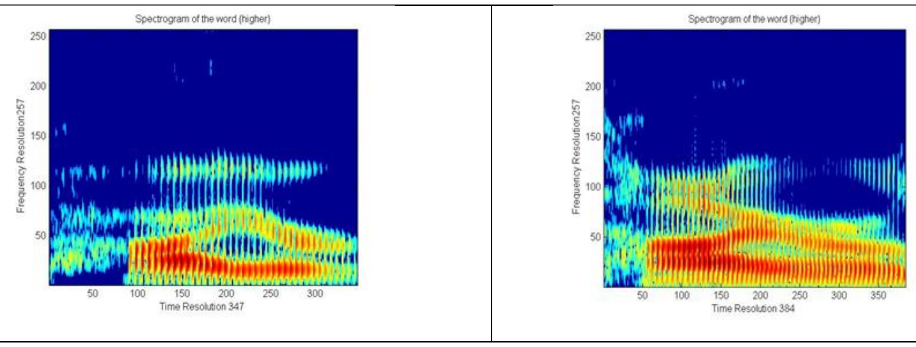
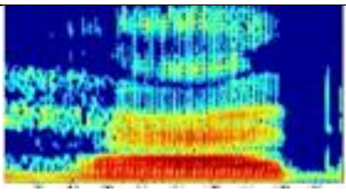
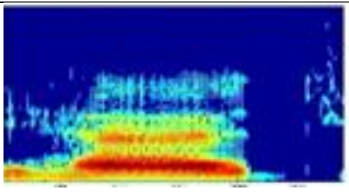
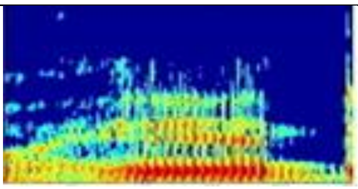
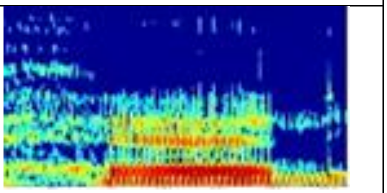
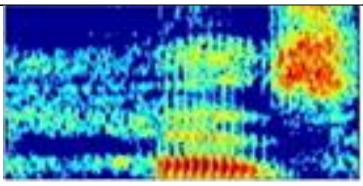
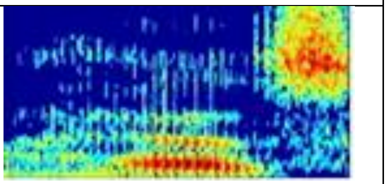
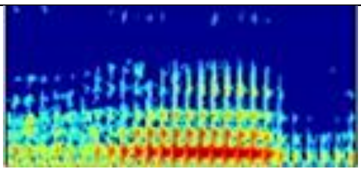
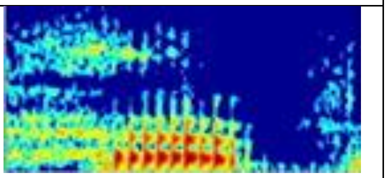


Figure 17 shows the SSI of word ‘Higher’.

In some cases, Nasal sounds are affected by surrounding sounds to give nested patterns too. The nested patterns do not have stable features and tend to random structures more than formant pattern structures as shown in Figure 18. Anyway, for more simplification of the recognition problem of SSI patterns, nested patterns can be supposed to be in either the AFP or FFP category.

Had		
Hand		
Has		
Have		

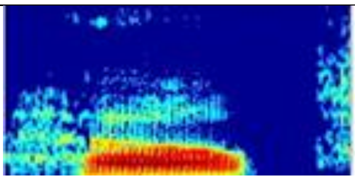
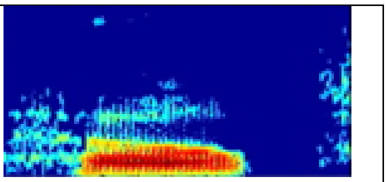
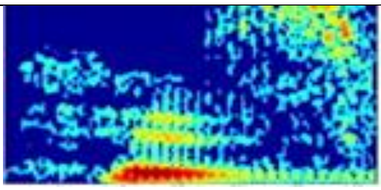
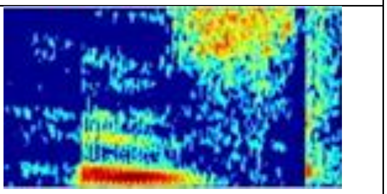
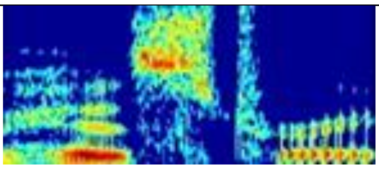
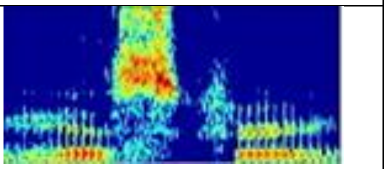
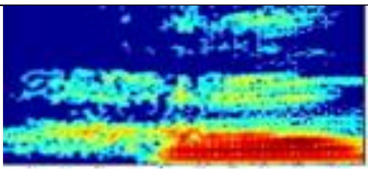
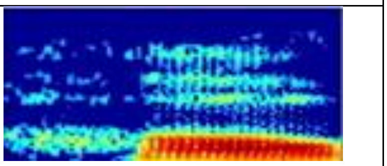
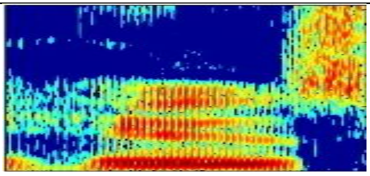
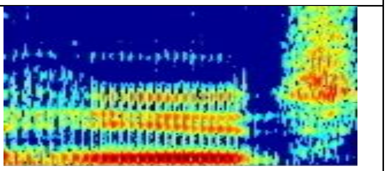
Help		
His		
History		
How		
Huge		

Figure 18 shows nested patterns produced by the sound /HH/ in different words.

Gap patterns are created by the stops sound (voiceless/P, T, K/ and voiced / B, D, G/). This sudden explosion and aspiration of the air characterizes the stop consonants [1, 78]. As we mentioned previously, the fraction of unvoiced stops is longer than voiced stops. Usually, the gap is followed by the noise of unvoiced stops [78] as shown Figure 19.

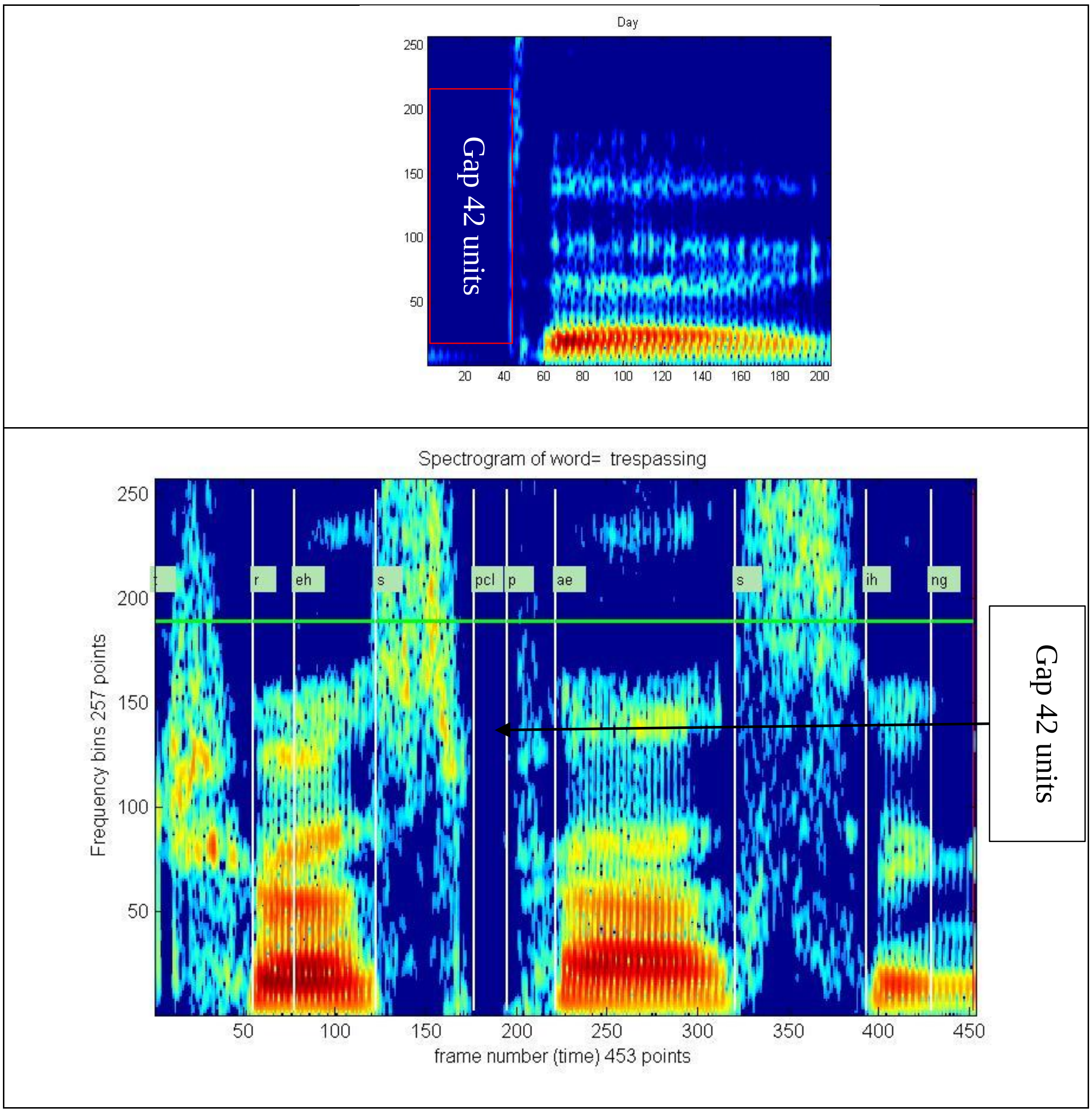


Figure 19 shows gap pattern examples.

All the English phoneme classes (vowels, diphthongs, semivowels, stops, Fricatives, Whisper, Affricatives, and nasals) have been scanned and identified using comments of expert speech spectrogram readers. However, that does not mean all the possible parameters have been defined. Further, a deeper study of the SSI patterns within the phone level can be used to define sub-SSI patterns that can make the SIR-CODE richer still.

An example is that of classifying the word “need” into SSI patterns which are: SFFP, FFP, G, and AFP is shown in Figure 20. All SSI patterns in this word can be easily identified by eye. It is also easy to distinguish between the SSI “need” and all the

SSIs in this study. Therefore, treating the SSI patterns as images it is possible to design matched filters based on the SSI patterns parameters for recognising such words.

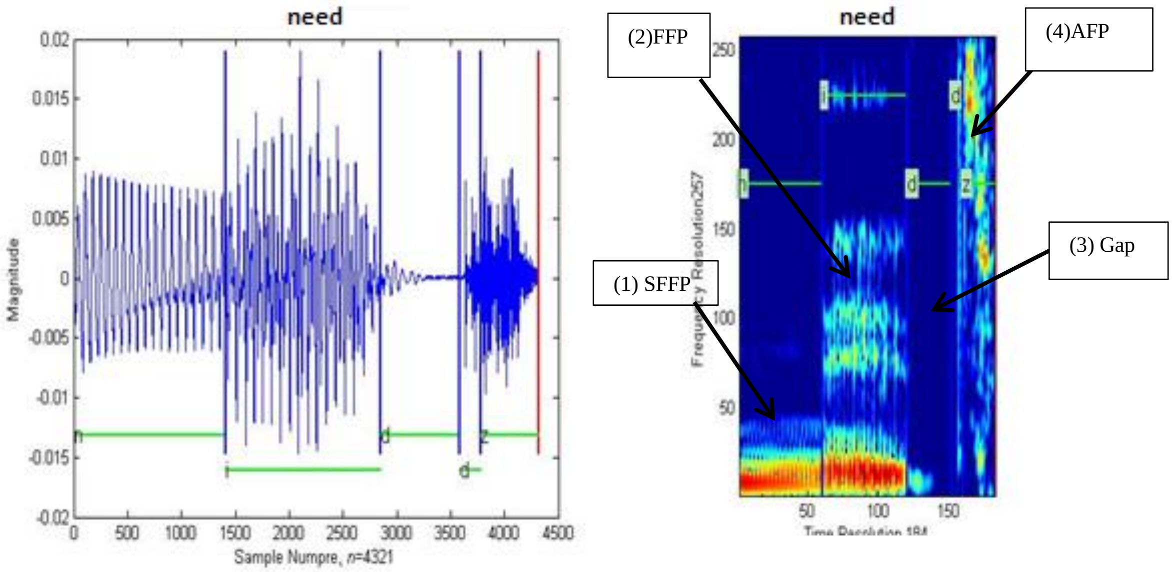
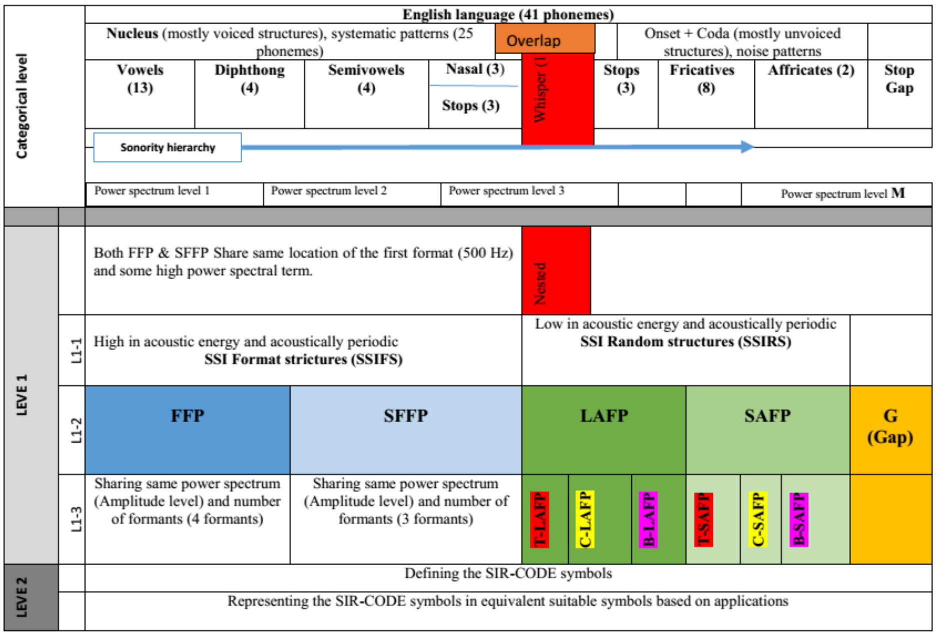


Figure 20 demonstration of the 4 main SSI patterns in the word “need”.

To do so, it is important to summarise the basic SSI patterns in a table, which helps in designing the SIR-CODE for different recognition levels of complexity and purposes. In general, we divide the process into two levels: level 1 is word recognition by SSI, and level 2 is prediction of continuous SSI patterns. Table 2 includes parameters such systematics, randomness, power spectrum sonority, duration, and gap.

The amount entropy in these basic parameters is unknown, and it is unknown how many symbols can represent each pattern to create the optimal SIR-CODE entropy. Thus, we have to test recognition ability of single examples, as well as permutations and combinations, of the SSI patterns by image matching in order to define the SIR-CODE symbols. As an example, a kind of FFP type symbol can be defined by estimating its probability of occurring in a certain lexicon (of limited number of words). While estimating permutations (order patterns) and combinations (group parameters), the probably of patterns can define entropy symbols for a word code in that lexicon.

Table 2 basic parameters of SSI patterns.



<table>
<tr><td colspan="2" rowspan="5">Categorical level</td><td colspan="9"><b>English language (41 phonemes)</b></td></tr>
<tr><td colspan="4"><b>Nucleus</b> (mostly voiced structures), systematic patterns (25 phonemes)</td><td>Overlap</td><td colspan="3">Onset + Coda (mostly unvoiced structures), noise patterns</td><td></td></tr>
<tr><td><b>Vowels (13)</b></td><td><b>Diphthong (4)</b></td><td><b>Semivowels (4)</b></td><td><b>Nasal (3)</b>
<b>Stops (3)</b></td><td>Whisper (</td><td><b>Stops (3)</b></td><td><b>Fricatives (8)</b></td><td><b>Affricates (2)</b></td><td><b>Stop Gap</b></td></tr>
<tr><td colspan="9">Sonority hierarchy</td></tr>
<tr><td colspan="2">Power spectrum level 1</td><td colspan="2">Power spectrum level 2</td><td colspan="2">Power spectrum level 3</td><td></td><td></td><td>Power spectrum level <b>M</b></td></tr>
<tr><td></td><td></td><td colspan="9"></td></tr>
<tr><td rowspan="4">LEVE 1</td><td></td><td colspan="4">Both FFP &amp; SFFP Share same location of the first format (500 Hz) and some high power spectral term.</td><td>Nested</td><td colspan="4"></td></tr>
<tr><td>L1-1</td><td colspan="4">High in acoustic energy and acoustically periodic
<b>SSI Format strictures (SSIFS)</b></td><td colspan="4">Low in acoustic energy and acoustically periodic
<b>SSI Random structures (SSIRS)</b></td><td></td></tr>
<tr><td>L1-2</td><td colspan="2"><b>FFP</b></td><td colspan="2"><b>SFFP</b></td><td colspan="2"><b>LAFP</b></td><td colspan="2"><b>SAFP</b></td><td><b>G (Gap)</b></td></tr>
<tr><td>L1-3</td><td colspan="2">Sharing same power spectrum (Amplitude level) and number of formants (4 formants)</td><td colspan="2">Sharing same power spectrum (Amplitude level) and number of formants (3 formants)</td><td colspan="2"><b>T-LAFP</b>
<b>C-LAFP</b>
<b>B-LAFP</b></td><td colspan="2"><b>T-SAFP</b>
<b>C-SAFP</b>
<b>B-SAFP</b></td><td></td></tr>
<tr><td rowspan="2">LEVE 2</td><td></td><td colspan="9">Defining the SIR-CODE symbols</td></tr>
<tr><td></td><td colspan="9">Representing the SIR-CODE symbols in equivalent suitable symbols based on applications</td></tr>
</table>

After defining the SIR-Code entropy symbols, this can be used to design entropy coding (lossless coding) of the coder and decoder for the SIR-CODE. In other words, the SIR-CODE symbols can be represented in different symbol styles for use in applications either in writing for the deaf or in ASR. Figure 21 shows a basic SIR-CODE for use in ASR by employing a barcode for representation of the speech signal.

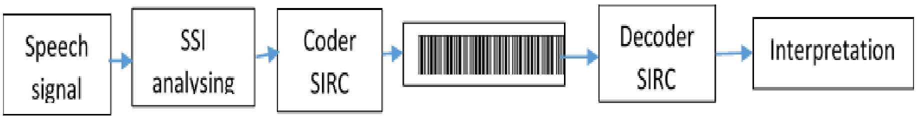


Figure 21 shows a basic SIR-CODE employed in ASR.

All spoken languages have vowels and consonants. The same linguistic definitions can be used to analyse vowels and consonants in all languages. The perception of the difference between a vowel and a consonant, however, may vary quite a bit. Even when two languages have similar sets of phonemes, other factors come into play, such as stress or intonation, length of phonemes, rhythm, speed, etc. The central part in our work is that all spoken languages have SSIFS and SSIRS patterns. This means all languages share the classification shown in Table 2.

3.9 Missing phonemes in SSI patterns

Missing phonemes in spectrograms have been reported, but never defined. Dennis *el at.* [20] have reported that 10% of phonemes in their test sample were omitted. That could be because of spectrogram reader inefficiency or coarticulation phenomena (phonemes are speech units including articulation information and so represent individual sound or speaker variations which are affected by spectrogram resolution (resulting in some sound parameters becoming hidden). Of course, this percentage increases somewhat after applying the noise background reduction algorithm, which is an essential step for SSI matching.

In some ways, the missing phonemes can be regarded as embedded into SSI patterns. Actually, we rely on only stable SSI patterns. Add to that, words are multi SSI patterns. So, it is expected that the entropy of stable patterns and their order can provide enough information (entropy) for recognition. We have mentioned that the SAFP classes are more likely affected by the threshold for the noise background reduction algorithm. On the other hand, the G pattern is evidence of the SAFP existence (usually G is followed by the SAFP). However, the use of the noise background reduction algorithm partially eliminates the SAFP, if not all of it. The G pattern is evidence that can fill the gap of missing information.

As an example, Figure 22 shows 18 samples of the SSIs of the word 'Dark' (which contains stops sounds /D/, and /K/). These samples belong to different genders of speakers and have had applied to them the same level of reduction of background clutter. The 'Dark' word can be classified it into three SSI patterns type: SAFP, FFP, and G, in the order SAFP, G, FFP, G and SAFP. Although the 18 samples of the SSI of the word 'Dark' are disparate, it can be easily determined from them.

As another example, this time for the stop sound /T/ which is been shown in 18 samples of the word 'Suit' in Figure 23. The word 'Suit' consists of LAFP, SAFP, FFP, and G, SSI patterns in the order: LAFP, FFP, G and SAFP. Both SSIs of the words 'Dark' and 'Suit' have one type of FFP pattern, but different classes of FFP. Obviously, it is easy to distinguish between ‘Dark’ SSIs and ‘Suit’ SSIs visually, so that we can look to invest in this ability of visual recognition in the image matching process.

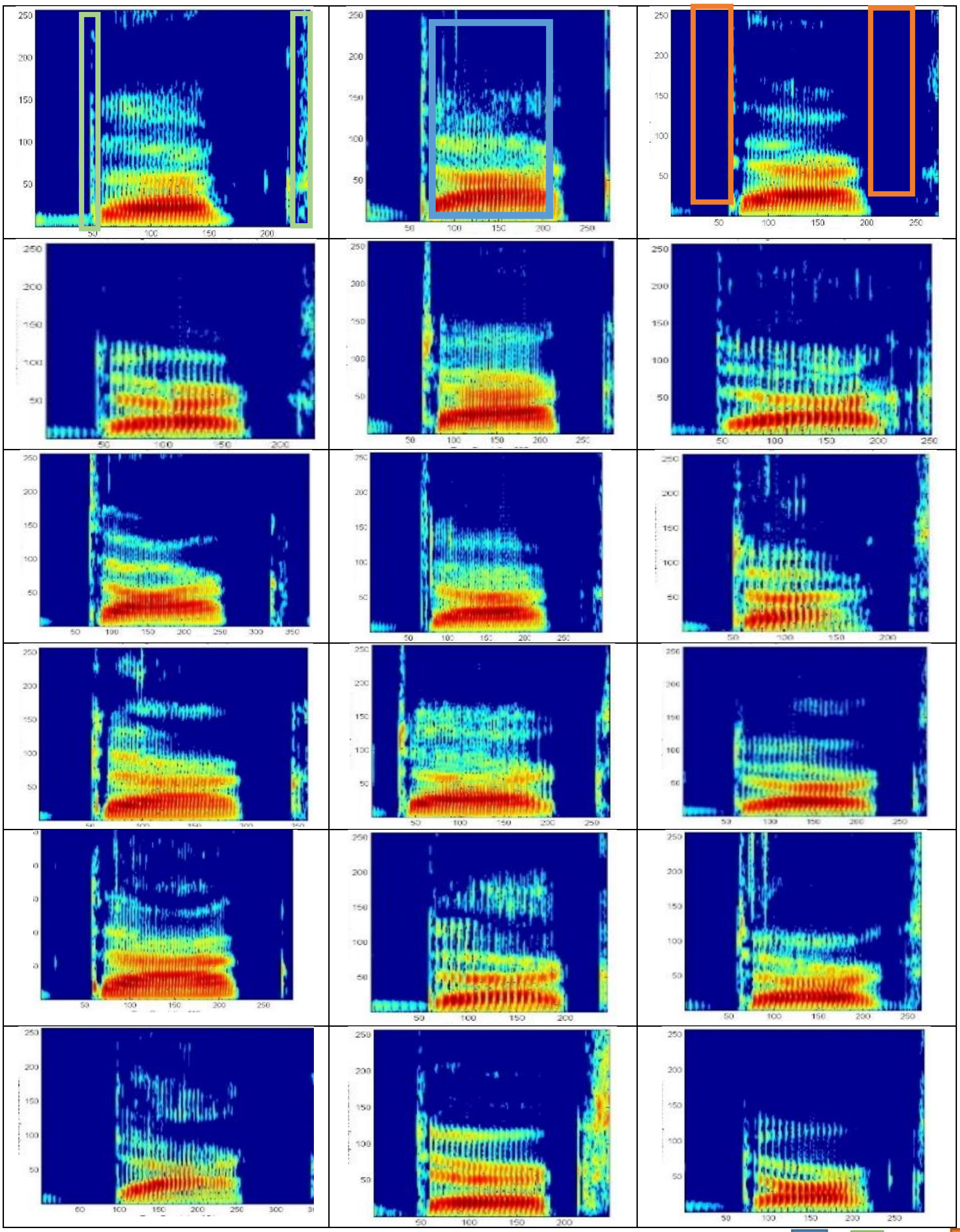


Figure 22 shows variant SSI samples of the word “Dark”, where; 🟦 🟩 and 🟧 are the FFP, SAFP and G patterns, respectively.

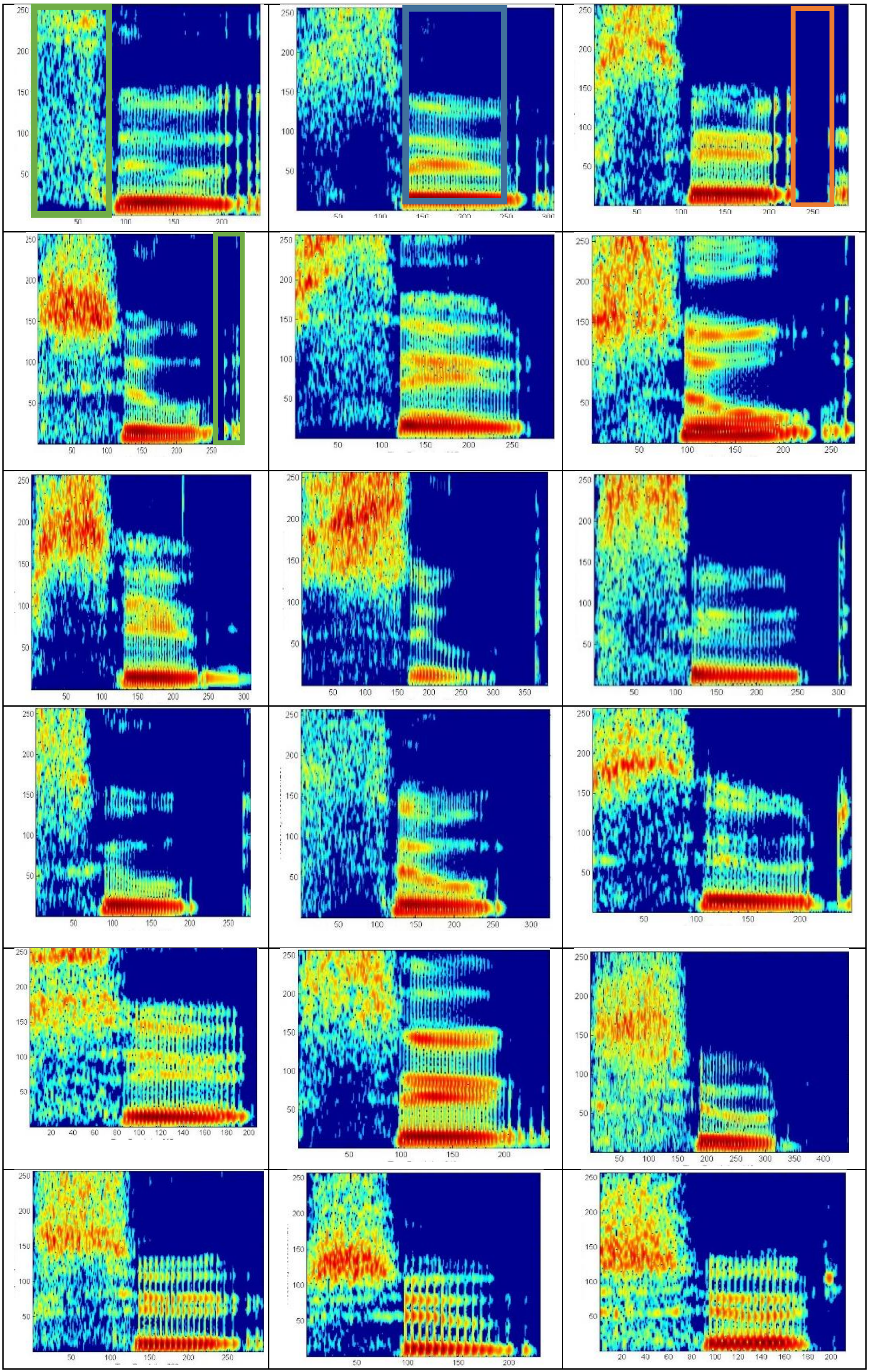


Figure 23 shows variants SSI samples of the word “Suit”, where; 🟦 🟩 🟩 and 🟧 are the FFP, SAFP, LAFP and G patterns, respectively.

The database that has used in this study is the TIMIT[12] database, and it should provide a standard signal to noise power ratio. However, studying case of missing phonemes in SSI patterns is a challenge due to threshold problem of the noise background reduction algorithm. Nevertheless, this can be optimised and will provide more information about SSI patterns. This can be done when all the types and kinds for a certain lexicon are defined. Surely, then we can make the SIR-CODE entropy richer.

3.10 Distortion problems of SSI patterns

The SSI represents a speech signal in three dimensions: frequency, time and power spectrum intensity. In general, all SSI patterns are governed by speaker variations. The speaker variations, e.g. realisation, speaking style, the gender of the speaker affecting the anatomy of the vocal tract, speed of speech, regional and social dialects etc., have the strongest effect in the time dimension rather than the frequency dimension of the SSIs.

We have mentioned that nucleus syllables are occur more frequently in speech. Also, their format structures are suitable for image matched filtering than the random structures of SSIs. The SSIFS of the same word for different speakers (including different genders) is effected by frequency and time dimension distortion more than variations in pixel contrast (which is due to variations of speaker power in the speech). Only exact position of the higher formants is slightly varied from speaker to speaker, while the low frequencies are the same for the same word [62, 79, 80]. On the other hand, the vertical distortion (frequency dimension) in the SSIFS is due to the pitch of the voice. The pitch is due to the parodic nature of glottal excitations (the fundamental frequency) of the voice and appears as vertical striations in the SSIFS. Figure 24 demonstrates the pitch distortion on the SSIFS by displaying two SSI samples of the words ‘Murky’ and ‘Dark’, where they belong to different genders and speakers. Usually, a female’s pitch (bound of frequencies) is higher than the male, and appears to

[12] TIMIT is a database designed to further acoustic-phonetic familiarity for use in automatic speech recognition systems. For more information, see, https://catalog.ldc.upenn.edu/LDC93S1.

generate more sharp vertical striations than the male voice in the SSIFS. Although, there is a slight vertical distortion in the SSIFS, there is still a big correlation between the two different SSIFSs of the same class in an SSI word. Thus, the vertical distortion may not be a serious problem for the image matching.

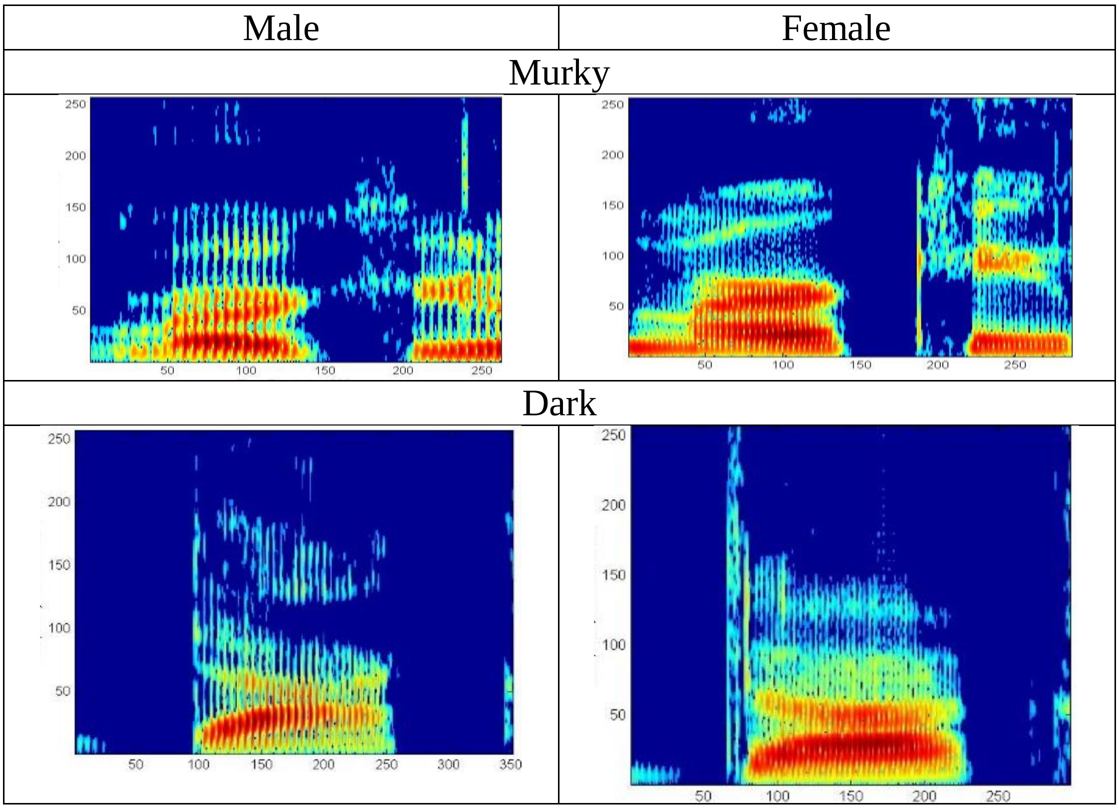


Figure 24 shows pitch voice distortion in the frequency dimension of the SSI.

However, the SSIFS may be distorted badly by the time dimension change (i.e. how fast a word is uttered). The duration has the effect of stretching or shrinking some or all the SSI patterns, and ruins their positions within SSI too (there are different start and end positions for almost the same durations of two utterances of the same words). Figure 25 shows the time variation in the SSI pattern of the same word for different speakers. It is obvious that the individual SSI pattern has different durations and location within the same SSIs for utterance of the word ‘Dark’ by different speakers.

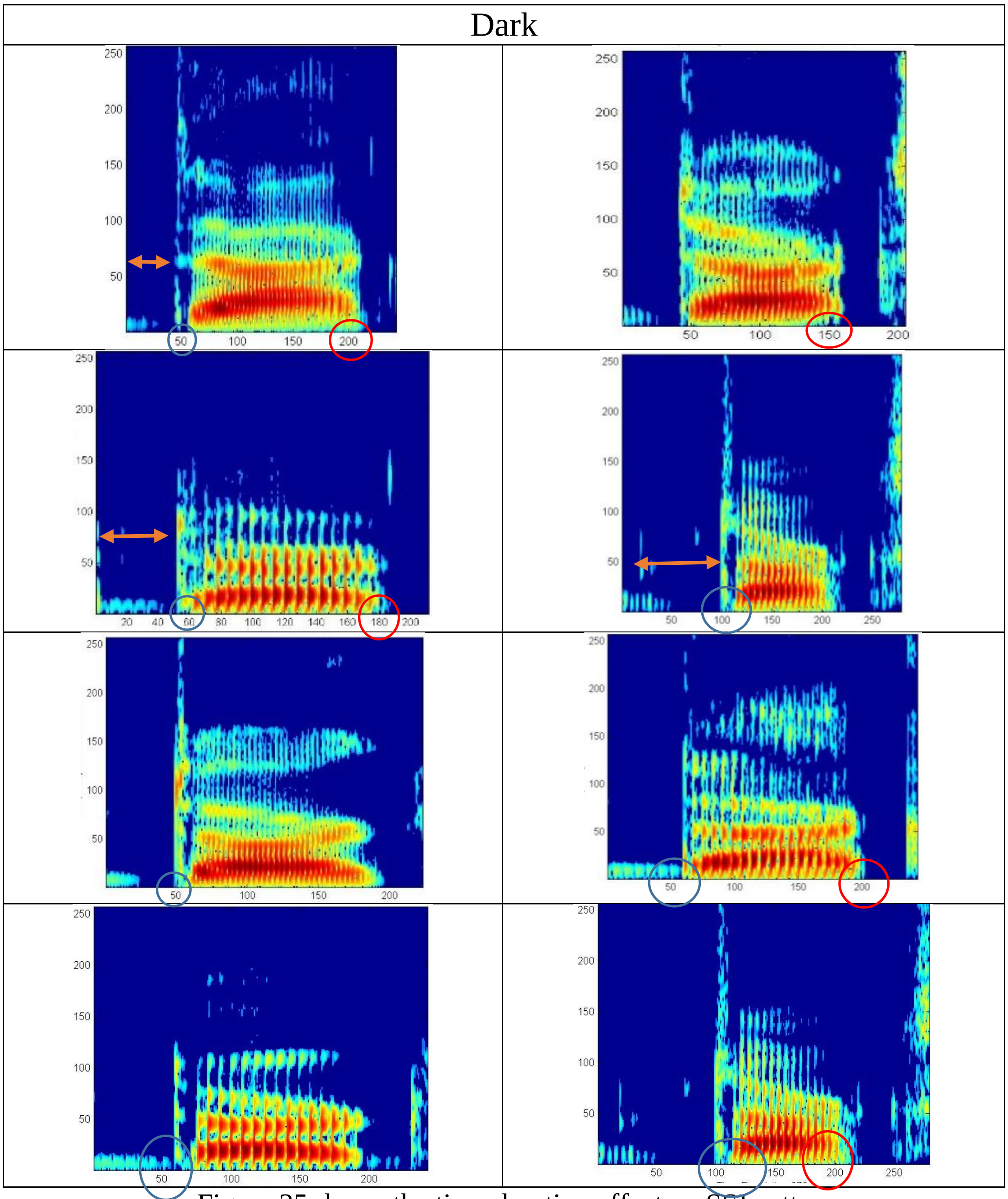


Figure 25 shows the time duration effect on SSI patterns.

However, the differences among SSIFSs result in some distortions. The techniques derived from invariant pattern recognition can accommodate these distortions. We have used normalized length durations of the SSIFS patterns. Word recognition by using SSI patterns has been shown in our work [81] and most details of it will be explained in Section 3.12.1 of this chapter.

The PSD of speech is the third piece of information provided by the SSI, which is represented by pixel intensity values. In a simple form, image matching is accomplished by pixel intensity matching. However, when there is variation in speaker loudness, the SSI patterns of the same word by speakers have some range of power spectrum disruption. That can be determined clearly from all the SSI figures (of the same word) that are provided in this study.

Because of the particulate nature of the FFPs (which is the mean of the SSIFSs), they have a higher average power spectrum than other SSI patterns. Therefore, the SSIFS is less effective than the SSIRS due to variation of the power spectrum of speakers. Actually, the correlation measurement of SSI patterns is affected badly by the SSIRS due to the random nature of it. On other hand, the SSIRS type of data make the regularities of shape help pattern recognition.

3.11 Example of using basic parameters of SSI patterns table for design case L1

We use Table 2 is to select common parameters or combined parameters. It is used to help to define the SIR-CODE entropy for certain purposes. This, then should be implemented in image analysis and mathematical morphology design. The level 1 can distinguish the SSIFS types among the SSIRS and gap types. The SSIFS entropy is a common parameter between the FFP and SFFP which is the approximate location of the high power spectrum of the first formant. The complementary information is the SSIRS. Level 1 may be enough for recognition of isolated SSI words.

In fact, SSIFSs are dominant in the English language due to their variety and probabilities of occurrence. Beside, a word that is a multi SSI type has at least to contain one type of SSIFS (it is supposed to be a unique type). Consequently, recognising a particular type of SSIFS among SSI patterns by using image matching may provide enough entropy for recognition of a word by its SSI in a certain lexicon. Recognition by SSI may fail due to some difficult types of SSIFSs being unrecognisable by using image processing procedures. Then we must consider including higher levels of L1 (e.g. level 1- 2 of the SIR-Code in Table 2) to increase the entropy. Therefore, identifying the type of SSIFS by image matching is a first priority. This guesswork has to be tested by image analysis and mathematical morphology. Then, we implement higher levels until we define perfectly the SIR-CODE entropy that can help to make SSI pattern prediction applicable.

The image matching methods are grayscale based and edge-based matching. The SSI patterns are not constant; they have many edges that are disrupted from exactly symmetric shapes. Besides, they are multi-variate in intensity contrast due to variable loudness of speakers. However, they generally have an average intensity contrast. Therefore, it may be that the correlation function works well for recognising SSI patterns. Add to that the correlation results can easily interpreted by eye which can help to define some naive procedures for SSI pattern recognising.

3.12 Some naive procedures for SSI pattern recognition

Matching techniques fall into two broad categories: area based matching (ABM) and feature based matching (FBM), respectively. The correlation and the least squares matching approach are well known methods for ABM. FBM determines the correspondence between image features and does not require very precise initial estimates. The correlation and the least squares matching approach are well known methods for area based matching in image recognition. Normalised cross-correlation (NCC) is a basic but effective method to provide a similarity measure and is often adopted for similarity measurements due to it is good robustness [82]. The NCC is a grayscale based matching method and is affected by variable intensity contrast of the object. Figure 26 shows a naive sample of the SSI (without any enhancements). The background is full of clutter, which is random in location and varying in intensity contrast which affects badly the NCC result.

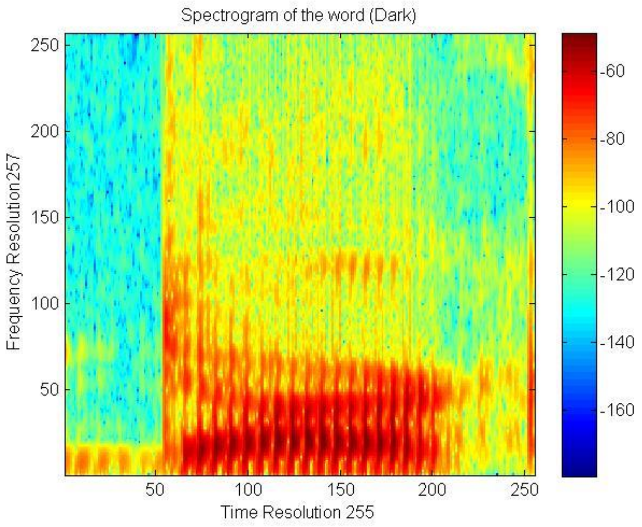


Figure 26 the SSI of the word ‘Dark’ without noise background reduction.

To maximise the NCC between the SSI pattern templates and the input image (to which the SSI is being matched), the variation in the SSI must be minimised, if possible by some form of pre-processing operation on the raw spectrogram data. Many of the disruptions to spectrogram images are related to clutter in the raw images and recognition improvements depend on how much this can be reduced. However, in addition, the speech signals of words which are uttered by different persons give rise to spectrograms with only quasi common shapes. If these shapes can be made more similar by appropriate pre -processing, the resulting NCC between the SSIs from different speakers will increase.

3.12.1 Reduction of Clutter in the SSI

Speech-only excerpts are difficult to obtain as we rarely encounter real-world segments with no noise. The spectrogram is usually accompanied by different forms of noise, including those formed during sound recording [83], and those produced during the transformation to the frequency domain which result from spectral leakage when the power spectra are computed. Thus the resultant spectrograms contain important sound patterns of the signal immersed in contaminating noise and disrupted by artefacts generated during discrete Fourier transformation to the frequency domain. Therefore, cleaning the signal in the time domain will not ensure a completely pure spectrogram [84].

3.12.2 The noise background reduction algorithm

The noise contaminating the spectrograms can have almost the same power level as the weaker power spectrum formant features at the same frequency which when removed from the whole spectrogram will thus necessarily eliminate the weaker formant patterns located at higher frequencies. However, this effect is not very critical since recognition depends largely on the high power lower frequency formant components of the SSIs. The dynamic range is limited to - 40 dB below the maximum value for all tested sounds [75]. Therefore, any points with a power value outside this range, that includes the noise as well as very weak formant patterns, are eliminated from the spectrogram. Figure 27 shows the same word sample in the naive SSI shown above in Figure 26, but with noise below -55dB eliminated, while with the threshold set at -40 dB the SSI has the appearance shown in Figure 28.

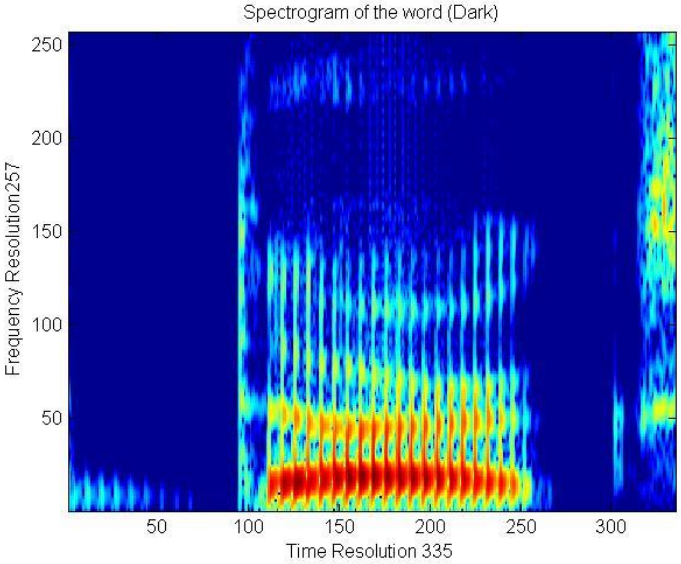


Figure 27 shows the SSI of the word 'Dark' with noise background threshold of -55dB.

Figure 28 shows that the AFT types are affected partly while the formant structures (type of SSIFS in this example of the SSI of the word 'Dark') are not affected and are more obvious. Importantly, the type of SSIFS forms a word identity. Consequently, the -45 dB threshold level for noise reduction has been used in this study.

The SSI is relatively stable in the presence of high levels of background noise and reverberation [85], which provides a significant improvement in performance on highly reverberant speech for ASR .

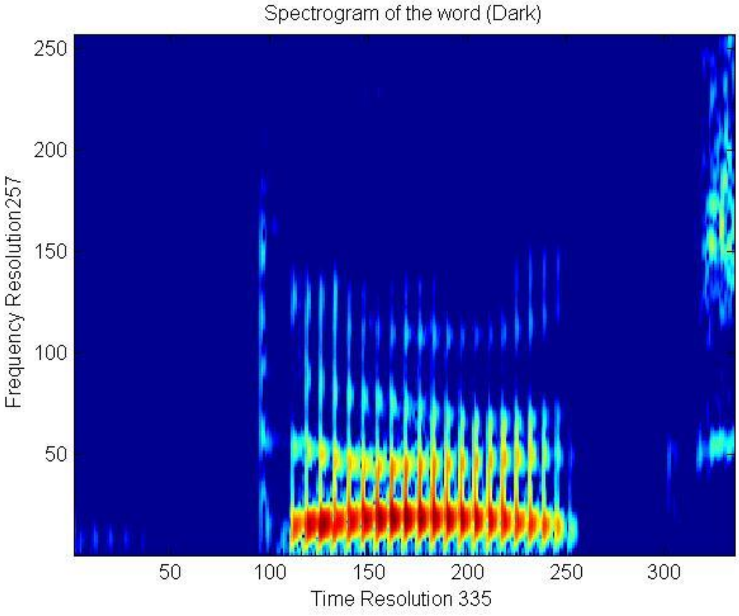


Figure 28 shows the SSI of the word 'Dark' with noise background threshold of -40dB.

Matching the two SSIs ('dark') is shown in Figure 29(a) and (b), in which the latter SSI is wider than the former but not sufficiently so to disrupt the NCC process significantly. However, the matching operation results in only part of the greater width image being matched with the smaller width image as shown in Figure 29(c). The NCC result is displayed in Figure 29(d). The NCC peak position is calculated with respect to the first line of pixels of both images (the template and input images are the same height), so the displacement of the maximum point of the NCC gives the starting position displacement of the matching area between the greater width image and the smaller width image. After clutter reduction, the NCC value is increased from 0.80771 to 0.82757 as shown in Figure 29(d) and Figure 30(d) the reduction from the normalised value being due to inexactly matched durations of the two SSIs in this example but demonstrating an effective match can be made despite this. In the latter figure, the position of the correlation peak is shown to be unaffected by the noise removal process.

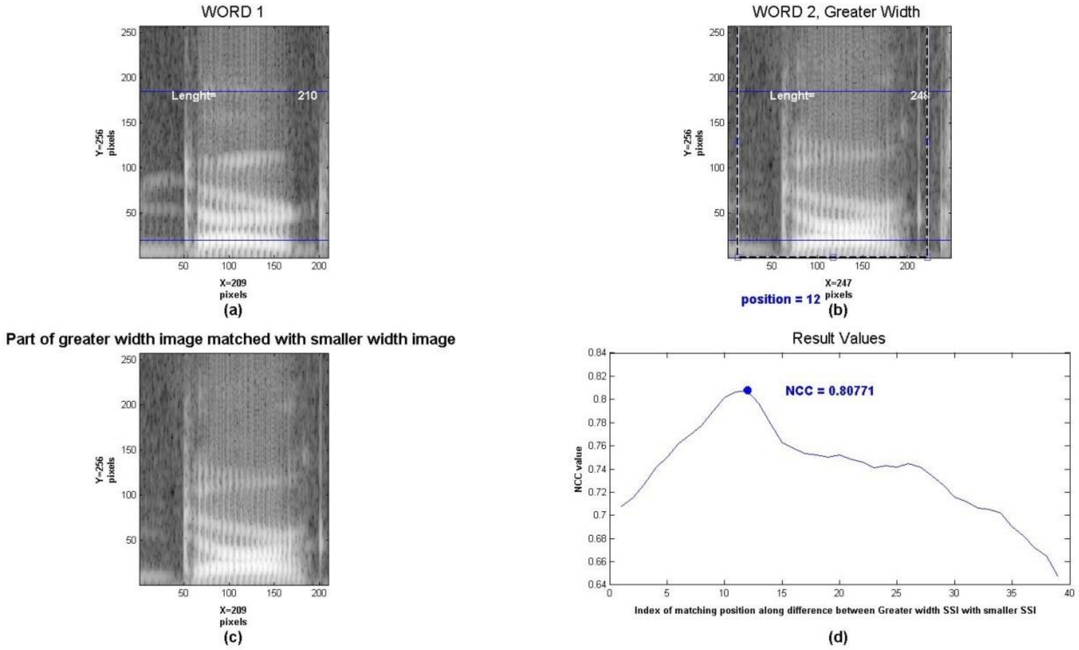


Figure 29 shows matching between two SSIs without clutter reduction: (a) Smaller width SSI (of the word 'Dark'); (b) Greater width SSI (of the same word uttered by different persons); (c) The part of the greater width image matched with the smaller width image; and (d) Display of the NCC value and the position of matching of the greater width image with the smaller width image.

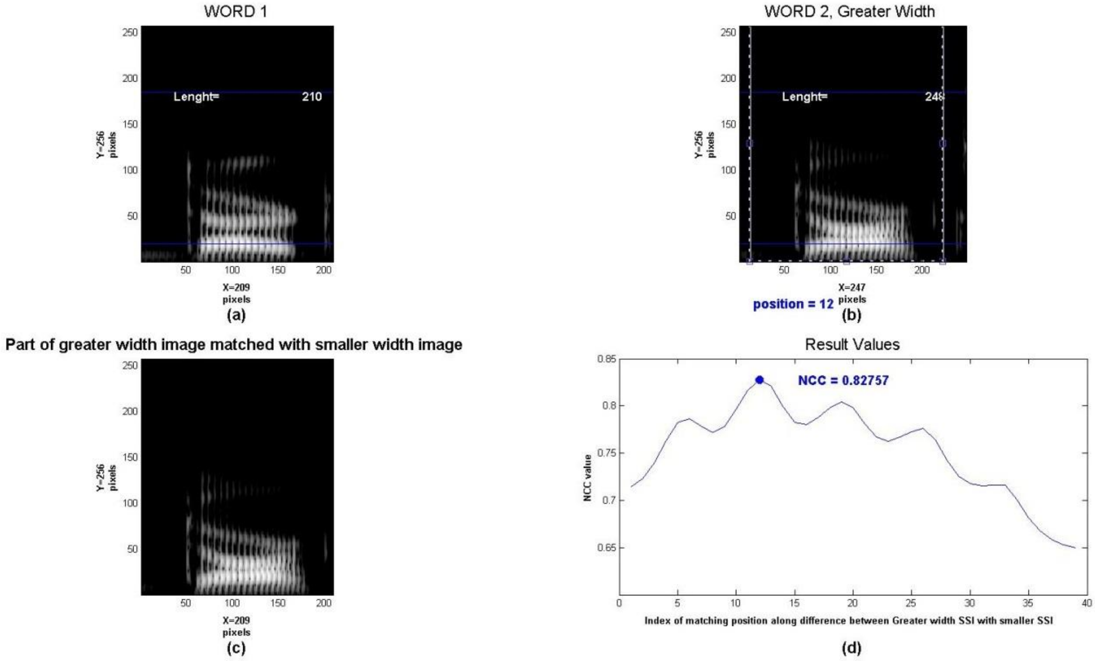
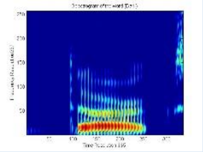
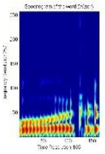
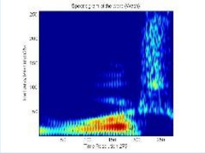
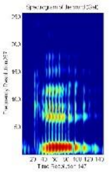


Figure 30 shows matching between two SSIs without clutter reduction: (a) Smaller width SSI (of the word 'Dark'); (b) Greater width SSI (of the same word uttered by different persons); (c) The part of the greater width image matched with the smaller width image; and (d). Display of the NCC value and the position of matching of the greater width image with the smaller width image (unchanged from the position in the unfiltered SSIs).

To show the reliability of the clutter reduction, the same procedure was repeated for matching of two SSIs of others words, as illustrated in Table 3.

Table 3 shows the increase of the NCC after clutter reduction of the SSIs of the words indicated

Two SSIs compared for the words shown	SSI:	NCC, before reducing the image clutter	NCC, after reducing the image clutter
Dark		0.80771	0.85813
Water		0.81826	0.87309
Wash		0.79392	0.8195
Get		0.70692	0.80723

3.12.3 Normalising the time dimension of the SSI based on increasing NCC value (overcoming speaker variation or speaker style)

The NCC is well known to be sensitive to differences in the scaling between the reference image (template image) and the matching pattern (input image) ; thus SSIs should be, ideally, normalised in time duration before matching is attempted (since of course, SSIs have variation in time duration).

The SSIs are not restricted by source or sensor specifications, but are created from these by an analytical transformation. Thus the resolution of the SSI, in both frequency and time, is controllable. We choose a recommended image resolution that is easily recognised by an expert human reader of sound spectrogram images. Thus the SSI resolution that we use in this study has a size of 256 × (var) pixels which is fixed on the vertical (frequency representation) axis but dynamic along the horizontal (time) axis. This is because the time representation depends on the time duration of the uttered words which is related to the speed of speech and the number of word phonemes included in the speech utterances, and so varies accordingly.

The frequency ranges within any particular word which is uttered by different people have relatively equal position of the low formants and slightly different positions in the high formants (which we have previously mentioned). The first formant in the SSIF types have very similar in positions along the frequency axis and so do not need to be explicitly corrected.

However, this is not the case with the duration of the SSIs which can vary considerably depending on the speed of speech. Figure 31(a) and (b) show a shorter and longer SSI length image of the word ‘dark’, the latter being almost double the time duration of the first. NCC between these results is a low value of 0.63668 due to this scale change.

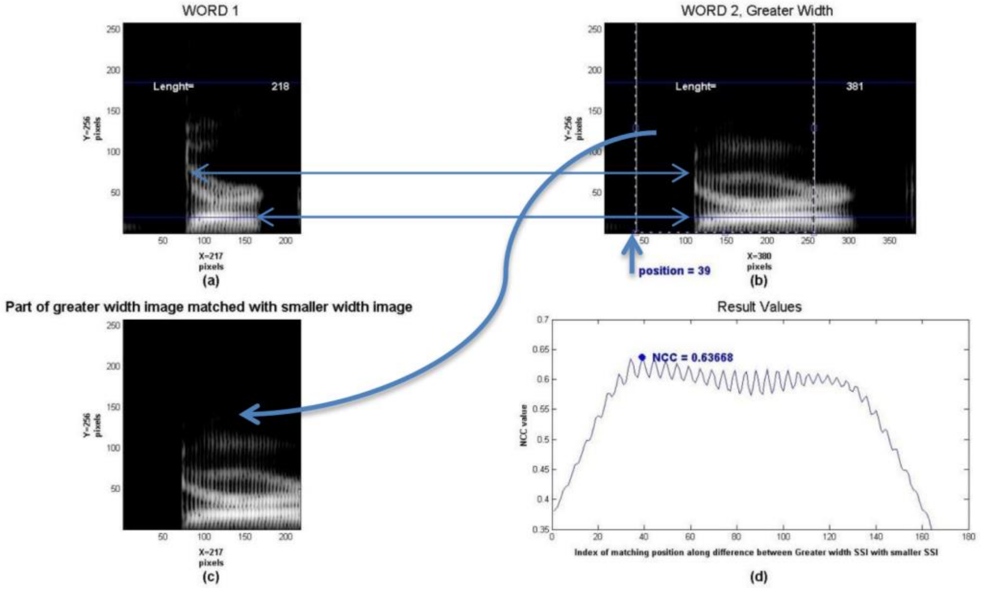


Figure 31 shows the matching between SSIs from different speakers without normalisation of the length of Frequency Transition FT: (a) Shorter FT length of SSI (of the word ’Dark’); (b) Longer length FT of same SSI from different speakers; (c) The part of the longer SSI width is matched with the shorter (FT) length image; and (d) Display of the NCC value and the position of matching between the two SSIs.

Consequently, the correlation of SSIs is affected by scaling invariance, mostly in the x-axis variable, i.e. time. Therefore, it is important to perform the correlation after transforming the input SSI and the reference SSI (they are expected to be of the same word type) to a domain such that the effect of scaling invariance is a minimum. Thus, the transformation can be represented as:

$$I\ (x,y) = I(x-(a+bx),y) \approx (a+bx)I_x\ (x,y)$$ Equation 6

The parameters (a and b) can be found by forcing the partial derivatives of the error to zero and solving the system:

$$\sum_{x\in\rho}(I(x,y)-(a+bx)\cdot I_x\ (x,y))^2 = min$$ Equation 7

Since our work focusses on a real - time implementation, we have avoided computationally expensive transformation to reduce the effect of scaling invariance.

In this study, this problem is solved at the individual level of the word by extending the shorter FT image to make it closer to the longer FT image and vice-versa i.e. changing the length of both images to a chosen normalized value. Figure 32(a) shows an initially longer time SSI compressed in length and Figure 32(b) shows an initially shorter time SSI lengthened, so they both match to a given standardised pixel count. This will clearly improve the NCC between the two SSIs which increases to a value of 0.86685 to reflect the fact that the two SSIs are now very similar. However, to achieve this normalisation successfully we need to reliably detect the FTs at the beginning and the end of the SSIs. Thus we use the FTs as markers for the SSI boundaries and normalise the pixel count (along the time axis) between them.

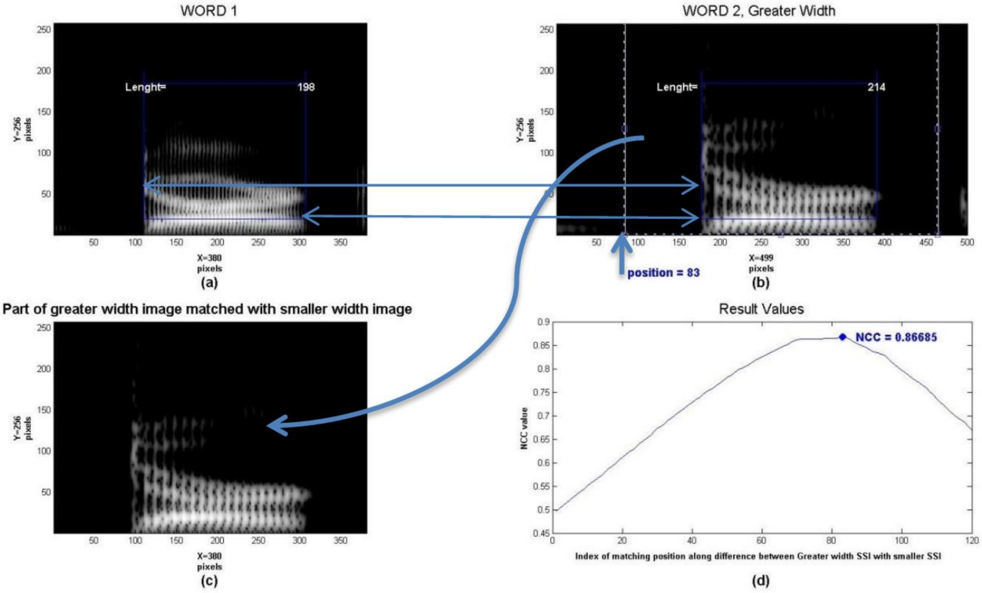
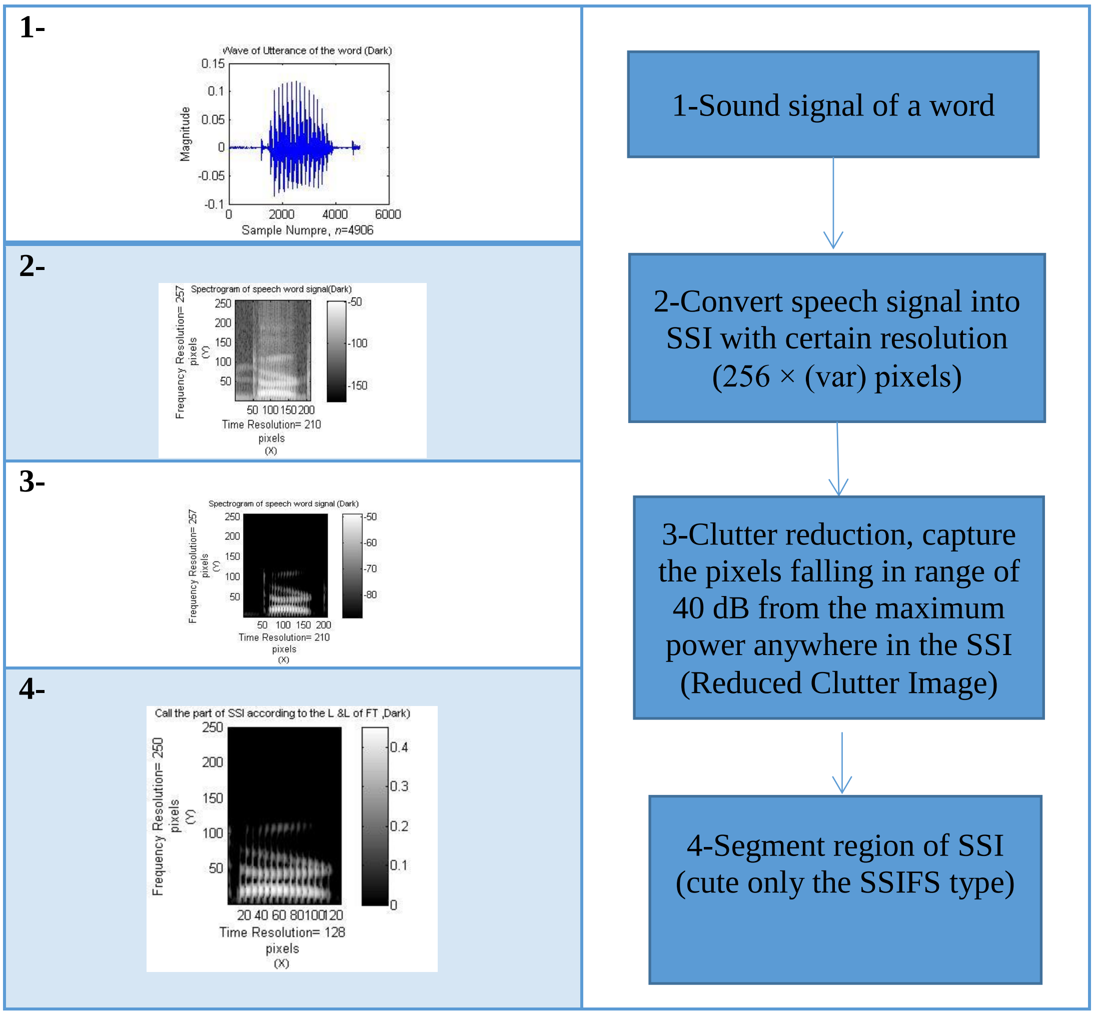


Figure 32 shows the matching with the normalised the length of SSI: (a) Smaller width (originally longer length) of SSI (of the word 'Dark'); (b) Greater width (originally shorter) of SSI; (c) The part of the greater width image matched with the smaller width image, showing full coverage of the SSI; and (d) Display of the NCC value and the position of matching for the normalised SSI.

Normalising the time dimension of the SSI could generate some spacious matching when the input SSI overlaps a part of the reference SSI (template image) and so gives a high NCC. Defining all the classes and types of a certain lexicon can help to avoid these causes of false overlap matching. It should be that each class of SSI's type has an average length duration. Another alternative solution has been used by us previously [81], which is matching only SSIFS in the SSI of a word rather than whole

SSI matching. The form of the SSIFS is assumed as unique, so the probability of false overlap matching will be very low. The Pre-processing steps for preparing an SSI are summarised in Figure 33.



1-
Wave of Utterance of the word (Dark)
Magnitude
0.15
0.1
0.05
0
-0.05
-0.1
0
2000
4000
6000
Sample Numpre, n=4906
2-
Spectrogram of speech word signal(Dark)
Frequency Resolution= 257 pixels (Y)
250
200
150
100
50
50 100 150 200
Time Resolution= 210 pixels (X)
-50
-100
-150
3-
Spectrogram of speech word signal (Dark)
Frequency Resolution= 257 pixels (Y)
250
200
150
100
50
50 100 150 200
Time Resolution= 210 pixels (X)
-50
-60
-70
-80
4-
Call the part of SSI according to the L &L of FT ,Dark)
Frequency Resolution= 250 pixels (Y)
250
200
150
100
50
20 40 60 80 100 120
Time Resolution= 128 pixels (X)
0.4
0.3
0.2
0.1
0
1-Sound signal of a word
2-Convert speech signal into SSI with certain resolution (256 × (var) pixels)
3-Clutter reduction, capture the pixels falling in range of 40 dB from the maximum power anywhere in the SSI (Reduced Clutter Image)
4-Segment region of SSI (cute only the SSIFS type)

Figure 33 Pre-processing steps for preparing SSI.

A test (1) was conducted between a reference SSI and a group of twelve SSIs of the word ‘Dark’, each spoken by different individuals. The reference SSI has a formant transition (FT) length equal to 108 pixels and the test SSIs have FT lengths of: 150, 145, 171, 169, 157, 122, 129, 111, 100, 120, 215 and 121 units, respectively. The test has been repeated for cases: 1) without applying the pre-processing steps; 2) after applying only clutter reduction; and finally 3) by applying all pre-processing steps. Thus the test of case 3 was done by re-sizing the SSIFS lengths of the test group so they each become closer to the SSIFS length of the reference SSI, resulting in the SSIFS lengths of the test group becoming modified to be: 108, 107, 106, 108, 107, 104, 106, 100, 106, 108 and 107 units, respectively. Figure 34 shows how the results of the NCCs are improved after applying the pre-processing procedure. It can be observed that the NCC values become high enough to give an average similarity around 78%, a significant improvement from the approximately 60% average obtained without pre-processing.

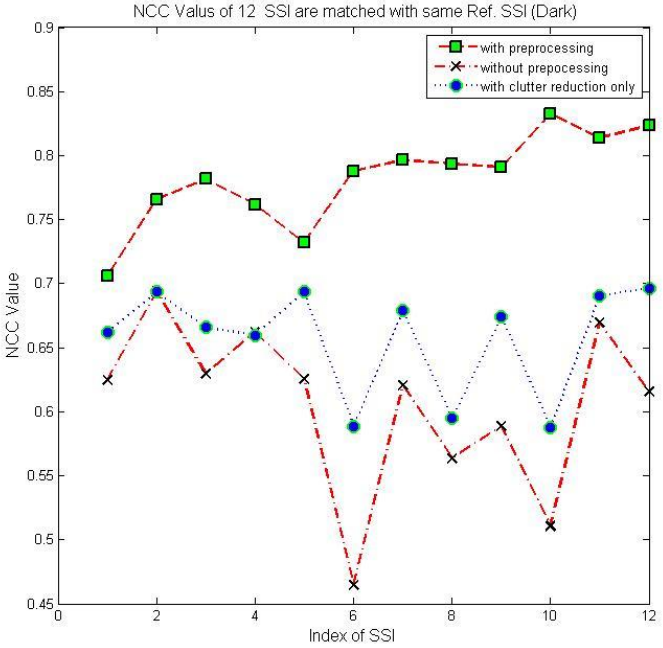


Figure 34 the pre-processing procedure is shown to improve the NCCs between the reference SSI and the test group SSIs of the same word ’Dark’.

Test (2) was run to recognise the SSI of the word ‘Dark’ (reference SSI) among a set of SSIs of words [‘Dark’, ‘Water, ‘Wash’ and ‘Get’] (as input SSIs). The results of the NNC between the reference SSI and input SSI are: 0.8568, 0.6557, 0.6247, and 0.5987, respectively for the input SSIs: ‘Dark’, ‘Water, ‘Wash’ and ‘Get’.

The pre-processing procedures described in this study form a sequence of steps prior to matching two SSIs of any given word. The steps are straightforward and can be implemented computationally efficiently. The results presented in Figure 34 show some preliminary but encouraging results indicating the described pre-processing operations when applied to the SSIs do indeed improve subsequent inter-person NCC based word recognition.

3.13 The start and end points of words on the SIR-CODE (L1-2) classification

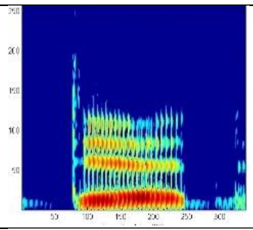
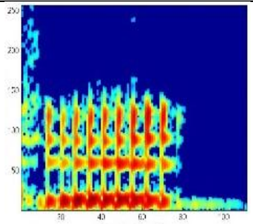
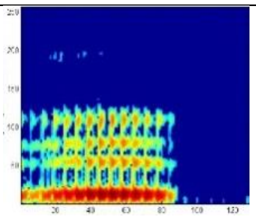
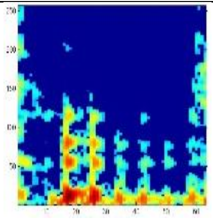
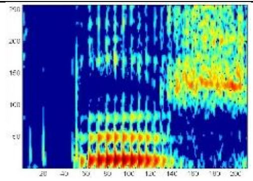
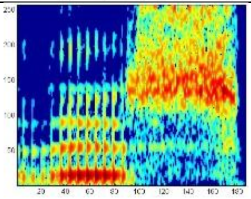
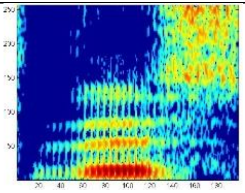
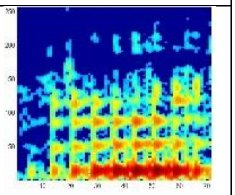
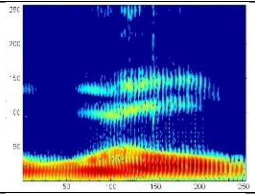
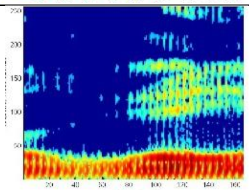
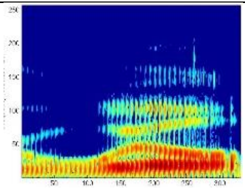
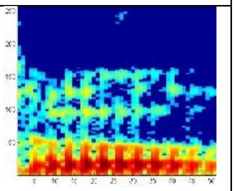
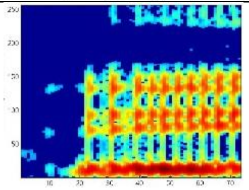
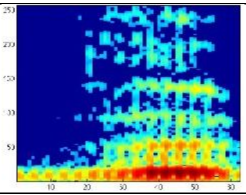
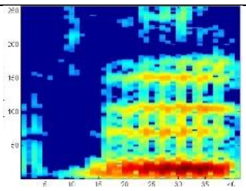
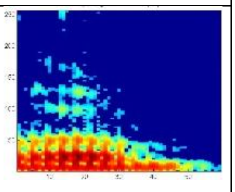
The endpoint detection is a process to separate the speech segments of an utterance from the background, i.e., the non-speech segments. Endpoint detection is a verification of speech segments that becomes relatively difficult in noisy environments. A problem of the endpoint detection is that it cannot be recovered in the later stages of recognition. Thus, when a word is actually spoken, the speech segments can be reliably separated from the non-speech segments in the initial stage of recognition. One of the advantages of the SSI is that it is robust for speech noise. Indeed, the start and end points of the segments of five patterns (FFP, SFFP, LAFP, SAFP and G in the SIR-CODE-L1-2) are very clear. There is no overlap of any of the patterns of L1-2 over each other. This is obvious through the figures demonstrated previously in this chapter. The four segments of L1-2 (apart from the G pattern) already have an underlying information and the combination of them including the G segment makes a word in language have a unique image object. Therefore, the SIR-CODE (L1-2) can increase the performance of ASR effectively.

Clear start-end points can help to implement the dynamic time warping of the SSI patterns. This is very important for implementing a continuous speech recognition based only on SSI recogniser.

3.14 Labelled and segmented speech of TIMIT Database

The TIMIT database defines the units of speech into deferent levels: continuous speech sentence level; word level; and phone level, that can be trained. The success of the training algorithms is highly dependent on the quality and detail of the annotation of those units. TIMIT contains a total of 6300 sentences, 10 sentences spoken by each of 630 speakers from 8 major dialect regions of the United States. TIMIT transcriptions are based on 61 phones [86]. TIMIT has become the database most widely used by the speech recognition research community.

However, we have observed an important observation of the TIMIT speech database. Some words in TIMIT are incorrectly different in regard to starting and ending points. This variation has been checked both by hearing them and by comparing their SSIs. As an example, Figure 26 shows some such words : ‘Did’, ’This’, ’Will’, ’The ’, ’In’, ’From’, ’And’ and ’To’. Each word has been displayed by 4 SSIs to present visually the difference in pronunciations of four different speakers and from different sentences in the TIMIT database for the same word. The SSI words on the right hand side of Figure 35 are very different from the first three signatures and that is due to the inaccurate defining of the start and end of some words. In fact, the issue of the difference in starting of ending points is more obvious in short words like ‘And’, ‘To’, and ‘In’. That shows the shortcomings in the TIMIT database rather than the failure of SSI recognition ability.

word	**Three examples of the same word for different speakers and genders from different sentences.**			**inaccurate defining of start and end points**
Did				
This				
Will				
The				

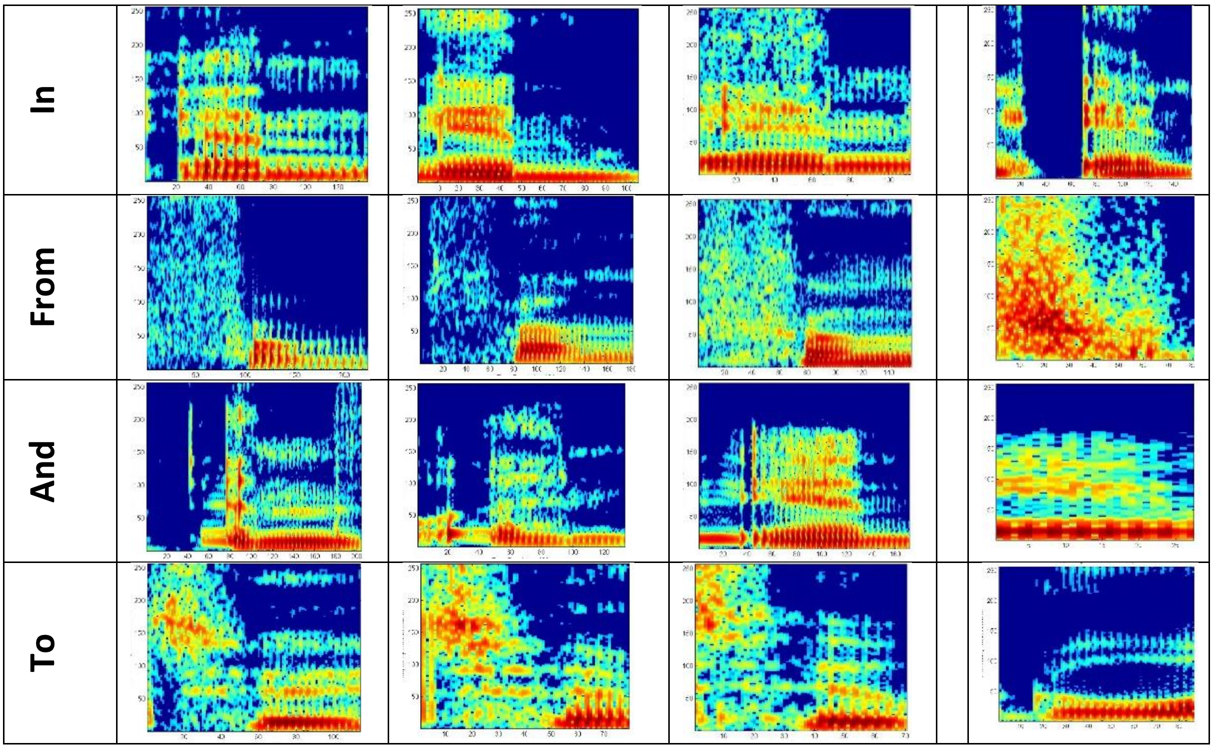


Figure 35 shows the shortcomings in the TIMIT database in defining start and end points for some words.

3.15 Discussion

Our work is focussed on the English language, but in principle it could be applicable for all spoken languages. Speech segmentation is a mental process used by humans (subjective process) to identify the boundaries between words, syllables and phonemes in spoken languages. Speech segmentation is also applied by artificial processes (objective processes) to a natural language for recognition. The gap between the subjective and the objective processes, in terms of their phoneme levels, cannot be adequately solved without training algorithms, which are sensitive to any background noise.

The SSI is a robust speech recognition method for background noise and reverberation because the SSI patterns are relatively stable in their presence which makes the subsequent ASR effective. Also, the SSI matching technique is useful in matching the same word uttered by different persons. Indeed, the SSI of a word uttered by different people has locality features that are adequate for recognising it. This type of matching allows for whole feature matching of a word in one process, which cannot be achieved by the traditional speech processing, which is a sequential training process. The technique of matching at one time is suitable for recognising an isolated word or improving ASR performance. The one-go matching technique is suitable for recognition at the word level. Then, it can be used for improving ASR performance, in general.

The one go matching technique has many advantages, which we will explore later in Chapter 4 is discussing word level recognition. This does not mean that predicting speech processes such the dynamic time warping technique or HMMs cannot by applied on SSI patterns. However, they surely need adaptations to work properly with such prediction processes. We are interested in showing that the SSI patterns have sufficient speech features that they can be used for speech recognition. Hopefully, this will draw the attention of researchers to develop the SIR-CODE as an independent method for ASR.

It is important to emphasise that in this chapter only words that contain SSIFS patterns have been tested for SSI recognition and the results are effective and reliable. As we mentioned, most words in the English language contain the FFP (one of the two patterns of SSIFS), consequently, a word can be classified into the number and order in the SSI patterns and can be easily recognised. We will look for this feature and explore other image matching techniques in the next chapter that deals with image techniques to analyse the SSI patterns.

CHAPTER FOUR

4 CHAPTER FOUR

SSI Pattern Recognition (SSIPR)

4.1 Introduction

In general, image analysis techniques can be divided into three basic areas: (1) low-level pre-processing; (2) intermediate-level pre-processing; and (3) high-level pre-processing [82]. Low –level-pre-processing deals with the image sensor that may involve an automatic reaction to be applied not requiring intelligent functions in order to perform compensation actions such as noise reduction or image de-blurring. Extracting and characterizing components are tasks of intermediate-level pre-processing of an image using techniques of segmentation and description. Finally, high-level pre-processing deals with recognition and interpretation.

The image pre-processing methods we have used in SSI recognition are based on those in Table 2 in Chapter 3. These methods are based on deep understanding of SSI patterns and their sub- pattern structure. Therefore, we will now discuss some concepts and strategies from image pre-processing and recognition that can be used to perform precise matching of SSI patterns.

4.2 SSI Analysis Techniques

The SSI is in the form of an image. It is a mathematical transformation to display the magnitude of the signal spectral components versus time as a three-dimensional plot (time vs. frequency vs. amplitude). Therefore, the accuracy of displaying the SSI is controlled by a windowing process, as has been discussed in detail in Chapter 2. Therefore, the SSI analysis is somewhat different from sensor image analysis, mainly in the form of image enhancements required. In addition, template matching (finding small parts of an image by matching a template image) is usually applied in image recognition. In contrast, the whole image can be matched to the SSI target in SSI recognition (rather than part of the image).

Some issues of SSI analysis and matching have been addressed in a paper by Al-Darkazali et al [81] and part of the procedure has been explained in Chapter 3. The details of analysis of SSIs in this work [81] can be summarised in three stages, as illustrated in Figure 36.

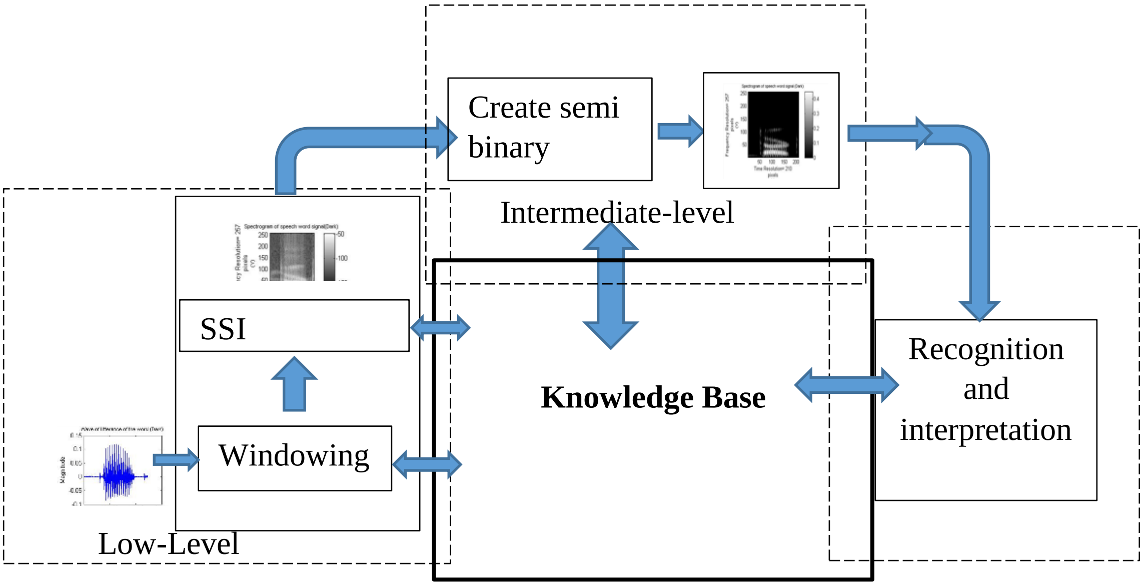


Figure 36 Elements of SSI analysis.

4.2.1 The three elements of SSI analysis:

1) The SSI low-level analysis deals with functions to represent a speech signal as an image (SSI). We have explained in Chapter 2 the methods of representing a speech signal as an image by using time-frequency transforms, and what the optimum values are of the parameters in these transforms. Indeed, the spectrogram has been created for speech analysis. Although, it has some limitations it is the most commonly used method for speech analysis because it provides a simple and powerful means to read speech information.

2) The SSI intermediate-level pre-processing deals with the task of extracting SSI patterns. To do so, the first stage is the reduction of clutter in the SSI

which has been introduced in Chapter 3. The reduction of clutter makes abrupt changes in the grey level values of the SSI patterns more obvious which can help to define the discontinuity and similarity of the SSI patterns which is a basis of extracting objects from an image [82]. The detection of isolated objects in SSI patterns can be made easier if the SSI is converted to what we have called a semi-binary SSI. The structure of the semi-binary SSI is an SSI that contains the SSI patterns embedded in a zero intensity level pixel background. Therefore, we avoid using traditional image segmentation to isolate objects (patterns) in the SSI. As mentioned previously, the pixel intensities of the SSI patterns (PSD) show important features of the speech wave signal that assist in the next recognition stage.

We have suggested a method to create the semi binary SSI (SBSSI) of an SSI in our previous work [81]. The method is based on spectral subtraction. Technically, speech s(n) (a clean speech signal) is modulated as a random process to which uncorrelated random noise *d(n)* (the degrading noise) is added which results in y(n) [1]:

$$y(n) = s(n) + d(n) \qquad \text{Equation 8}$$

The power density spectrum of Equation 8 is:

$$\mathrm{C}_y(\omega) = \mathrm{C}_s(\omega) + \mathrm{C}_d(\omega) \qquad \text{Equation 9}$$

By normalising the range of the colour map image, the semi binary SSI of the SSI of is created as shown in Figure 37. Having achieved the semi binary SSI, the route to the next (final) stage has been prepared. Thus thresholding and region splitting can then be used on the SSI patterns. The SBSSI allows us to employ both digital image pre-processing and image matching.

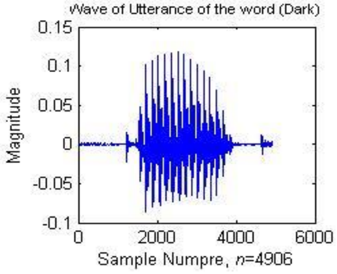
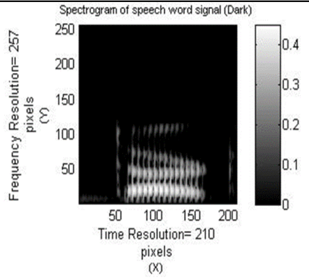
Clutter reduction the SSI →	Normalised SSI
	

Figure 37 shows the semi-binary SSI of the SSI of the word ‘Dark’.

3) SSI recognition and interpretation is the automatic analysis of the SSI into its pattern regions, which can be achieved with PR techniques (to identify them). The PR of the SSI can include both fixed and dynamic methods of recognition. The fixed method is implemented by matching a part of the complete SSI pattern to make the recognition decision. Dynamic sliding of the template to achieve recognition can be applied for continuous SSI patterns produced from natural speech. Indeed, the patterns are characterised in terms of primitive elements, sub-patterns, and their relationships [87]. PR can help to segment the SSI into classes and produce SSI clustering. PR methods applied to the SSI problem are used to identify the multiple classes and discriminate between them so allowing speech interpretation based on image pre-processing methods.

4.2.2 Matching technique for PR

Matching techniques fall into two broad categories: area based matching and feature based matching, respectively. The cross-correlation and the least squares matching approach are well known methods for area based matching. Feature based matching determines the correspondence between image features and does not require very precise initial estimates.

The area image matching methods can be classified as grayscale based and edge-based matching. The SSI patterns are not a one-target pattern (as occurs in template matching). Also, the SSI patterns have a large number of edges and there are disruptions to these patterns so that they do not have exactly symmetrical shapes. Also, the SSI patterns are variable in intensity contrast due to the variable loudness of speakers. However, the SSI patterns have limited levels of pixel intensity. Therefore, it may be that the normalised cross correlation function works well for recognising SSI patterns. Add to this that the cross correlation results can easily be interpreted in terms of image pre-processing which can help to define some naive procedures for SSI pattern recognition.

4.2.3 Normalised cross-correlation

The most straightforward method of matching is a minimum distance classifier which yields an optimum performance for an n-dimensional pattern in spatial space (more suitable for hyperploid patterns). Traditional image correlation is based on finding matches of a sub-image $w(x, y)$ of size $J \times K$ within an image $f(x, y)$ of size $M \times N$, where $J \leq M$ and $K \leq N$. The correlation between $f(x, y)$ and $w(x, y)$ is:

$$c(s,t) = \sum_{x} \sum_{y} f(x,y) w(x-s, y-t) \qquad \text{Equation 10}$$

where s=0,1,2….,M-1 , t=0,1,2,….,N-1 , and the summation is taken over the image region where w and f overlap [82], as shown in Figure 38.

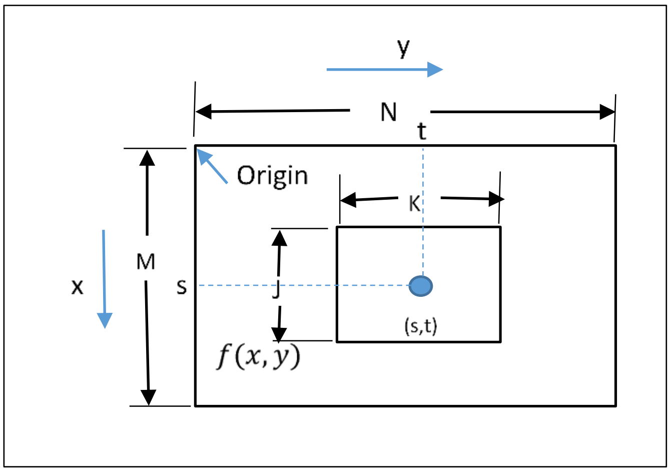


Figure 38 Template matching [82].

Images of SSIs are of equal width (J=M). Therefore, it is necessary to modify the conventional NCC to make it work with equal dimensional matching. This has been done in our reported work [81]. We called that process global matching. The global matching is to measure the similarity of the SSIs of two speech signal words uttered by different persons by calculating the NCC at each different displacement along the time axis between two comparative images, one being the greater width image $g_{SSI}\ (x, y)$ of size M×K and the other the smaller width image $f_{SSI}\ (x, y)$ of size M×N, as shown in Figure 39.

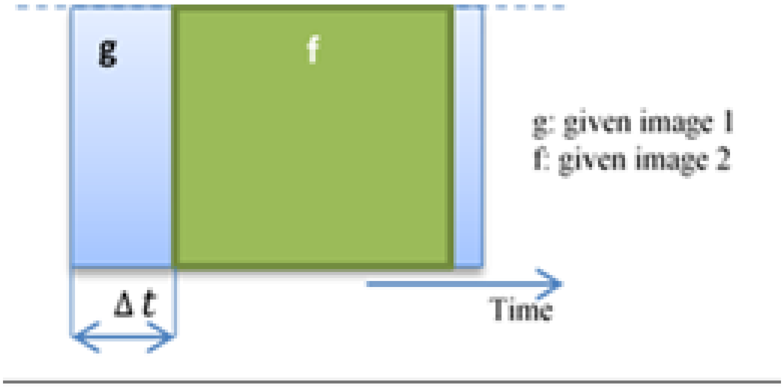


Figure 39 process global matching for SSIs.

The correlation between $f_{SSI}(x, y)$ and $g_{SSI}(x, y)$ is:

$$c(t) = \sum_{x}\sum_{y} f_{SSI}(x, y) g_{SSI}(x, y - t) \qquad \text{Equation 11}$$

where t=0,1,2,….,N-1is the position of the maximum of the correlation output which gives the starting position of the region within the greater width image that matches with the smaller width image.

Therefore, the normalised cross correlation (NCC) of SSIs can be written as:

$$NCC_{SSI}(t) = \frac{\sum_x \sum_y \left[[f_{SSI}(x, y) - \bar{f}(x, y)][g_{SSI}(x, y - t) - \bar{g}]\right]}{\left\{\sum_x \sum_y [f_{SSI}(x, y) - \bar{f}(x, y)]^2 \cdot \sum_x \sum_y [g_{SSI}(x, y - t) - \bar{g}]^2\right\}^{1/2}} \qquad \text{Equation 12}$$

where $\bar{g}$ is the average pixels value in $g_{SSI}(x, y)$, which is computed once, $\bar{f}(x, y)$ is the average value of $f_{SSI}(x, y)$ in the region coincident with the current location of $g_{SSI}(x, y)$, and the summations are taken over the coordinates common to both of f_{SSI} and g_{SSI}. The NCC_{SSI} is independent of change in amplitude of $f_{SSI}(x, y)$ and $g_{SSI}(x, y)$ (in a range between -1 to 1). However, since the NCC is average pixel based, the unvoiced patterns affect badly the NCC value, because the unvoiced patterns appear like image clutter and have no specific spatial structure.

4.2.4 Frequency matching

The multiplication in the frequency space is identical to convolution is the spatial domain. On the other hand the Fourier domain filter can be attenuated at certain frequencies and pass others. Therefore, frequency domain image matching can be used in classifying image objects. The object distortion and cluttered background are obstacles in image PR. Therefore, filters such as the Maximum Average Correlation Height (MACH) filter has been used to overcome some of these difficulties. The MACH filter has been used for classification of objects. It is employed in palm printer identification [88]. The MACH filter also has the ability to suppress clutter noise [89].

The optimal -trade-off (OT) Mach filter in the frequency domain is expressed as:

$$OT = \frac{\omega_x}{\alpha\, C_x + \beta\, D_x + \gamma\, S_x} \qquad \text{Equation 13}$$

ω_x is in the frequency domain and can be represented by training vectors $x_1, x_2, \dots x_N$. C_x is the diagonal of the matrix of the PSD of the object with additive input noise. D_x is the diagonal average PSD of the training images. S_x denotes the similarity matrix of the training images [89]. The MACH filter needs to be tuned by parameters $(\alpha, \beta, and\ \gamma)$ based on the object of interest. The noise can be considered as any unwanted objects. Mainly, the clutter noise is the additive noise which is added by the image sensor. Therefore, image clutter is random and mostly of low PSD.

4.3 Feature based matching

Pattern classification can be achieved by characterizing the quantitative information (features) of the SSI pattern; these features are primitive components such as abrupt endings, branching and merging, and disconnected segments. They are, together with their relative sizes and location, features of the patterns that help to provide recognition. In this section, we present some of the general methods to classify SSI patterns based on speech waveform distinguishing features.

4.3.1 Grey-level slicing

One of the fundamental first steps in any speech recognition system is the classification of frames as voiced or unvoiced. That is equivalent to classifying the SSI patterns of a word into the SSIFS and the SSIRS patterns. On first inspection, one of the features of the voiced frames is that they tend to be higher energy than unvoiced frames. Usually, the SSIFS are of higher energy [90]. Therefore, the SSIFS regions are represented as a darker colour than the SSIRS region in the SSIs. This can be a key to highlighting a specific range of grey levels for separating the voiced part in the SSI by using grey-level-slicing. Grey-level-slicing is one approach that can be used to display a high value for all grey levels in the range of interest (e.g. SSIFS intensity representation) and low values, below the set threshold to background grey-level tonality [82]. There are several ways of doing level slicing but they have the same basis by rounding the elements of an image to the nearest integer greater than or equal to the image.

4.3.2 Image Subtraction

The difference between the original SSI and sub-image (e.g. the SSIFS pattern of the SSI) to give an image with remaining interesting patterns in the SSI is accomplished by an image pixel by pixel subtraction, which can be expressed as:

$$\underbrace{g_{SSI}(x,y)}_{\substack{\text{remaining}\\ \text{patterns of SSI}}} = \underbrace{f_{SSI}(x,y)}_{\text{whole SSI}} - \underbrace{h_{SSI}(x,y)}_{\text{part of SSI}} \qquad \text{Equation 14}$$

4.3.3 Spatial matching

Image filters can emphasise certain features or remove other features. Image filtering is useful for applications such as removing noise, and smoothing, sharping, and edge enhancement image. It is known that in an image, the high frequency components characterise edges and other sharp details in the image, whilst image noise is a random variation of pixel intensities. On this basis, it is possible to build a filter for discrimination between the SSIFS and the SSIRS of SSI patterns.

The equivalent in the spatial domain of frequency domain filtering can be implemented by convolution processing, which multiplies the elements of the kernel by the matching pixel values when the kernel is centred over a pixel. The result is array elements of the same size of the kernel that are weighted by their neighbour values replacing the original pixel values.

Filtering can be equivalently applied in the frequency domain (FFP of the kernel), and it is often more suitable for filtering a range of frequencies. Based on the application, the filter types are low pass filter, high pass filter, directional filtering, and Laplacian of a Gaussian (i.e. band pass) filtering.

4.4 Common tasks in SSI recognition

As we have mentioned before, the SSI of a language consists of image patterns, which are a series of shapes interfaced to create a type. Thus, a language can be classified into types of SSI (e.g. SSIFS and SSIRS) so that they interface to create the SSI of a word. A type can be classified into different kinds, which defines how this pattern is implemented (e.g. each sound of Vowels, or Diphthong, or Semivowels, etc. creates a different kind of SSIFS).

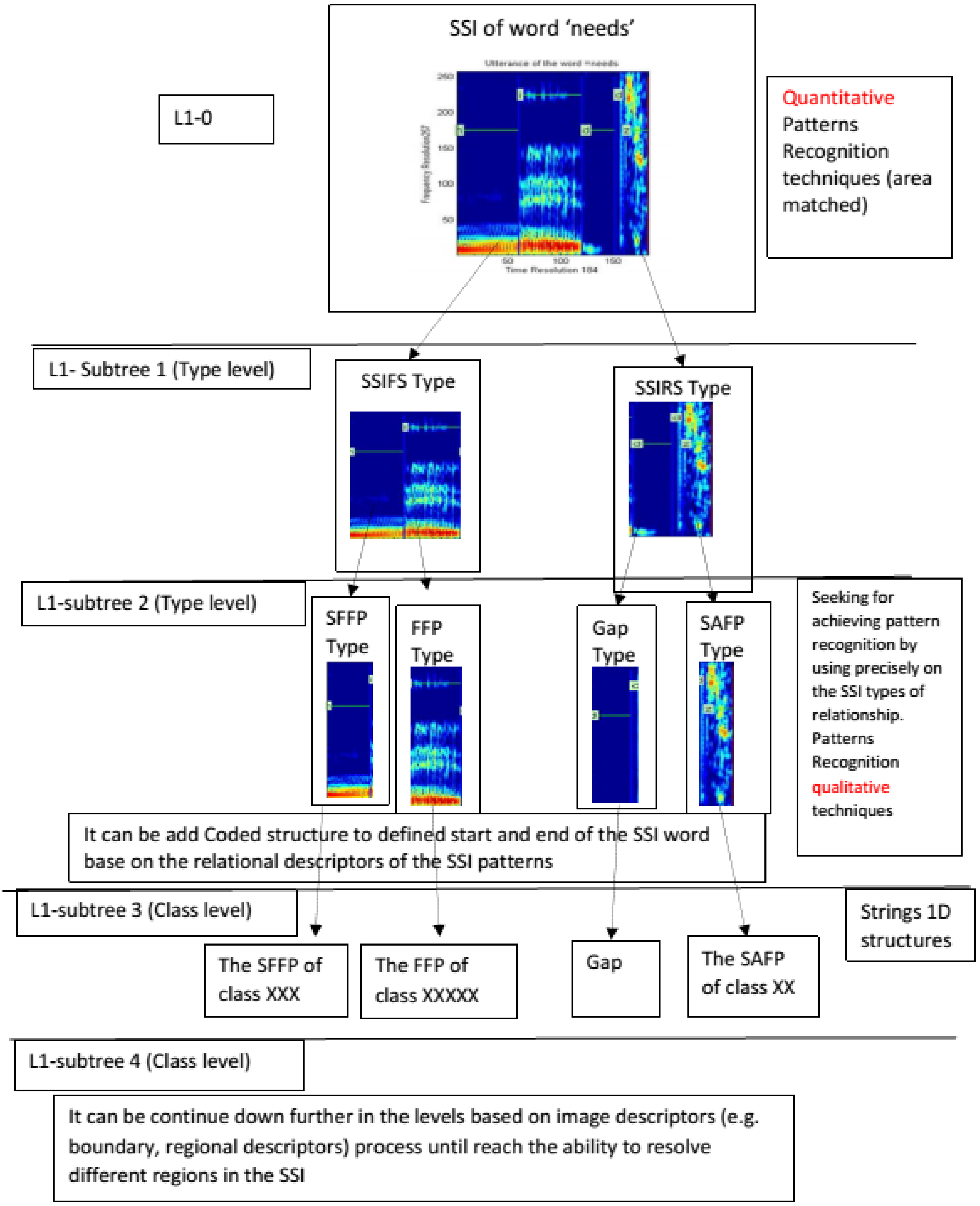
The SSIPR involves the extraction parameters, the detection of regions of interest (segmentation of SSI patterns) and, finally, the identification of the class of the SSI (category). Thus, the SSIPR process contains algorithms of segmentation, classification, and parsing. The parsing searches algorithm parameters for constructing structural descriptions (tree classifiers) to identify the segmented SSI patterns and their relationships in descriptions. The SSI recognition decision (2-D SSI) needs appropriate positional relations to be used as a method for reducing it into 1-D structures of the strings for PR. The tree structure graph helps to discriminate complex structures in a straight forward way because we seek to design and build machines that can recognise patterns automatically.

4.4.1 The tree structure example

A natural generalisation of trees is a graph that allows the description of complex structures in a straightforward way on a high, problem-oriented level. Figure 40 shows an example of describing the SSI of the word ‘needs’ into its primitive elements or sub-patterns, and their relationships based on Table 2 in Chapter 3. The word ‘need’ can be classified into SSIFS and SSIRS types. The type SSIFS can be classified into kinds of FFP and SFFP while the type SSRFS into kinds of Gap and SAFP. A type or kind can be classified into classes (e.g. SSIFS can be created by different speech sounds).

However, it may be possible to continue down in levels for further subdivision units based on image qualitative descriptors. Proceeding down the tree can resolve different regions in the SSI for more precise recognition of the class type of an SSI.

Consequently, the SSIPR decision can be by integrating both a numerical decision by SSI matching (quantitative) and category decision by the SSI structure (qualitative) or by one of the methods individually.



Seeking for achieving pattern recognition by using precisely on the SSI types of relationship. Patterns Recognition qualitative techniques

It can be add Coded structure to defined start and end of the SSI word base on the relational descriptors of the SSI patterns

It can be continue down further in the levels based on image descriptors (e.g. boundary, regional descriptors) process until reach the ability to resolve different regions in the SSI

Figure 40 A tree description of the SSI of the word 'need' based on L1-1 categories (Table 2, Chapter 3).

4.5 Lexicon speech recognition

The lexicon has been used in this study for providing the basic acoustic units of the SSI recogniser. Our lexicon is a data set consisting of the same 10 digits and some other words extracted from the TIMIT database. The data is of 39 words. Each word in the Lexicon has 15 samples and is uttered by 10 persons; ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'zero', 'start', 'stop', 'yes', 'no', 'go', 'help', 'erase', 'rubout', 'repeat', 'enter', 'dark', 'she', 'your', 'suit', 'greasy', 'wash', 'all', 'year', 'ask', 'carry', 'oily', 'rag', 'like', 'that', and 'him'). An exception is for four words, which have seven samples, i.e.: ('crab', 'challenged', 'quick', 'stab'). The total number of differently pronounced words is therefore equal to 553.

4.6 Lexicon word recognition by SSI matching (numerical decision)

This work comes out of standard speech recognition. So to demonstrate the results all the lexicon words have to be compared. In Chapter 3, it has been shown that the NCC does recognise the SSIFS patterns more effectively than the SSIRS. Moreover, the SSIRS patterns can effect badly the value of the correlation by decreasing the correlation value between SSIs of the same word (by a different speaker) or increasing the correlation value between SSIs of different words. This is because the SSIRS have random pixel distributions. Therefore, the expected matching of the SSI of a word (input image) with the remaining SSIs of the lexicon words cannot give encouraging results with NCC.

The recognition decision by matching only has been tested by both spatial and frequency matching. Figure 41 demonstrates the maximum cross correlation (MCC) of the SSI of the word ’two’ with the whole lexicon words. The label MCC in Figure 41 points to the input word location in the database (lexicon), which is row 2 of index 9. The X-axis demonstrates the same word index (word uttered by different speakers), whereas the Y-axis demonstrates the lexicon words. The NCC values of the word input with lexicon words are mapped in the range of the colour bar. The maximum value of the MCC is the result of correlation between the SSI of the word input ‘two’ with itself.

As expected the NCC values are affected by matching of SSIRS patterns. The results of Figure 41 show that the maximum NCC values on row 2 which contains the word 'two' but still there are other values that are close to the MCC of the word 'two'.

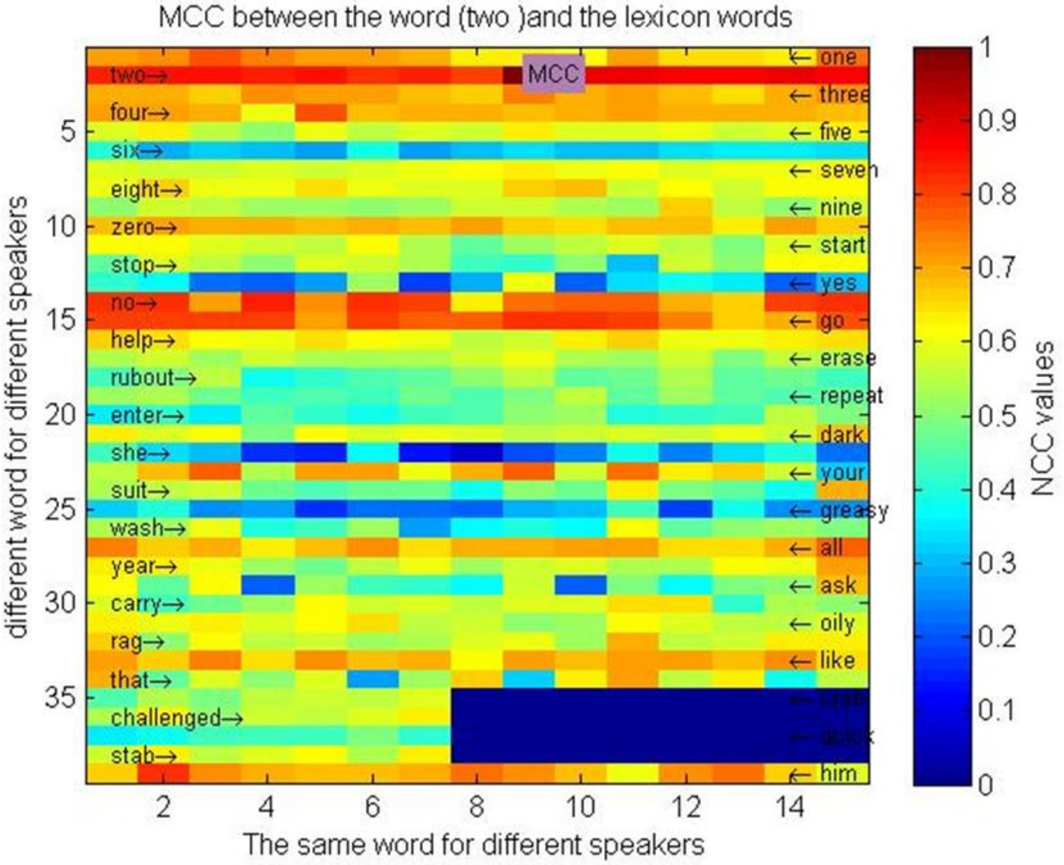


Figure 41 illustrates matching the SSI of the word' 'two' with the lexicon words in the spatial domain by NCC.

In this case, we determine that the SSIRS is a noise component in SSI and it is necessary to suppress it. Next, we have applied the MACH filter for recognising an SSI pattern. The SSIR is random pixels but it has high PSD values such as in the SSIFS. In order to suppress the SSIRS noise, the MACH filter should be tuned to properly remove its affect as far as possible.

The MACH filter tuning is accomplished by estimating three parameters (α, β, and γ) and has to determine their value for each SSI pattern to be recognised. So, the trade-off between (α and β) should be dealt with first. Figure 42 shows the procedure to optimise the MACH filter parameters.

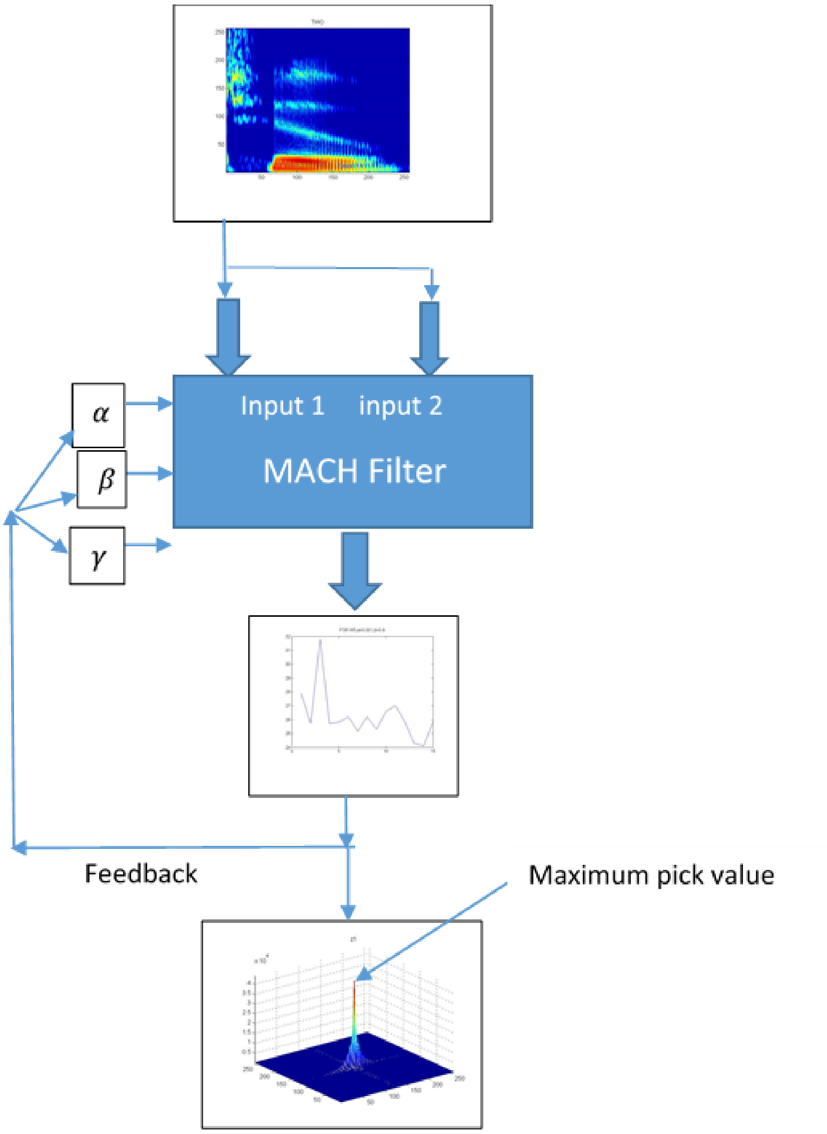


Figure 42 shows the procedure to optimise the MACH filter parameters.

We find the maximum peak at the best trade-off of α and β. The peak value is used to normalise the frequency matching results between an SSI of the input word and whole the lexicon words in order to display the result in the same way as the NCC matching of the SSI. Figure 43 shows the results of frequency matching between the word ‘two’ and the whole content of the lexicon words. The MACH filter produces better recognition results than NCC matching but still there are some incorrect recognition cases such as with words ‘no’ and ‘go’.

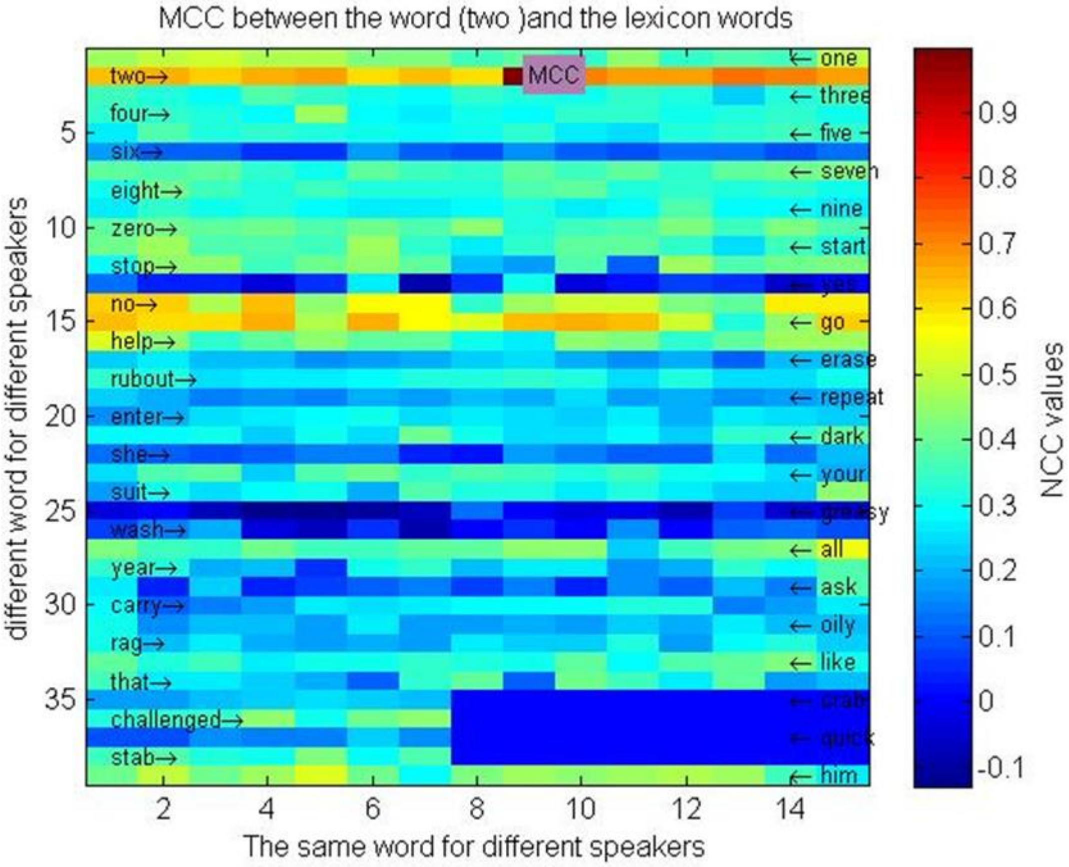


Figure 43 shows results of MACH filter matching the SSI of the word 'two' with other lexicon words.

It seems that the randomness of the SSIRS patterns cannot be completely suppressed by tuning the MACH filter. Indeed the SSIRS random patterns have some pixel values almost the same as the SSIFS pattern pixels. However, the fingerprint of the SSI is represented by the SSIFS patterns. Therefore, the tuning of the MACH filter cannot discriminate the SSIFS pattern(s) in an SSI properly.

Therefore, choosing a region of interest for matching is necessary. In other words, segmenting the area of interest based on Table 2 (in Chapter 3) of the SSI is required to improve the accuracy of the recognition decision of the SSI. The SSI patterns have been suggested by this study. Therefore, there is no literature for segmenting SSI patterns in such as SSIFS and SSIRS. In addition, the SSI patterns have regularity in the shape but there are no fixed number of features such as of point, line and pixel values. Therefore, standard image segmentation methods must be modified to be suitable for this kind of SSI image. A new method of SSI pattern segmentation will this be proposed in this study.

4.7 SSI Pattern recognition for L1-1 categories

The L1-1 category is a classifier level to classify an SSI as to which type of SSIFS and SSIRS patterns it contains. The differences between the SSIFS and the SSIRS speech are clearly visible in the SSI. The SSIFS speech has a comb-like spectrum transition (i.e. has sharp details); the SSIRS exhibits a non-harmonic spectral structure (such as image noise). Furthermore, the SSIRS segments have most of their energy at high frequencies. On other hand, speech tends to consist of periods of high power (voiced phonemes), followed by periods of low PSD e.g. unvoiced phonemes and inter-word pauses [90].

4.8 Distinguishing between SSIFS and SSIRS in the SSI (L1-1 categories)

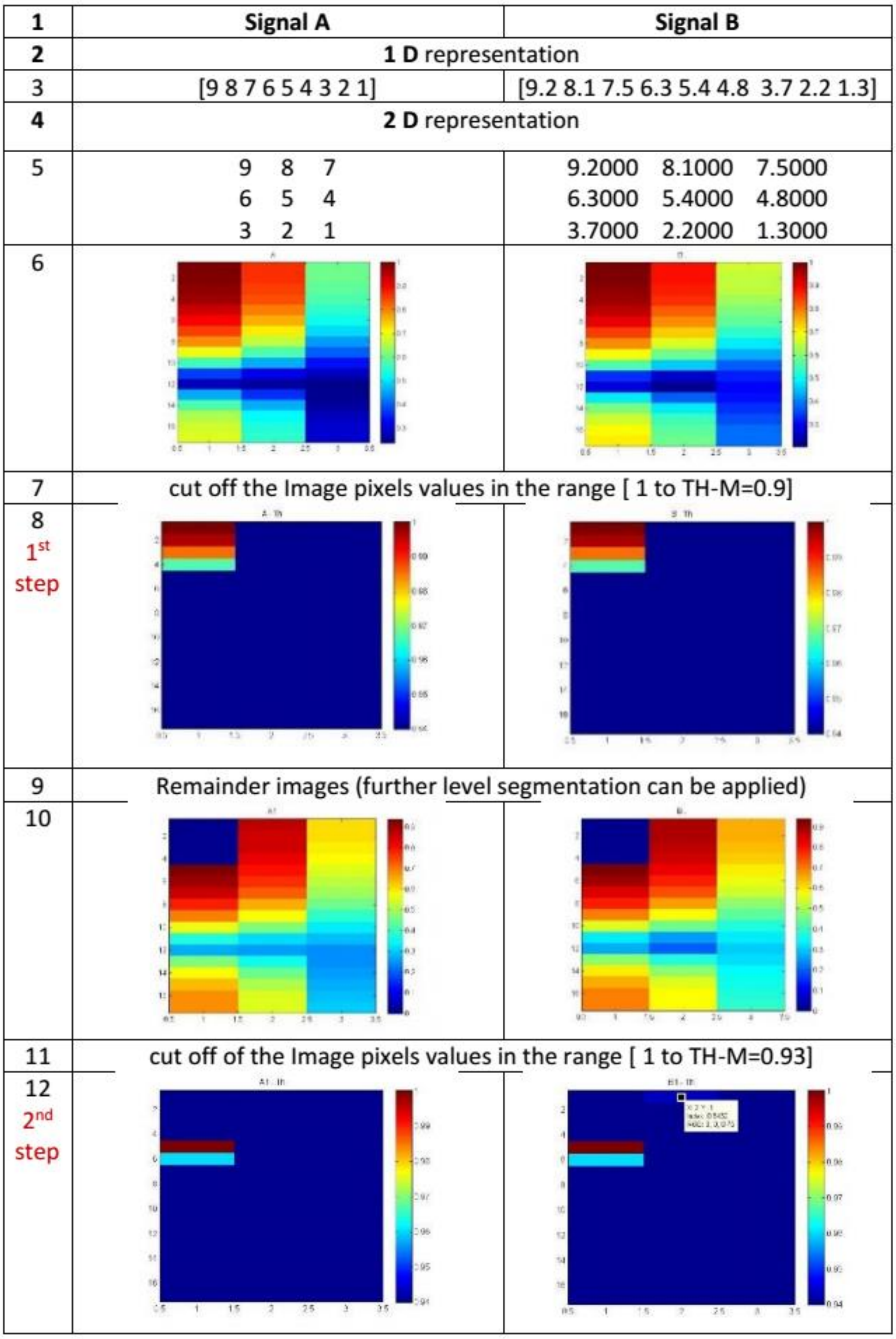
Segmentation refers to the labelling of objects and change of the representation of an image into something that is more simple and meaningful for analysis. By that means we can get an object of interest surrounded by zeros pixels. Recognition of an SSI pattern by matching needs accurate image segmentation. Therefore, the object of interest should be close to the true value (in precision and reproducibility). In image segmentation, unrelated operations of image analysis can be employed such as interpolation, (grey-level) filtering, and registration to optimise the performance of the image segmentation to provide all needed object information. Practically, the extent of the image segmentation is because it could be that the object of interest is a 3D object detected by different image sensors. The image registration is to align multiple sensor data into a single integrated image in regard of time and viewpoint. The integrating of the single image that is intensity-based and feature-based are suitable for computer vision, medical imaging and target recognition. The intensity-based method is a correlation metric of comparing intensity patterns in an image, while the feature-based methods find a correspondence between points, lines and contours (image features). The SSI patterns are 2D shapes formed by chaotic arrangement of pixel values, so it could be that there are no continuous lines in these regions.

4.8.1 Specific Segmentation algorithm for the SSI based on pixel grey-level

Words uttered by different speakers produce different acoustic energy levels. The acoustic spectral density of speech is the amount of acoustic energy that is represented as pixel values in the SSI. Moreover, the first formant pixels are expected to be the higher values in the SSI. Therefore, the difference in peak values of input speech can avoided by normalising pixel values in the SSI. Then, the threshold from the maximum (Th-M) can be applied to leave only the higher value pixels, which are equivalent to approximating the higher spectral format(s) in the SSI. In other words, the SSI patterns can be classified as bands of pixel values from the maximum regions in an SSI.

To explain this issue let us take the experimental example shown in Figure 44. The two matrices are slightly different in value and are displayed as images that represent SSI images. Normalisation of the two image is shown in part 6 of Figure 44. Then, a high-pass filter between the range of 1: Th-M= 0.94 is applied to both. The results are shown for the same separation in part 8 of Figure 44. Then, putting zeros instead of the first segmented part, the remainder images are obtained as shown in part 10 of Figure 44. Pixels values in both remainder images are almost in the same value range. The image can be segmented to a further level in the remainder image after renormalizing the remainder images by applying the same sequences (except Th-M =0.93) on the remainder images to segment a new level as shown part 12 of Figure 44, and so on, for further lower level equivalent pixels in the image example.

Thus, the equivalent high PSD regions of a speech signal in an SSI can be segmented in the same way. The core idea in this study is to segment the SSI patterns. The algorithm is based on the noise background reduction algorithm. The noise background reduction method has been used in Chapter 3 to represent a speech wave signal (within only maximum and a minimum human generated PSD) into the equivalent pixels in the SSI. Therefore, clutter in the SSI can be cut at level that does not destroy the SSI patterns of a word.



1	**Signal A**	**Signal B**
2	**1 D** representation	
3	[9 8 7 6 5 4 3 2 1]	[9.2 8.1 7.5 6.3 5.4 4.8 3.7 2.2 1.3]
4	**2 D** representation	

5	9 8 7
	6 5 4
	3 2 1
		9.2000 8.1000 7.5000
		6.3000 5.4000 4.8000
		3.7000 2.2000 1.3000

6		
7	cut off the Image pixels values in the range [1 to TH-M=0.9]	
8 1st step		
9	Remainder images (further level segmentation can be applied)	
10		
11	cut off of the Image pixels values in the range [1 to TH-M=0.93]	
12 2nd step		

Figure 44 shows the threshold example of segmented regions of the higher pixel values in the SSI.

4.8.2 First formant segmentation in the SSI algorithm

The FFP is the part of the SSIFS that contains the highest PSD in the SSI which is represented by darker pixels in the SSI (red colour). The FFP can be segmented by increasing the level cut until leaving only the FFP pattern in the SSI (SSI pixel values should be normalised). In other words, the SSI patterns are formed of different pixel levels (different PSD).

Then, segmenting of the SSI patterns can be controlled between the maximum pixels (maximum PSD) to a lower certain level (threshold value) to leave only pixels equivalent to the higher PSD of a word by the SBSSI.

Figure 45 shows images of the SSI of the word ‘carry’ and the $SSI_{Th=0.8}$. The $SSI_{Th=0.8}$ contains only the first format in the SSIFS in the word ‘carry’. Indeed, the $SSI_{Th=0.8}$ contains only pixels within 0.8% from the maximum in the PSD of the word ‘carry’.

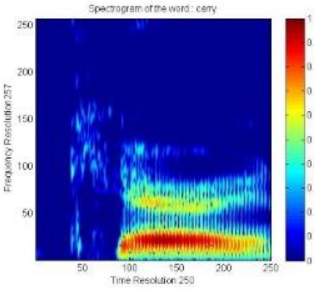
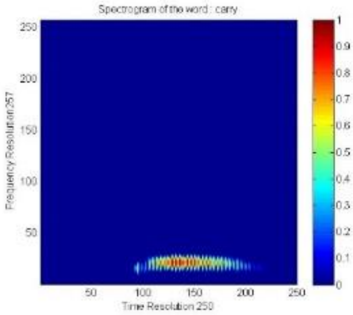
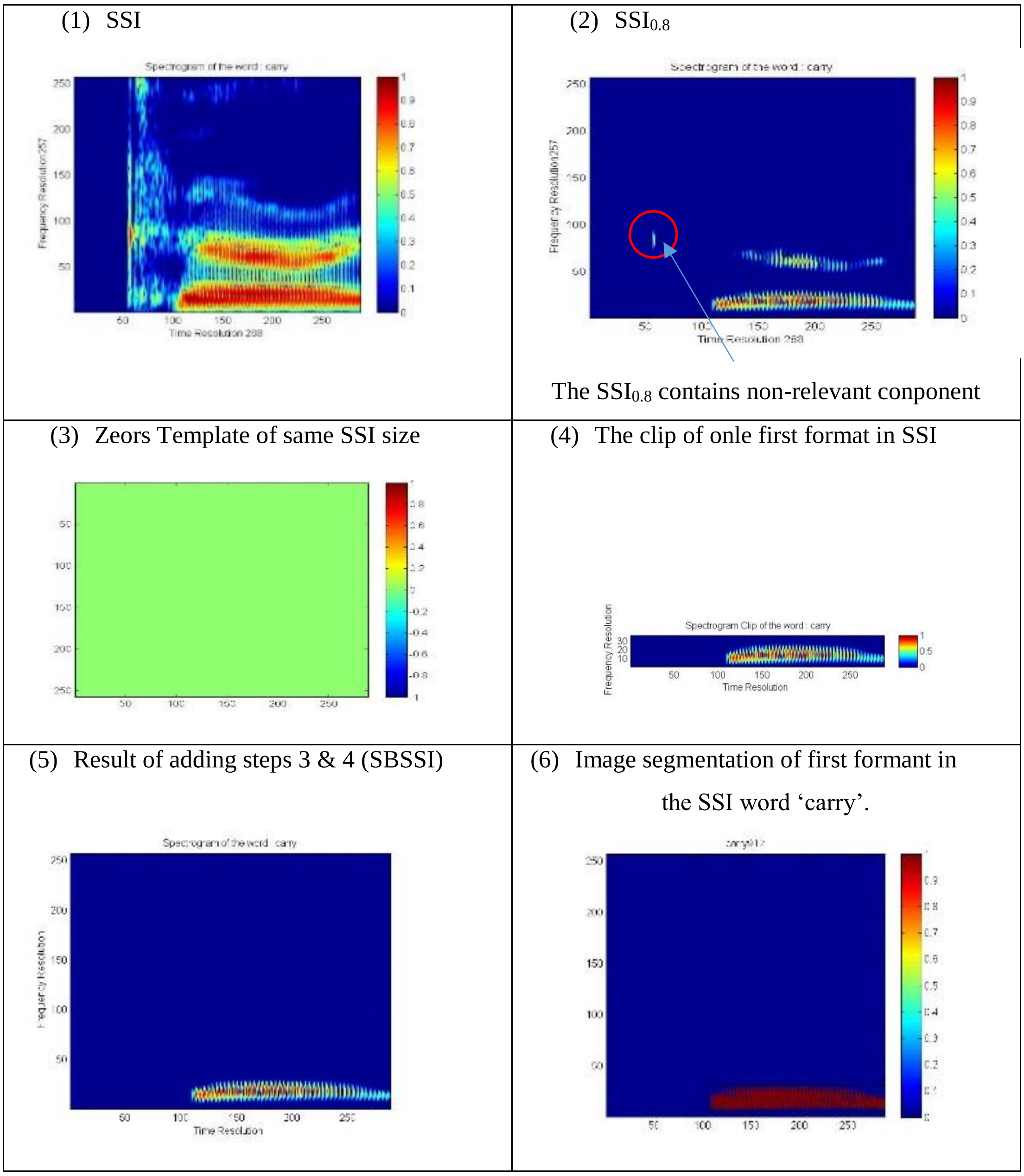
A) The SSI of the word ‘Carry’	B) The $SSI_{Th=0.8}$
	
The whole patterns of the word ‘carry’	The SBSSI of the word ‘carry’

Figure 45 shows cutting of the first format in the SSIs of the word ’carry’.

There is no guarantee that the remaining pixels even approximately represent the first formant in an SSI. To make sure that this is the case it is necessary to avoid the appearance of unexpected remainder pixels in an SSI_{Th} apart from those in the first formant. This can be done by cutting a strip of $SSI_{0.8}$ between 5 to 40 horizontal lines in the $SSI_{0.8}$ and adding it to a zero matrix of the same size as the SSI.

Finally, the binary image containing only the segmented part of relevance can be created by replacing all the remainder pixel values of the part of relevance by ones. These steps are illustrated in Figure 46. As a result of this a binary image has been obtained that allows the determination of the location of the approximate first formant in the SSI.



(1) SSI

(2) $SSI_{0.8}$

The $SSI_{0.8}$ contains non-relevant conponent

(3) Zeors Template of same SSI size

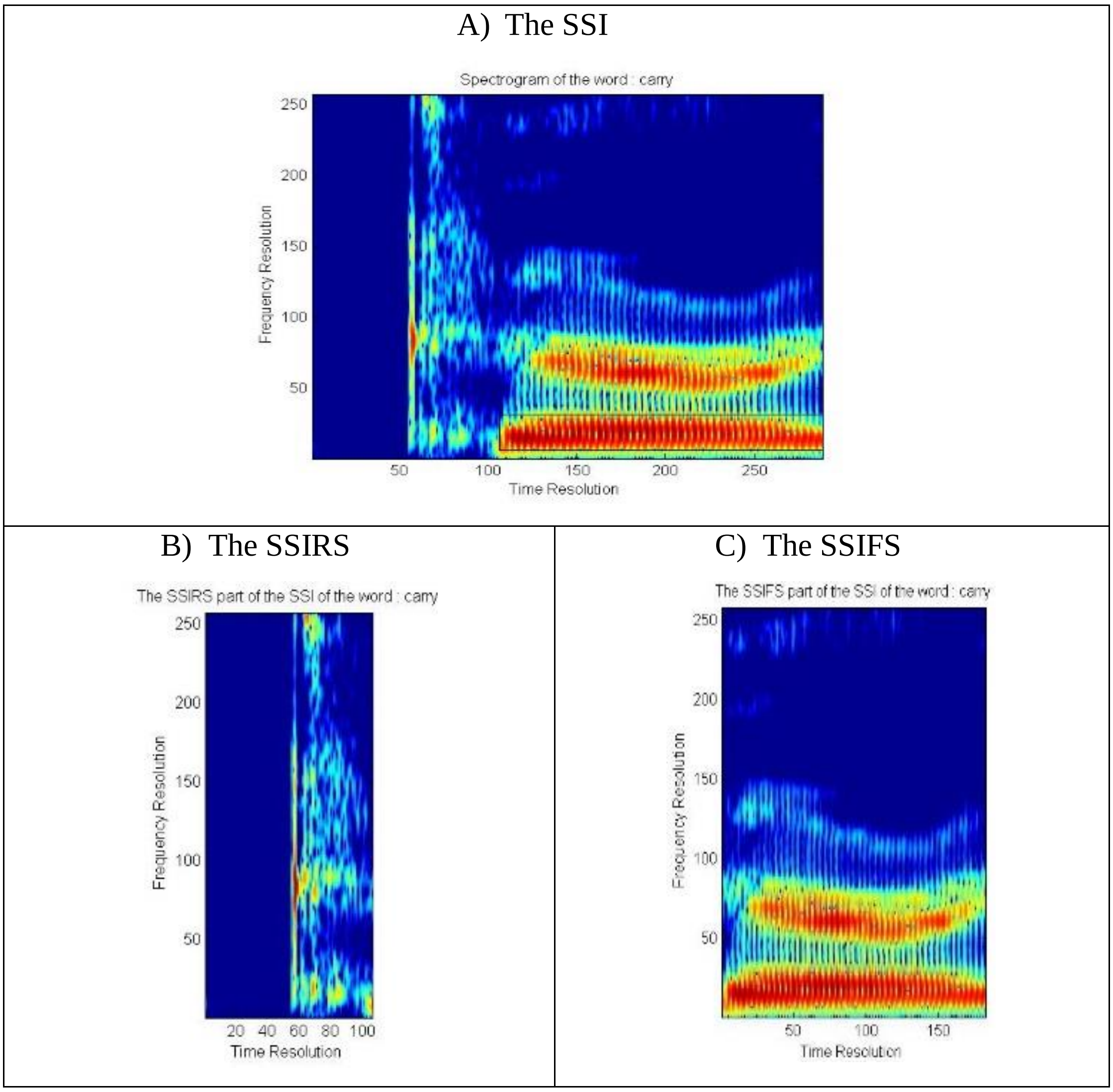
(4) The clip of onle first format in SSI

(5) Result of adding steps 3 & 4 (SBSSI)

(6) Image segmentation of first formant in the SSI word 'carry'.

Figure 46 shows steps to segment the FFP in the SSI (higher PSD part of SSI).

The SSIFS in the word 'carry' consists of only an FFP pattern (the SSIFS is either only an FFP pattern or both an FFP plus an SFFP pattern). In this case, the discriminant FFP leads to recognition of the SSIFS in the SSI and so the SSIRS can be discriminated too. Figure 47 (A) shows the SSIFS is indicated by a black rectangular area so differentiating between the SSIFS and SSIRS part in the SSI of the word ‘carry’ as shown in (B) and (C) in Figure 47.



A) The SSI

B) The SSIRS

C) The SSIFS

Figure 47 shows an example of differentiating between the SSIFS and SSIRS patterns based on a segmented first format in the word ‘carry’.

The SSIs are combinations of different types of patterns of FFP, SFFP, LAFP, SAFP, and G. Thus the discrimination between all SSI patterns is not as simple as the example of the SSI of the word ’carry’. Therefore, the algorithm needs to be improved to be more generally effective.

However, this specific segmentation algorithm for the FFP in the SSI is the core of a general algorithm for discriminating all SSI types. The differentiation between the SSIFS and SSIRS patterns allows the collection of statistical information about these patterns in an SSI (e.g. location, centroid, dimension etc.) for PR.

Matching SSIFS patterns is much easier than matching the FFP patterns for the following reasons: firstly, the precise capture of the FFP patterns from within the SSI patterns is avoided (the segmentation of the FFPs is approximate); secondly, matching of SSIFSs (FFP+SFFP) is richer than matching only FFPs, because the SSIFS contains additional information to the FFP pattern.

The SSIFS pattern could be formed of FFP and SFFP patterns. As an example, the SSI of word 'seven' can be classified into SSIFS (in the orange box) and SSIRS patterns as shown in Figure 48. In this example, the SSIRS is shaped by the sound /S/ and the SSIFS consists of the FFP (/EY/ and /AX/) plus an SFFP /N/, while, the /V/ is part of the SSIFS because it is effected by surrounding sounds of the FFP.

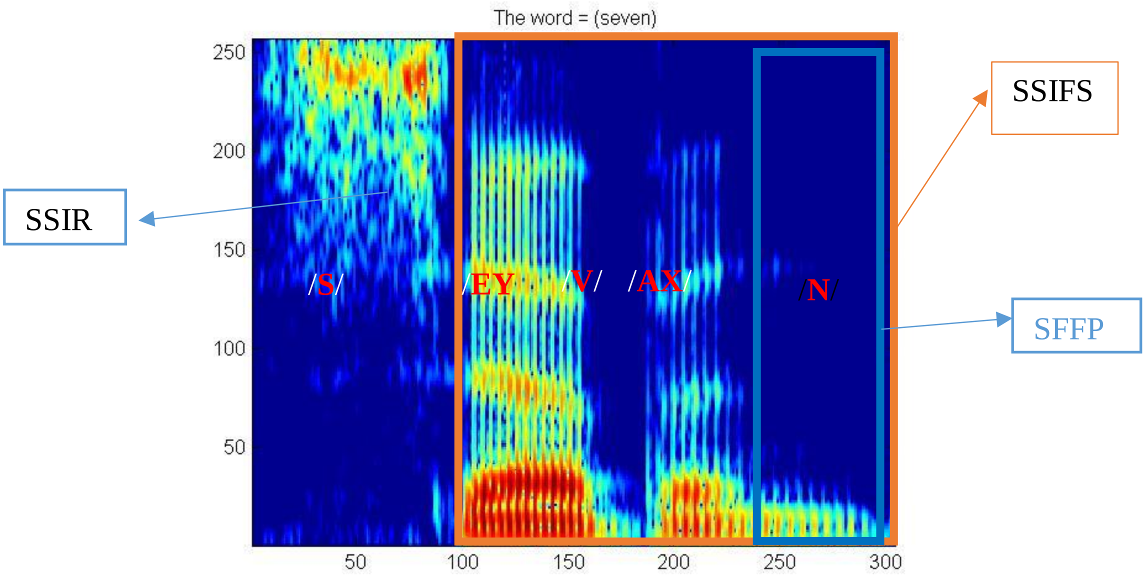
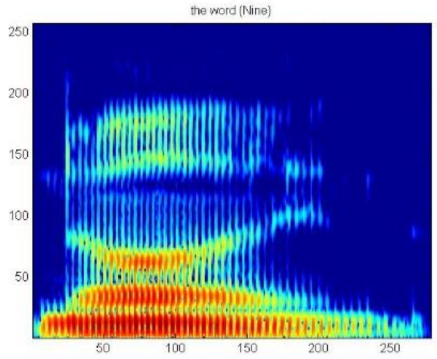
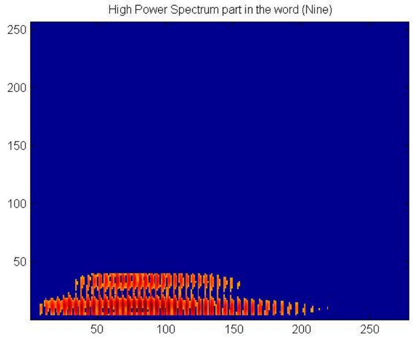


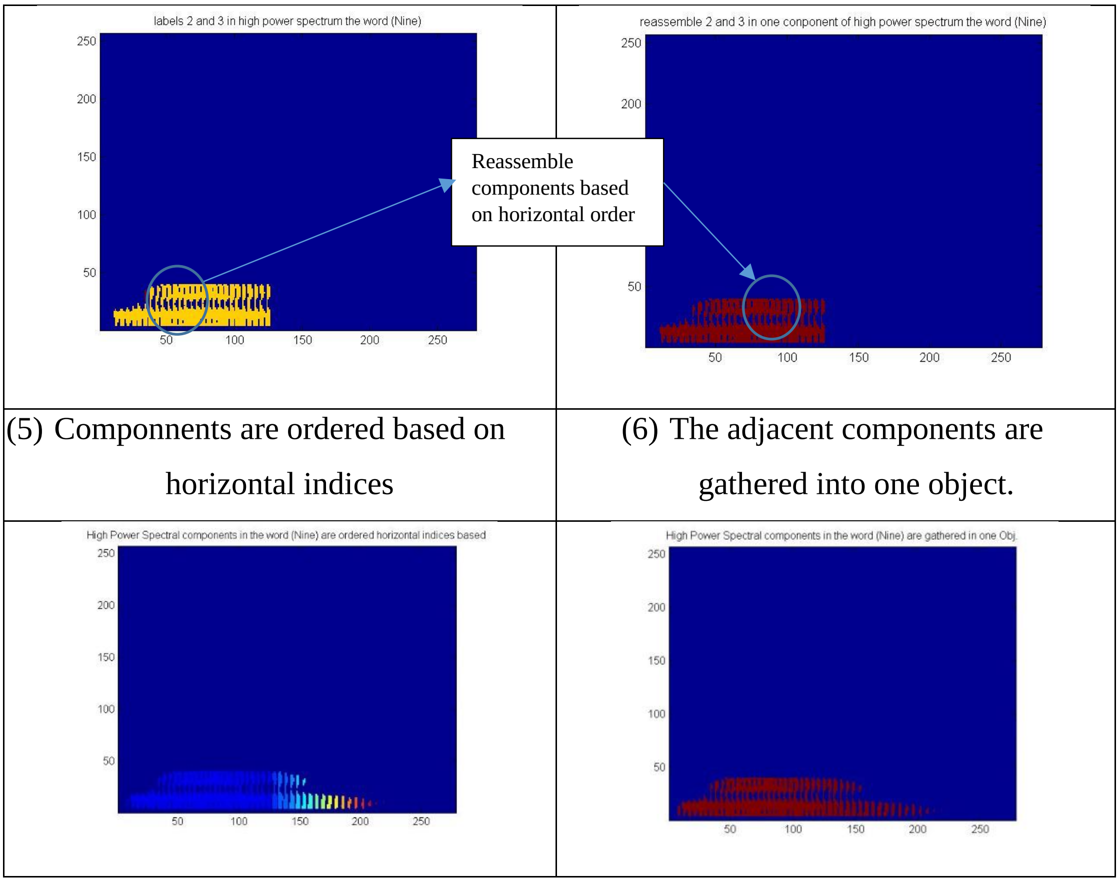
Figure 48 shows the SSI patterns of the word 'seven'.

4.9 General algorithm for segmentation of SSIFS and SSIRS patterns in the SSI

Since the SSI$_{Th}$ binary image is available, further MatLab image processing functions can be used. The general algorithm is based on finding connected components in the SSI$_{Th}$. The MatLab function “bwconncomp” returns the connected components in a binary image. However, it may return objects as connected components which are located linked to each other as shown in part 3 of Figure 49. Actually, we are interested in gathering connected components within objects based on their horizontal order indices since, this gives the length of the relevant object (FFP pattern). Therefore, there should be mapping of the connected components into the horizontal component indices. Then, based on the distance between the horizontal components indices it can be decided how many relevant objects in the SSI$_{Th}$ there are.

From now on, when an index component is mentioned, this means the horizontal component indices. The adjacent index components are formed around an object. Then, the MatLab function “regionpriorps” can be applied to measure properties of the identified objects in the SSI$_{Th}$.

(1) Input ‘nine’ 	(2) 
Labeling the connected components in the binary image of the high PSD in the word ‘nine’.	
(3) e.g. lableling results of indices (2 and 3)	(4) mapping components based on horizontal order indices



(5) Componnents are ordered based on horizontal indices

(6) The adjacent components are gathered into one object.

Figure 49 shows steps in creating a relevant object of connected components based on the identified horizontal component indices.

4.9.1 Composition the SSIFS pattern

The SSIFS is composed of the FFP and the SFFP. Then the general algorithm can summarised by the following steps:

1. The FFP Segmentation: applying the “regionpriorps” on identified object regions in the SSI gives information such as the bounding box dimension of the objects. The number of FFP objects existing in an SSI can be obtained by scanning the distance between two consecutive indices of an object. Then, two objects exist if the distance is bigger than 15 units (where 15 is an estimated a typical dimension). Therefore, the number of the FFPs is equal to the number of distances greater than 15 plus one. These steps have been applied to the SSI of the word ‘seven’ (which contains two of the FFPs) and the result is displayed in Figure 50. The algorithm demonstrates successful distinguishing of the FFP patterns.

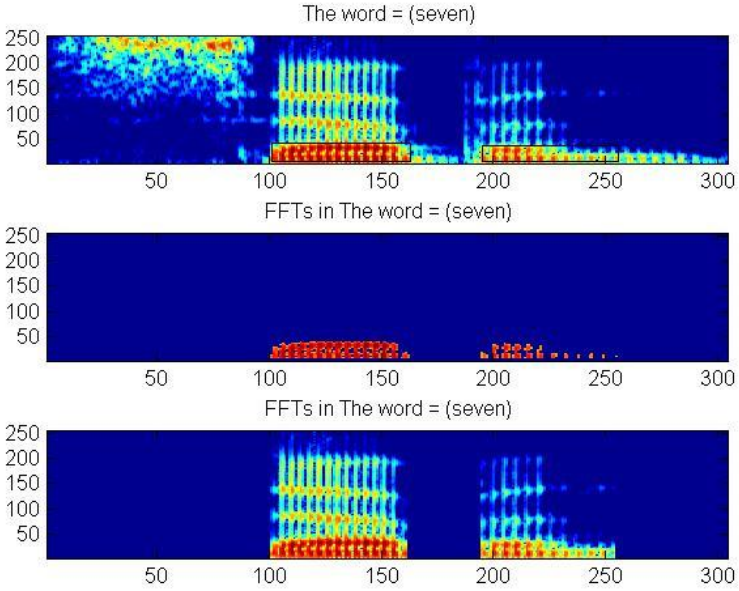


Figure 50 demonstrates the FFP pattern segmentation in the SSI of the word ‘seven’.

2. SSIFS segmentation: the next step is the approximate segmentation of the SSIFS pattern. Let us first name the initial segmentation processing results. To do this we go back to the results of segmentation of the FFP pattern pixels in the SSI of the word ‘seven’ which is illustrated in Figure 51. The segmented FFP pattern is denominated by SSI_{Th1}, which is at the first threshold, and the remainder of the patterns are denominated by SSI_{Rem1}. The process is applied to the SSI, which is denominated by $SSI_{Seg1.}$

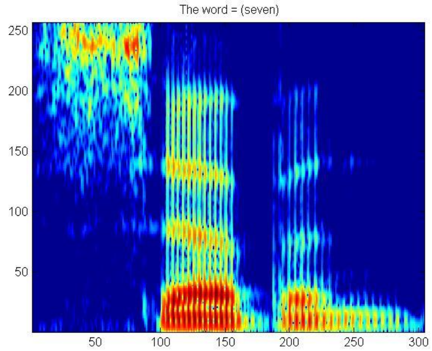
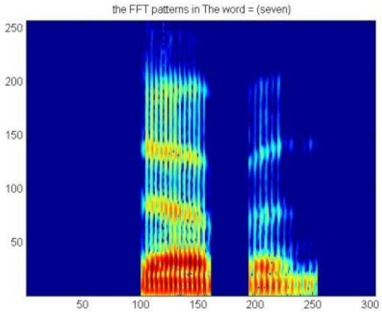
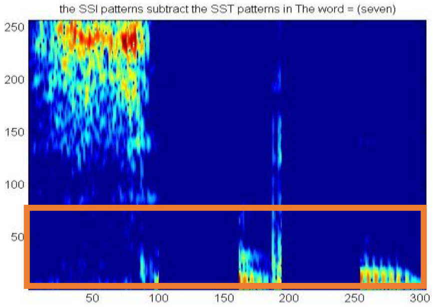
 SSI$_{\text{Seg1}}$	
The FFP patterns in the word ‘seven’ which is the SSI$_{\text{Th1}}$.	The remainder patterns in the SSI of the word ‘seven’ after subtracting the FFP patterns, SSI$_{\text{Rem1}}$.
	

Figure 51 shows the SSI$_{\text{Seg1}}$ results of processing of the word ‘seven’.

3. SFFP segmentation: the next target is to capture the SFFP patterns from the SSI$_{\text{Rem1}}$. The SFFP is the high PSD content in the SSI$_{\text{Rem1}}$. Moreover, the first formant location contains the higher PSD in the SSI. Thus, the same steps can be applied as to the SSI$_{\text{Seg1}}$ with a new threshold on the SSI$_{\text{Rem1}}$ to segment the SFFP pattern which is denoted by SSI$_{\text{Seg2}}$. The threshold cut-off for the SFFP is less than the FFP, which could be within the range of the SSIRS pixels. Therefore, the SSI$_{\text{Rem1}}$ needs to perform the necessary action to avoid the area of the SSIRS in the SSI$_{\text{Rem1}}$ as shown in part 2 of Figure 52 to obtain what is denoted by SSI$_{\text{Th2,}}$ as shown in part 3 in Figure 52. The SSI$_{\text{Th2}}$ contains only equivalent pixels of high PSD in the location of the first format in the SSI$_{\text{Rem1}}$. These locations are approximate locations of the SFFP in an SSI.

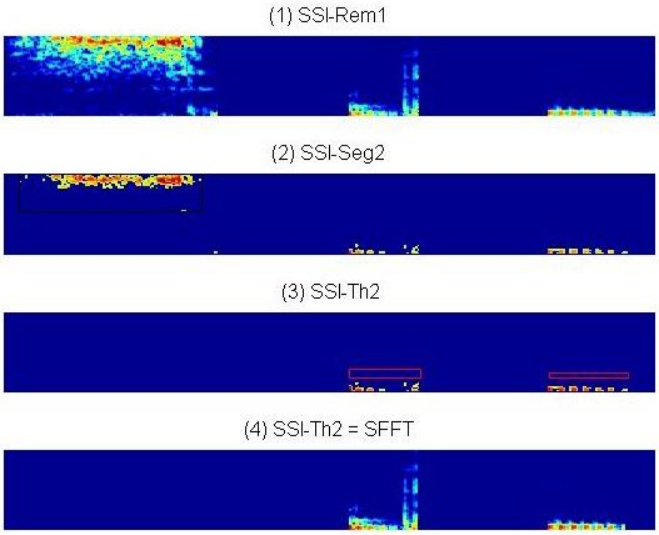


Figure 52 illustration the segmentation of the SFFP patterns in the SSI of the word ‘seven’.

4. Composing the SSIFS: finally, by adding the result of Figure 48 (containing the FFP pattern) with the result of Figure 52 (containing the SFFP) we can compose the SSISF pattern in the SSI of the word ‘seven’ as shown in part 4 of Figure 53. By this recognition process, the SSIFS in the SSI leads us to be able to distinguish the SSIRS in the SSI.

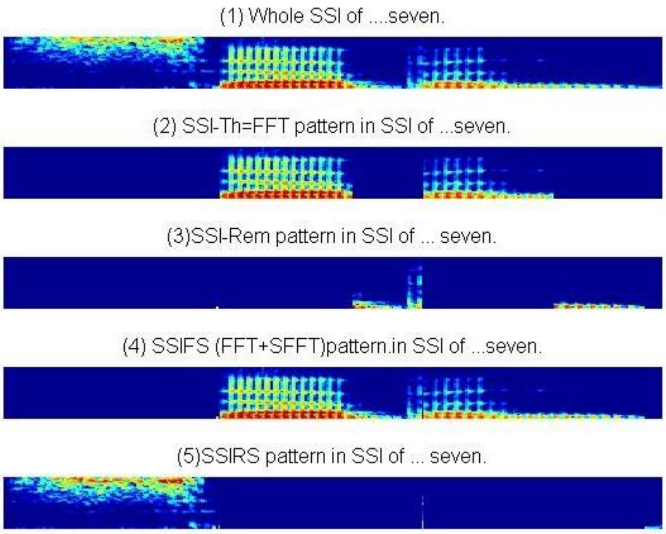


Figure 53 shows the SSI of the word ’seven’ separated from the SSIFS-seven and SSRS-seven.

4.9.2 Steps in general algorithm segmentation of the SSI based on the L1-1 category

The main steps of distinguishing between the SSIFS and the SSIRS patterns of the SSI of a word are shown summarised in Figure 54.

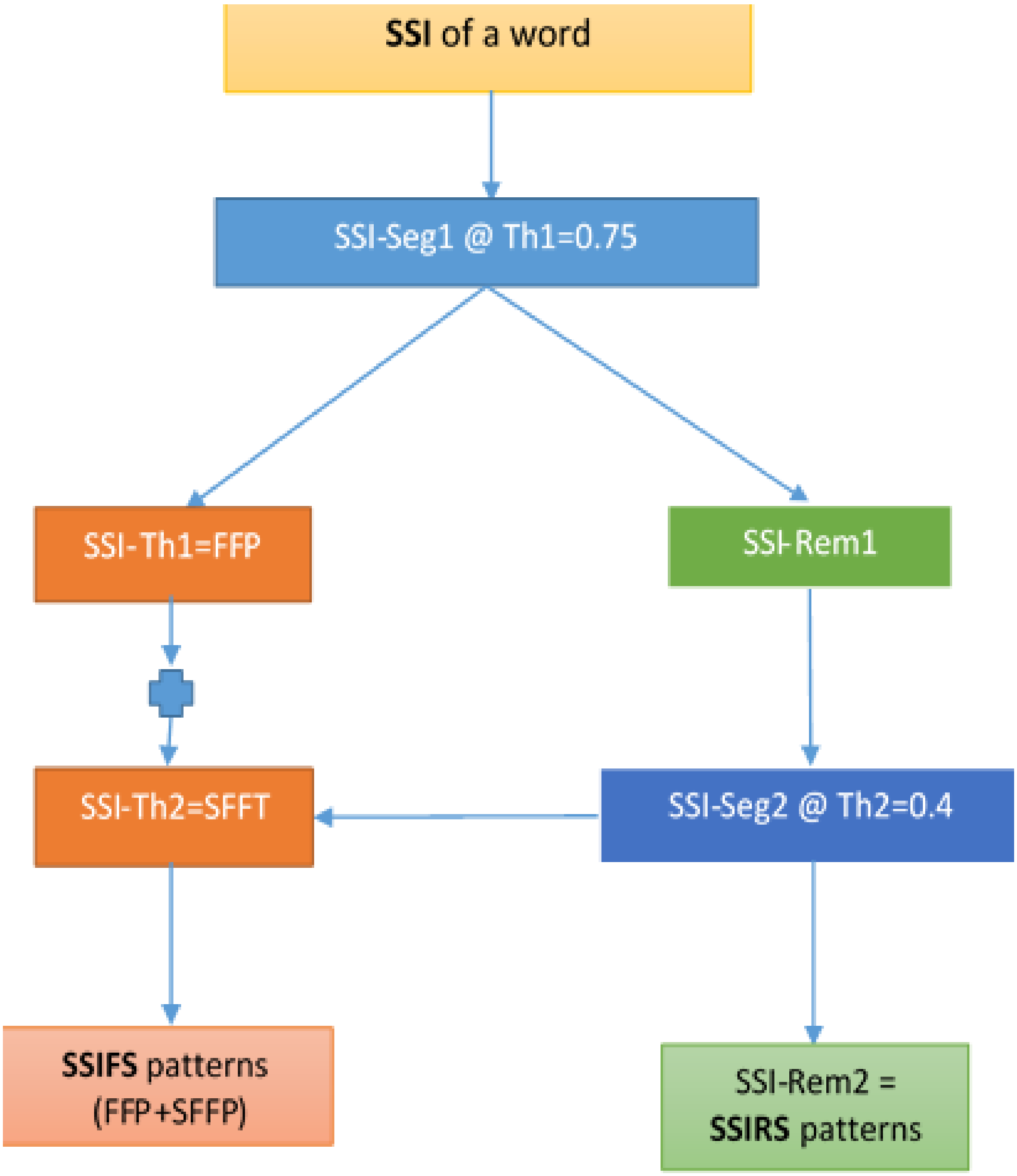


Figure 54 shows the main steps of segmenting the SSIFS and the SSIRS patterns in the SSI of a word.

4.10 Testing and results of the general algorithm segmentation of the SSI

The general algorithm for distinguishing between the SSIFS and the SSIRS patterns of a word has been applied on the lexicon words. Since it is difficult to display all the lexicon word results, three random samples of each word (the lexicon contains 15 or 7 samples for each word) has been displayed in Figure 55 to Figure 62.

The algorithm demonstrates good accuracy, with zero error in distinguishing between the SSIFS and SSIRS patterns for the whole lexicon of words. In addition, there is not a big difference in the capturing of the SSIFS and the SSIRS patterns for the three samples. Indeed, by optimising thresholds (i.e. Th-M) the algorithm can achieve perfect capturing of the SSIFS from the SSI. However, that is not the focus of this study. This study wishes to demonstrate that speech recognition of words by matching SSIs should be considered for ASR applications.

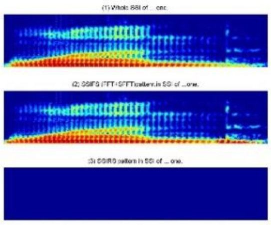
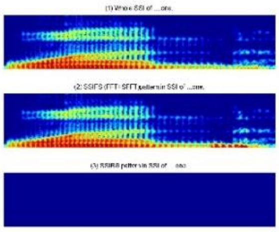
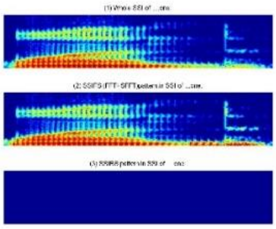
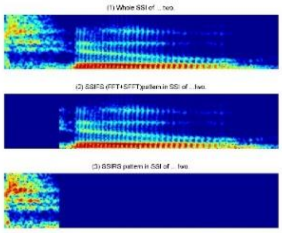
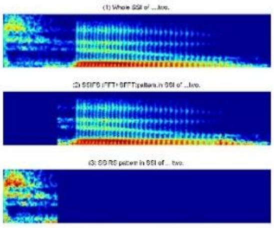
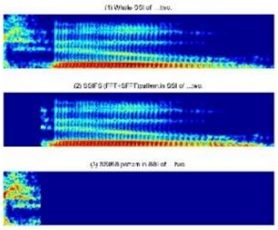
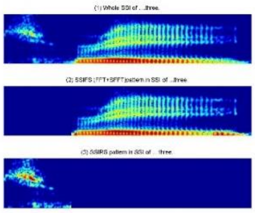
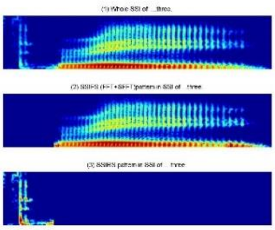
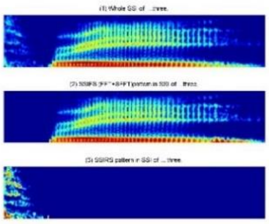
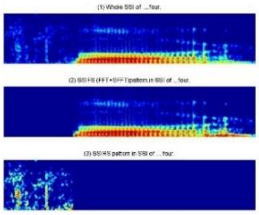
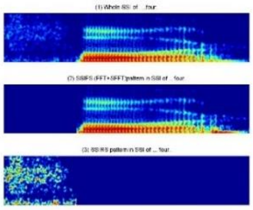
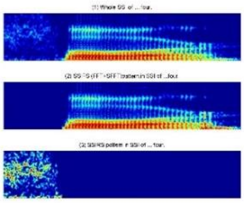
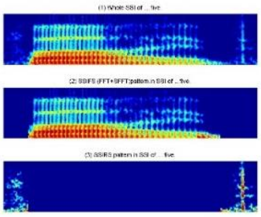
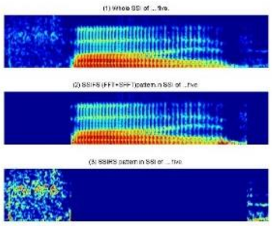
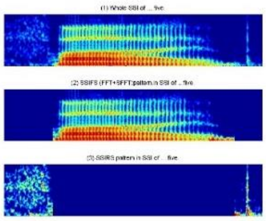
Word index	Sample (1)	Sample (2)	Sample (3)
(1) One			
(2) Two			
(3) Three			
(4) Four			
(5) Five			

Figure 55 discrimination between the SSIFS and the SSIRS by using general algorithm segmentation of the SSI for words: ‘one’, ‘two’, ‘three’, ‘four’, and ‘five’.

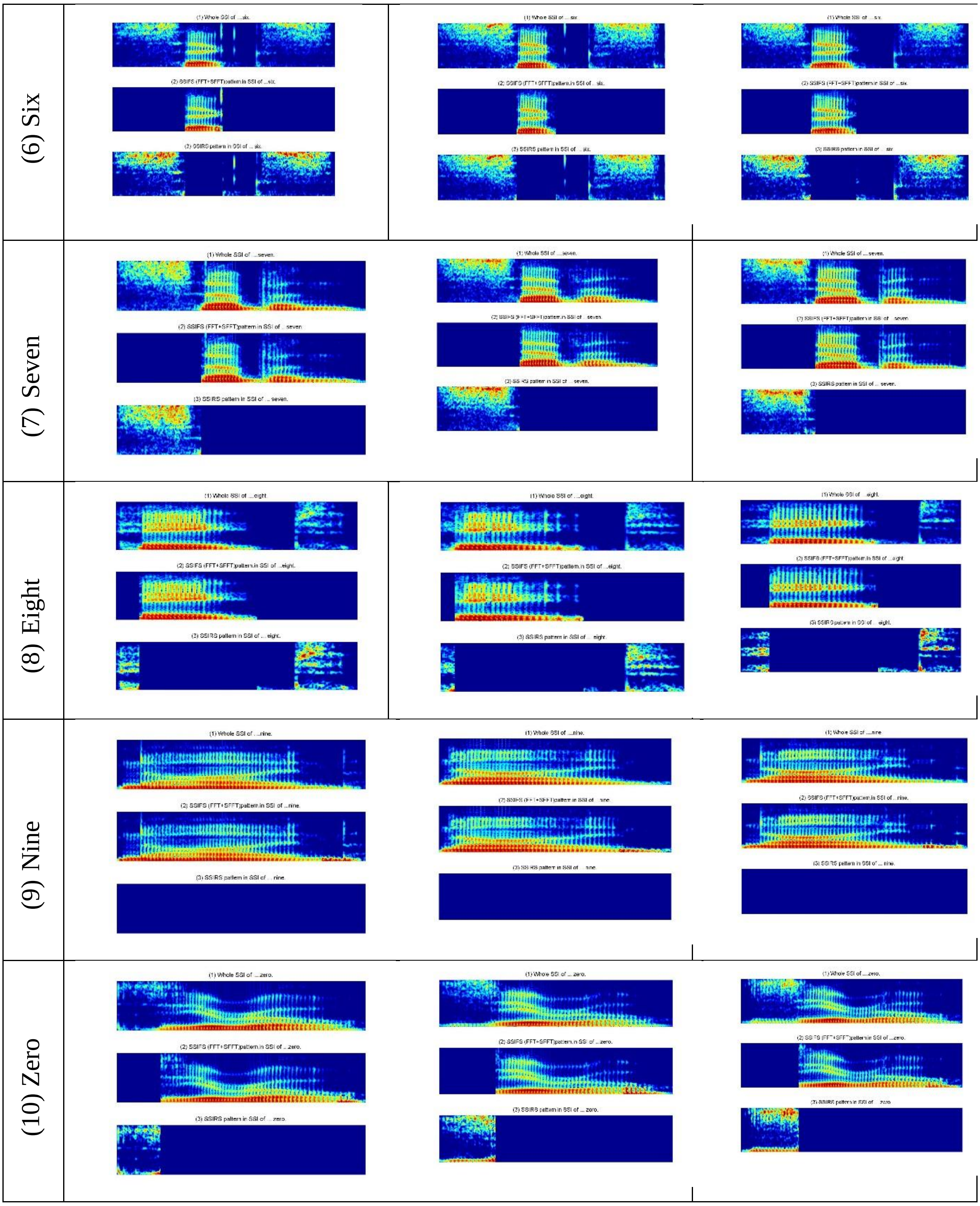


Figure 56 discrimination between the SSIFS and the SSIRS by using general algorithm segmentation of the SSI for words; ‘six’, ‘seven’, ‘eight, ‘nine, and ‘zero’.

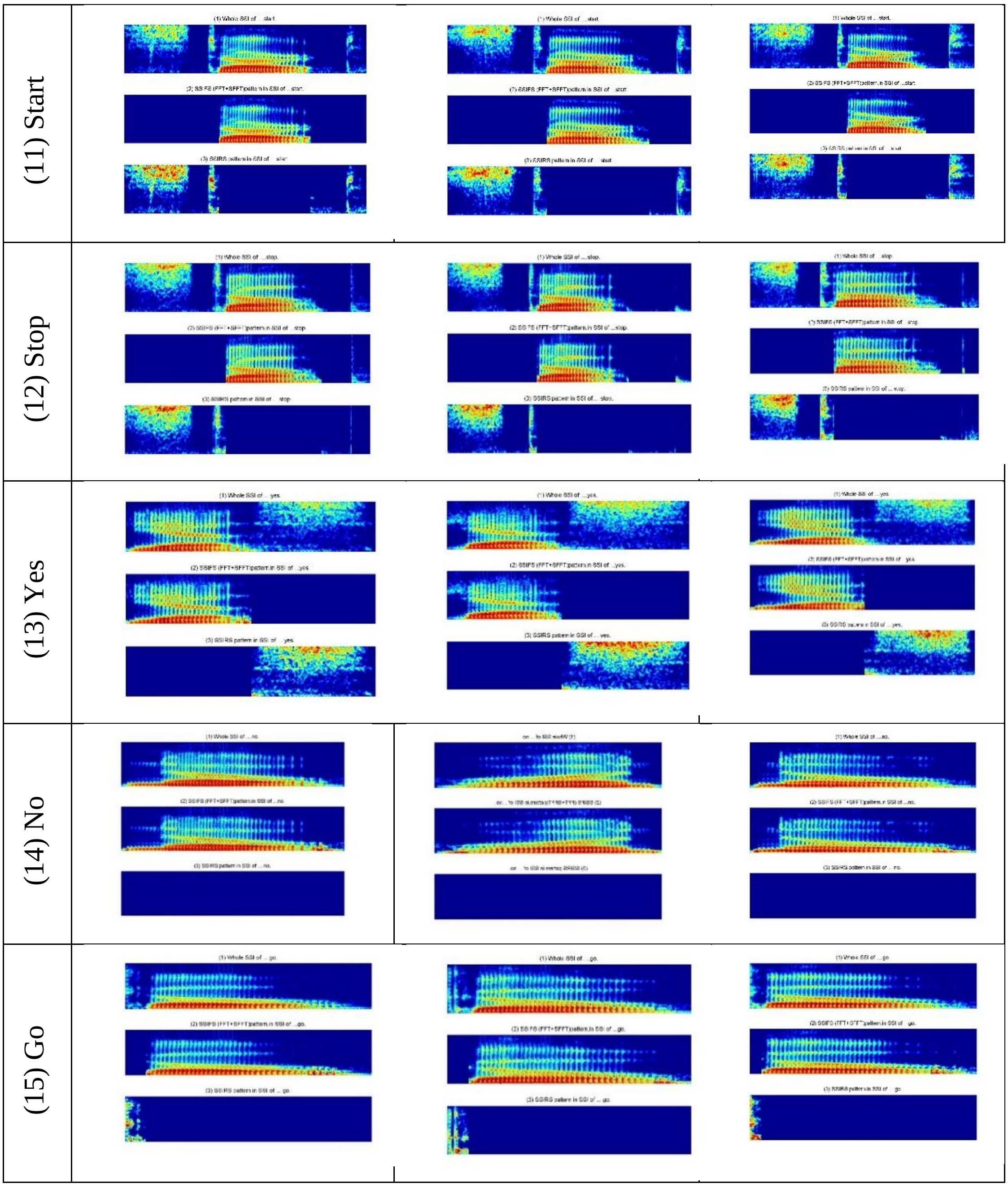


Figure 57 discrimination between the SSIFS and the SSIRS by using general algorithm segmentation of the SSI for words; ’start’, ‘stop’, ‘yes’, ‘no, and ‘go’.

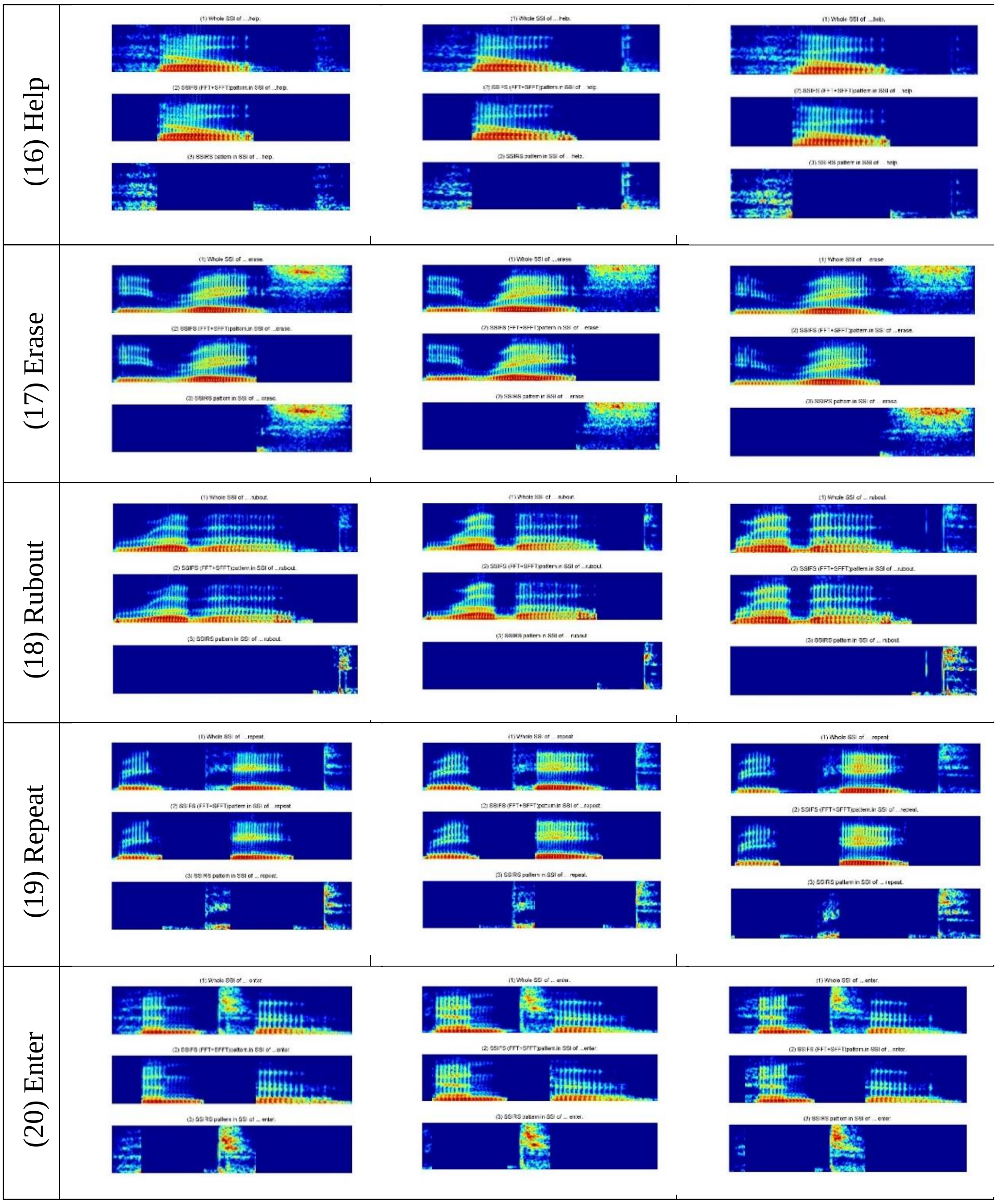


Figure 58 discrimination between the SSIFS and the SSIRS by using general algorithm segmentation of the SSI for words: ‘help’, ‘erase’, ‘rubout’, ‘repeat’, ‘enter’.

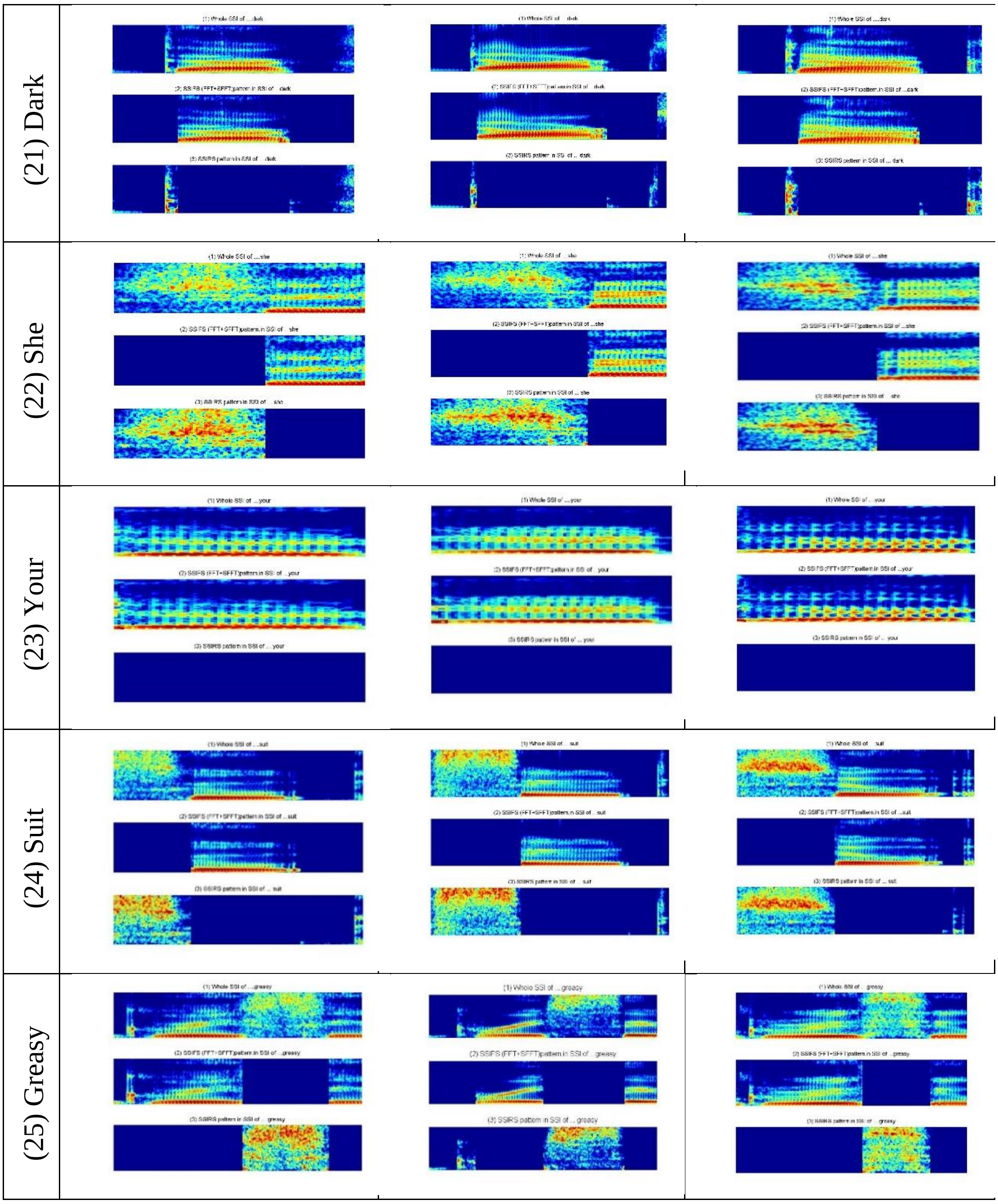


Figure 59 discrimination between the SSIFS and the SSIRS by using general algorithm segmentation of the SSI for words: ’dark’, ‘she’, ‘your’, ‘suit’, and ‘greasy’.

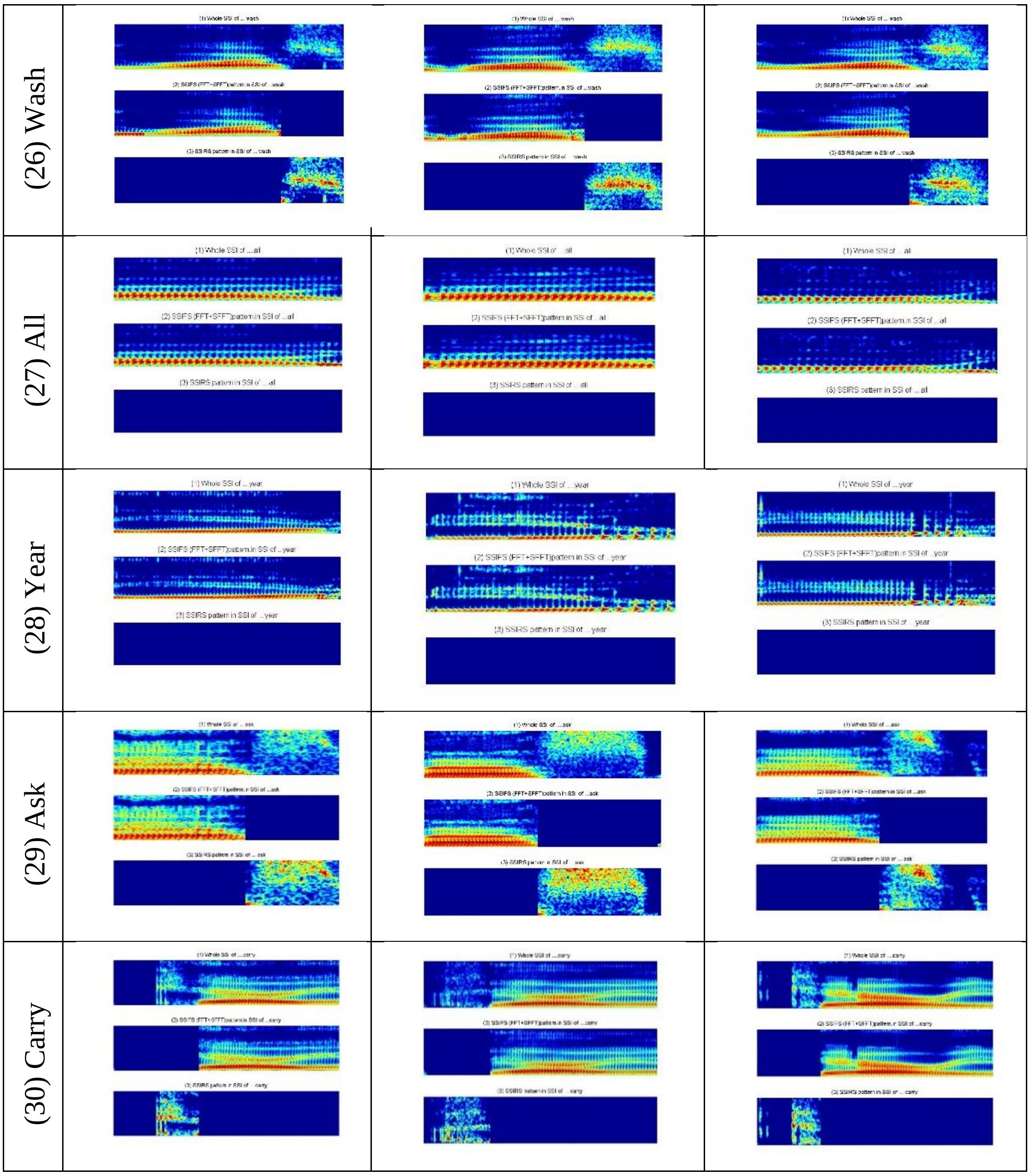


Figure 60 discrimination between the SSIFS and the SSIRS by using general algorithm segmentation of the SSI for words; ‘wash’, ‘all’, ‘year’, ‘ask’, and ‘carry’.

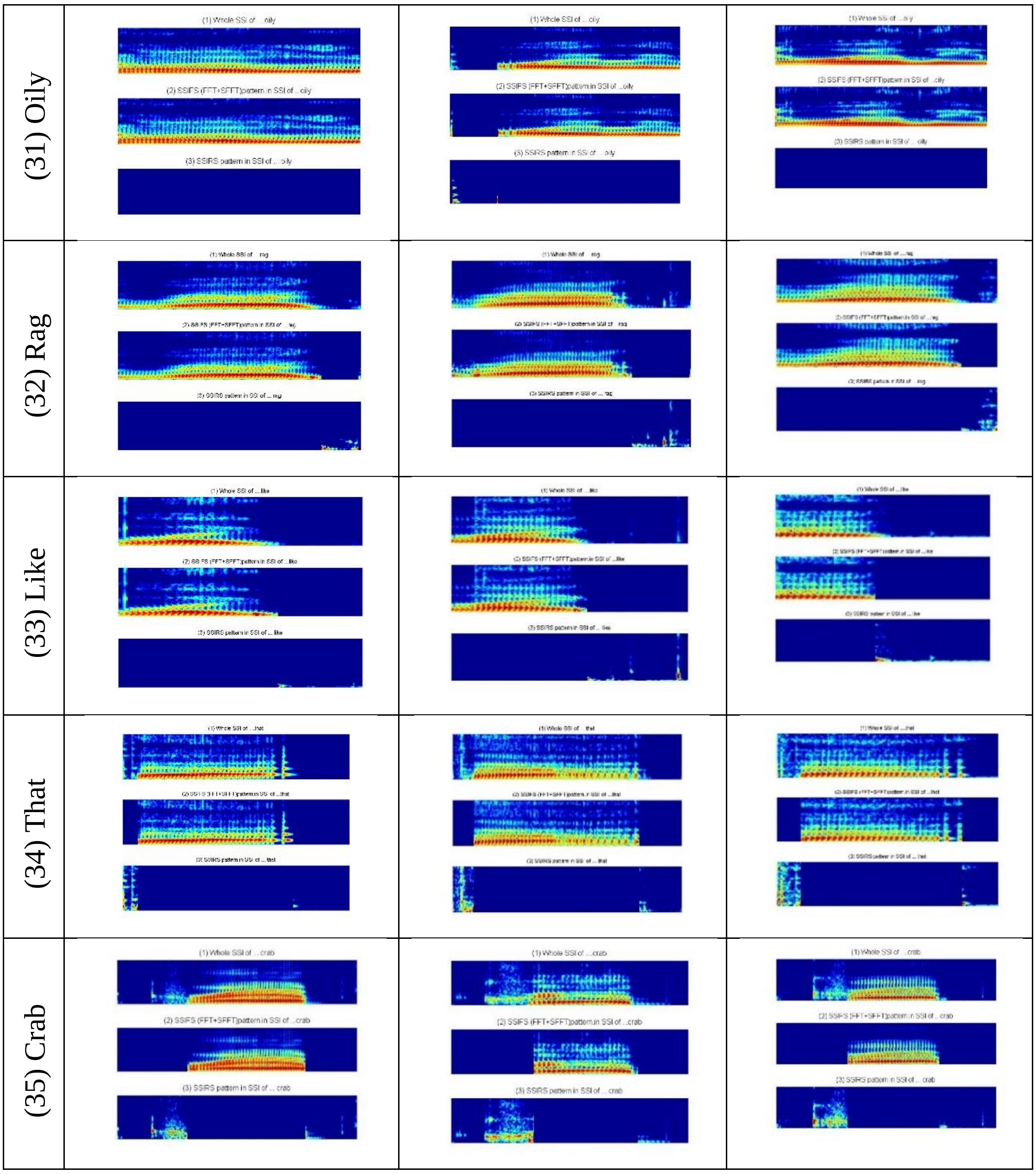


Figure 61 discrimination between the SSIFS and the SSIRS by using general algorithm segmentation of the SSI for words; ‘oily’, ‘rag’, ‘like’, ‘that’, and ‘crab’.

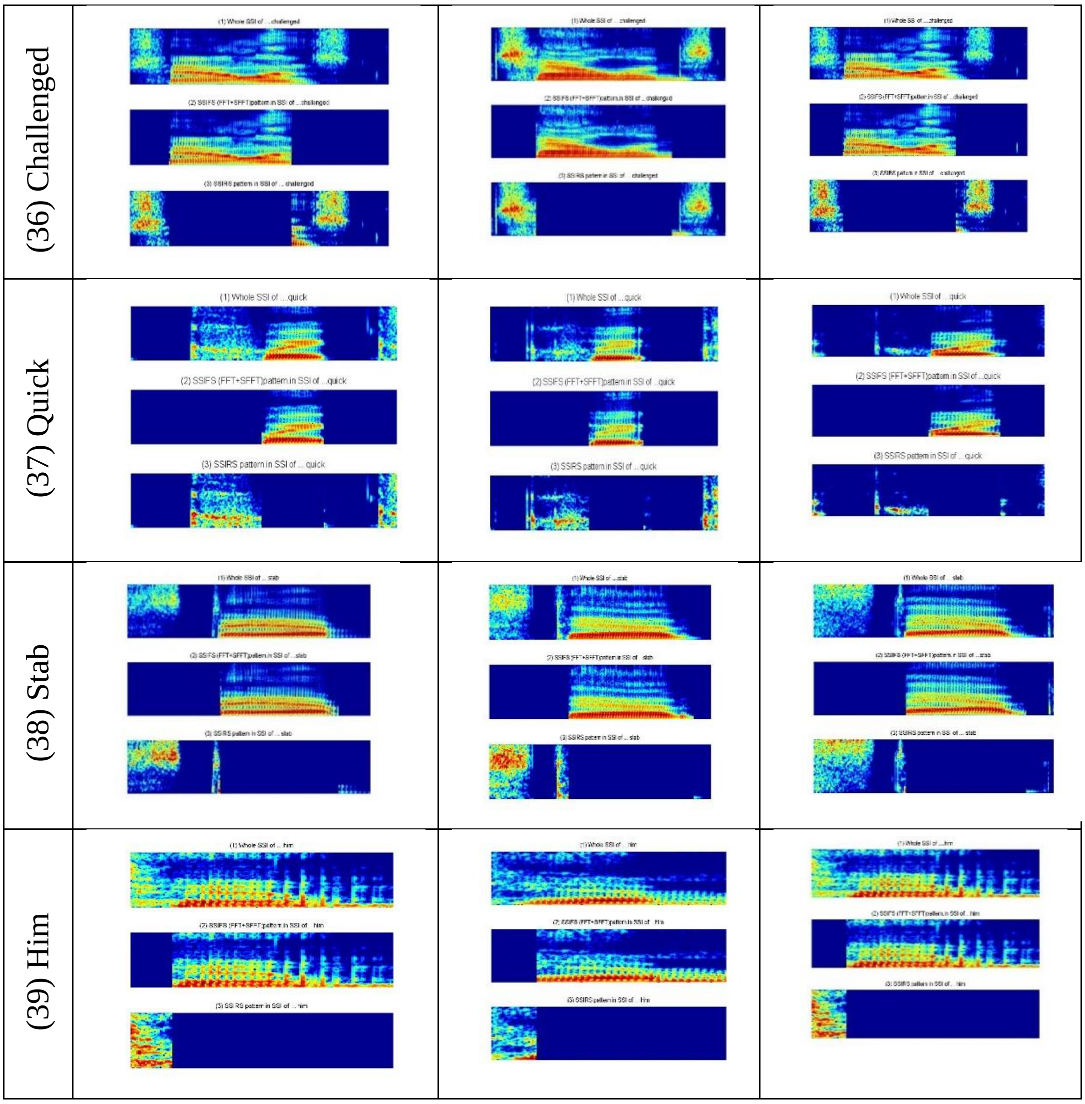


Figure 62 discrimination between the SSIFS and the SSIRS by using general algorithm segmentation of the SSI for words; ‘challenged’, ‘quick’, ‘stab’, and ‘him’.

4.11 Classification of the lexicon words by the general algorithm

If the symbol A is given to the SSIFS and the symbol B to the SSIRS, the lexicon words can be classified into codes A and B by the general algorithm. The codes are then: A, BA, AB, BAB, ABAB, ABA and BABA. These, then appear in the lexicon words as follows. The code A appears in 7 words which are: ‘one’, ‘nine’, ‘no’, ‘your’, ‘all’, ‘year’, and ‘oily’. The code BA is appears in 9 words which are ‘two’, ‘three’, ‘four’, ‘seven’, ‘zero’, ‘go’, ‘she’, ‘carry’, and ‘him’. The code AB is appears in 7 words which are: ‘yes’, ‘erase’, ‘rubout’, ‘wash’, ‘ask’, ‘like’, and ‘rag’. The code BAB appears in 13 words which are: ‘five’, ‘six’, ‘eight’, ‘stat’, ‘stop’, ‘help’, ‘dark’, ‘suit’, ‘that’, ‘crab’, ‘challenged’, ‘quick’, and ‘stab’. The code is appeared in ABAB 1 word which is ‘repeat’. The code ABA appears in one word which is ‘creasy’. The last code BABA is appeared 1 word which is ‘enter’. The codes are shown in Figure 63 to Figure 68.

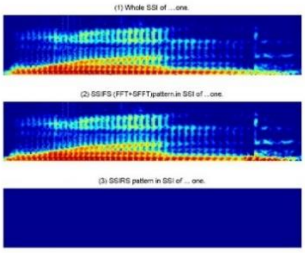
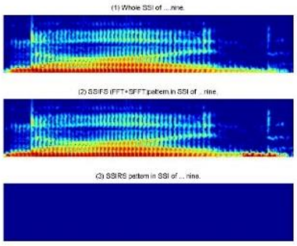
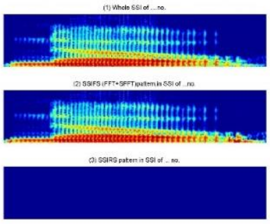
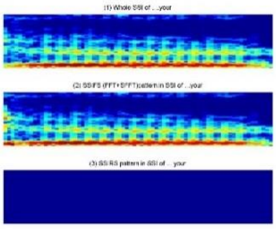
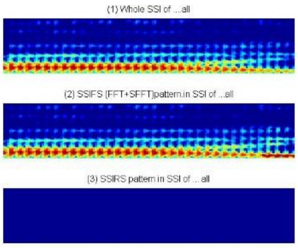
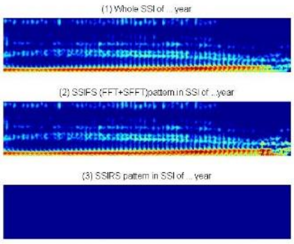
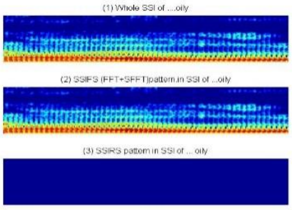
The words of code **A** (7 words)	One	Nine
	 (1) Whole SSI of ....one. (2) SSIFS (FFT+SFFT)pattern.in SSI of ...one. (3) SSIRS pattern in SSI of ... one.	 (1) Whole SSI of ....nine. (2) SSIFS (FFT+SFFT)pattern.in SSI of ...nine. (3) SSIRS pattern in SSI of ... nine.
	No	Your
	 (1) Whole SSI of ....no. (2) SSIFS (FFT+SFFT)pattern.in SSI of ...no. (3) SSIRS pattern in SSI of ... no.	 (1) Whole SSI of ...your (2) SSIFS (FFT+SFFT)pattern.in SSI of ...your (3) SSIRS pattern in SSI of ... your
	All	Year
	 (1) Whole SSI of ...all (2) SSIFS (FFT+SFFT)pattern.in SSI of ...all (3) SSIRS pattern in SSI of ... all	 (1) Whole SSI of ... year (2) SSIFS (FFT+SFFT)pattern.in SSI of ...year (3) SSIRS pattern in SSI of ... year
	Oily	
	 (1) Whole SSI of ....oily (2) SSIFS (FFT+SFFT)pattern.in SSI of ...oily (3) SSIRS pattern in SSI of ... oily	

Figure 63 shows the lexicon words of code **A**.

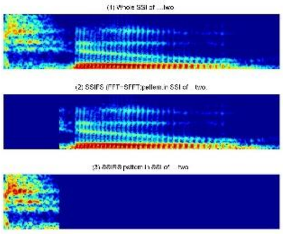
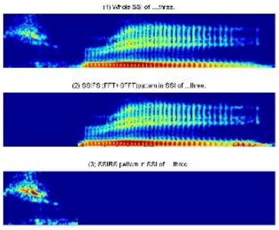
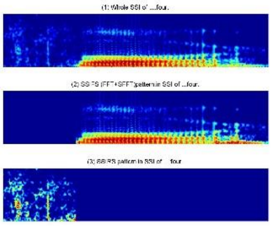
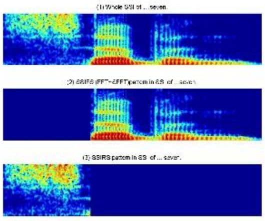
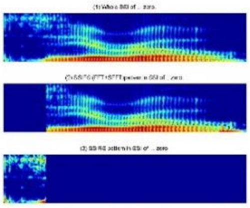
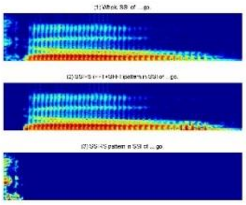
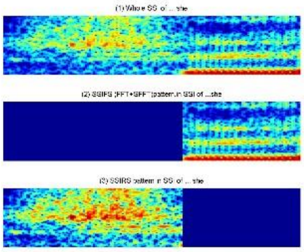
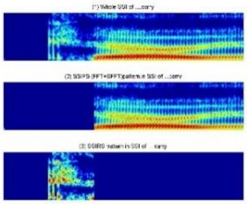
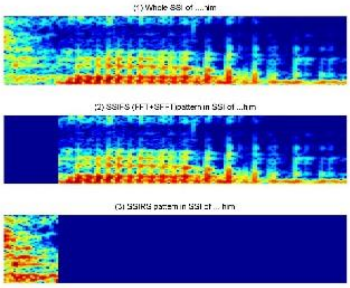
The words of code **BA** (9 words)	Two	Three
		
	Four	Seven
		
	Zero	Go
		
	She	Carry
		
	Him	
		

Figure 64 shows the lexicon words of code **BA**.

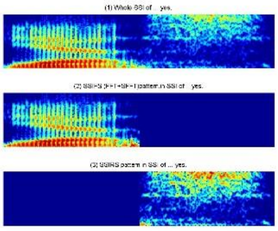
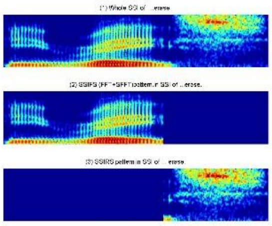
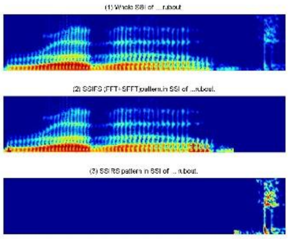
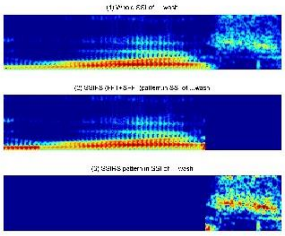
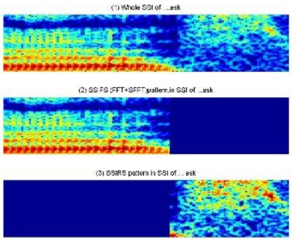
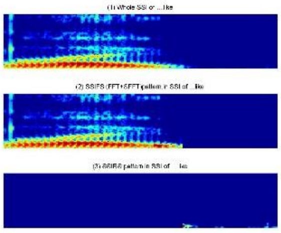
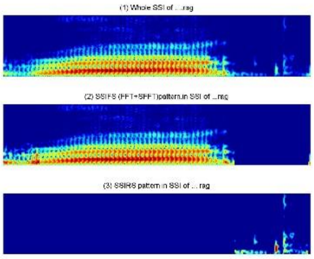
The words of code **AB** (7 words)	Yes	Erase
		
	Rubout	Wash
		
	Ask	Like
		
	Rag	
		

Figure 65 shows the lexicon words of code **AB**.

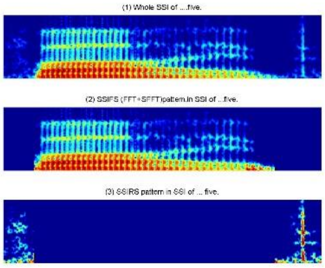
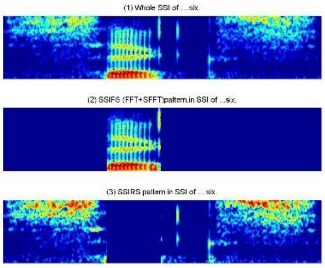
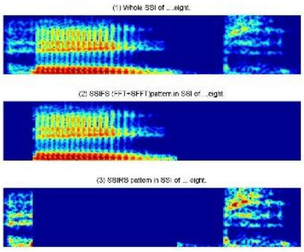
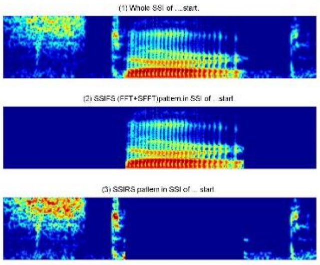
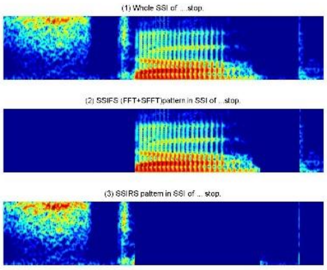
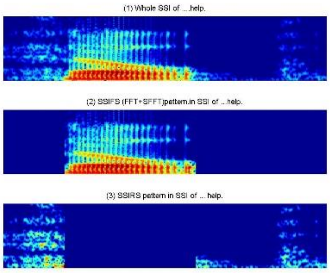
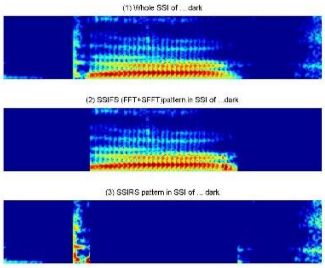
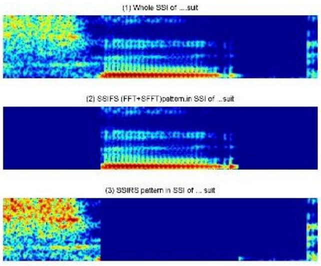
The words of code **BAB** Code _1 (8 words)	Five	Six
		
	Eight	Start
		
	Stop	Help
		
	Dark	Suit
		

Figure 66 shows the lexicon words of code **BAB**, group (1).

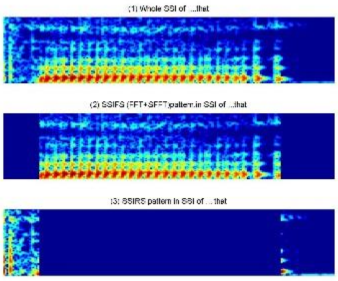
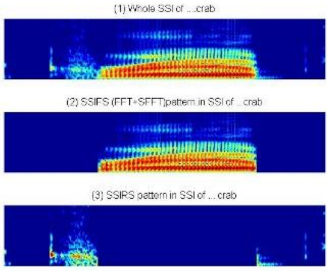
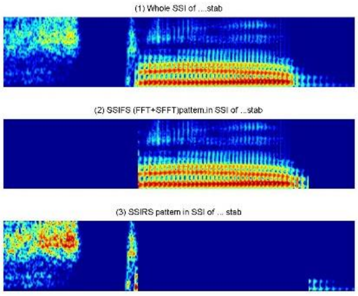
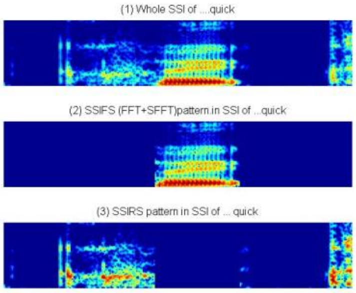
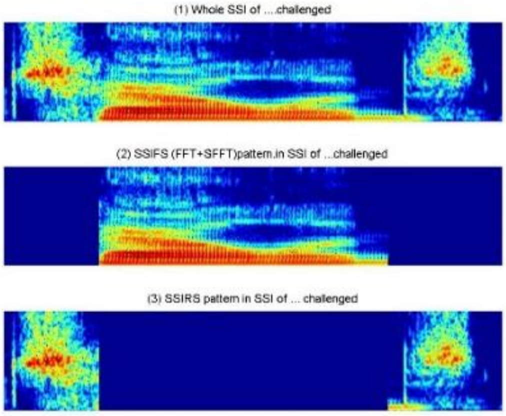
The words of code **BAB** _2 (5 words)	That	Crab
		
	Challenged	Quick
		
	Stab	
		

Figure 67 shows the lexicon words of code **BAB**, group (2).

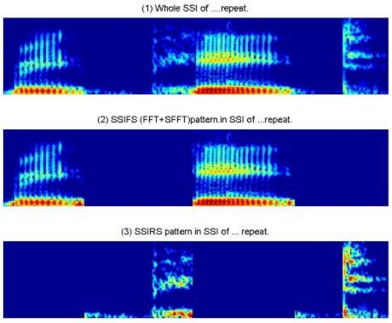
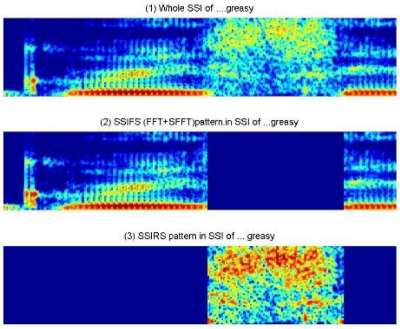
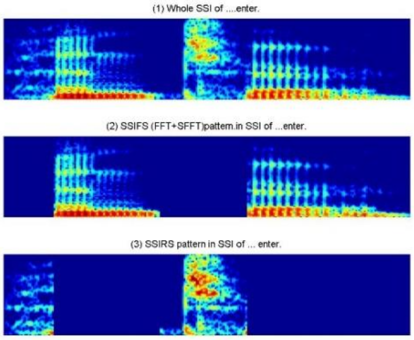
The words of code **ABAB** (1)	Repeat
	
The words of code **ABA** (1)	Greasy
	
The words of code **BABA** (1)	Enter
	

Figure 68 shows the lexicon word of code **BABA**, **ABA** and **ABAB**.

Then, the lexicon word can be parsed in a tree classification of the set codes of A and B as shown in Figure 69. The classification tree is based on the L1-1 category. The tree classification reduces the number of matches of the SSI of the word input with the lexicon words.

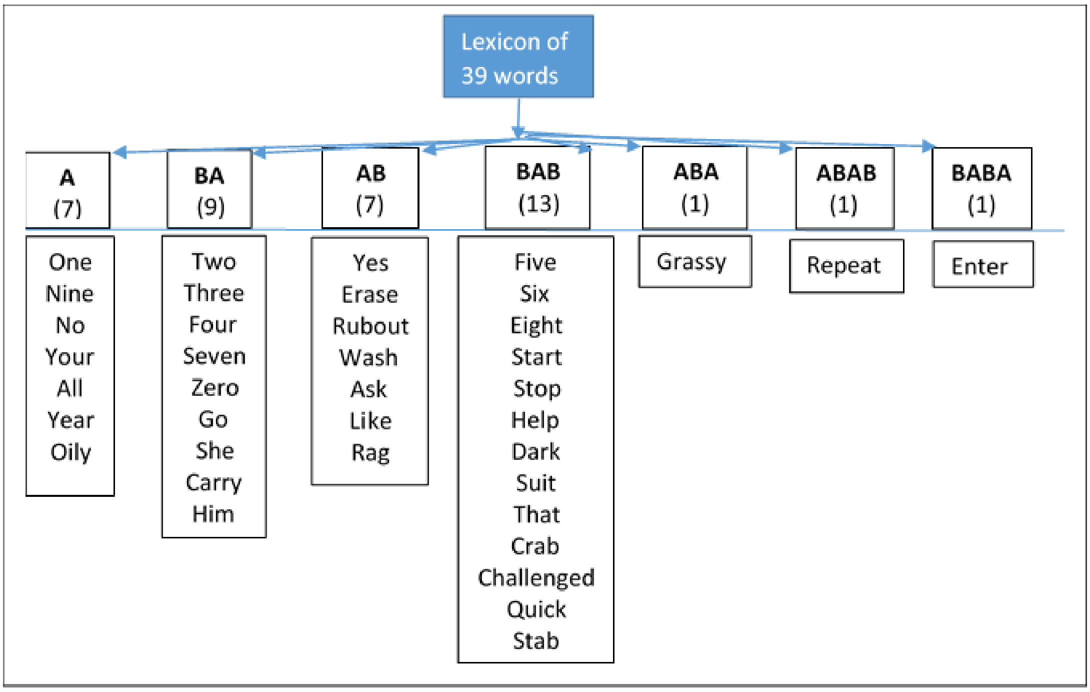


Figure 69 the tree classification codes of the lexicon words based on the L1-1 category.

4.12 Parsing input word and matching decision based on the L1-1 category

Parsing a word into the SSI patterns based on the L1-1 category can help to make the SSI recognition decision more accurate and reduce the number of false matches of the input word with lexicon words. Indeed, the matching is only for the SSIFS of the word input and lexicon words. Figure 70 shows the general steps for recognition of a word by parsing the word input based on the L1-1 category.

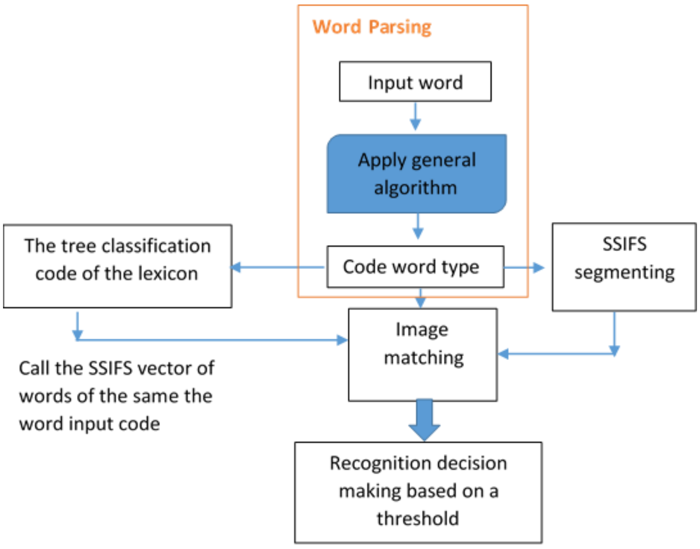

Word Parsing
Input word
Apply general algorithm
Code word type
The tree classification code of the lexicon
SSIFS segmenting
Image matching
Call the SSIFS vector of words of the same the word input code
Recognition decision making based on a threshold

Figure 70 shows the SSI recognition by parsing and matching of the SSI patterns.

4.13 Lexicon word recognition

4.13.1 Test No.3

An unsuccessful recognition example of recognition of the word ‘two’ by NCC and the MACH filter are displayed in Figure 41 and Figure 43, respectively. The same input word ‘two’ has been retested by NCC matching of the SSI based on the L1-1 category. The results are displayed in Figure 71, which shows successful recognition of the word ‘two’. Indeed, the lexicon has 15 samples of word the ‘two’, which is represented in row two in Figure 71. The successful recognition of the word ‘two’ is indicated by the minimum NCC of index 15 NCC_{15} (=0.8319) and is located in row 2 column 13 where the maximum value is located, while the NCC_{16} (= 0.7173) is located in row 11 column 15 which is the word ‘start’. Then, the word ‘two’ of index 13 (the minimum NCC of the word ‘two’ in the previous test) has been selected as input to be matched with the lexicon words. The matching is successful for recognition of the word ‘two’, as shown in Figure 72. The NCC_{15} = 0.7973 is located in row 2 column 7, while NCC_{16} (=0.70475) is matched with the word ‘that’ that is located in row 34 column 11.

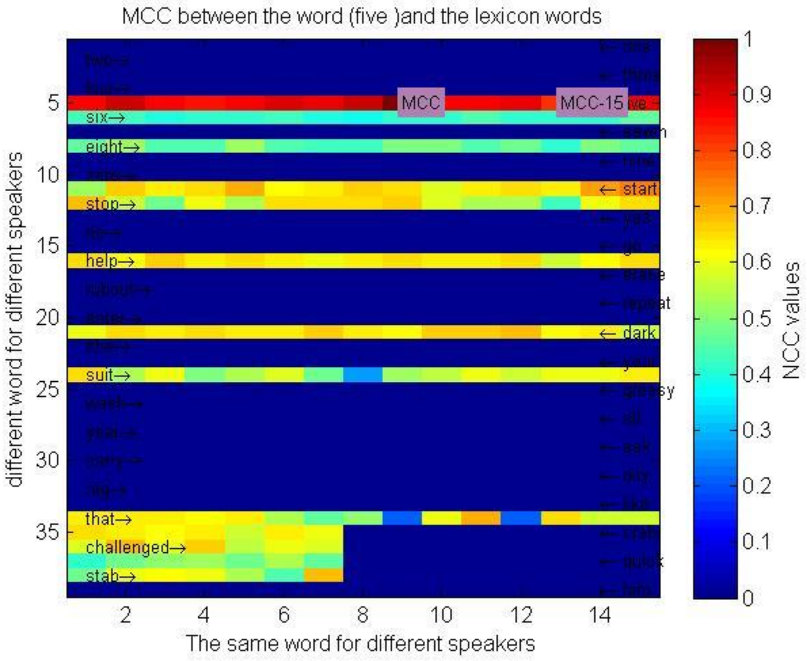


Figure 71 matching the SSI of the word' 'two' of index 9 with the lexicon words in the spatial domain by NCC based on the L1-1 category of the SSI pattern classification algorithm.

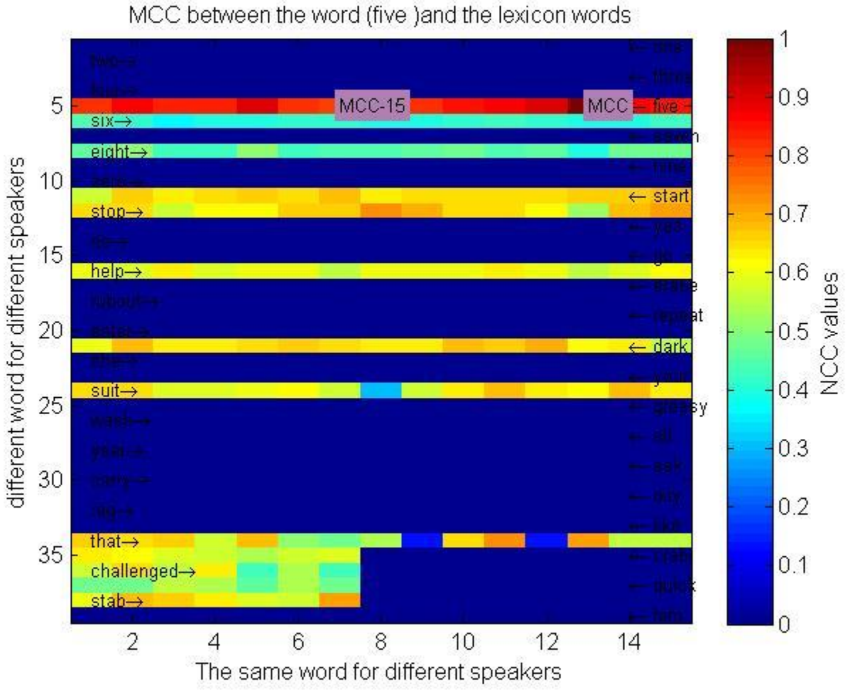


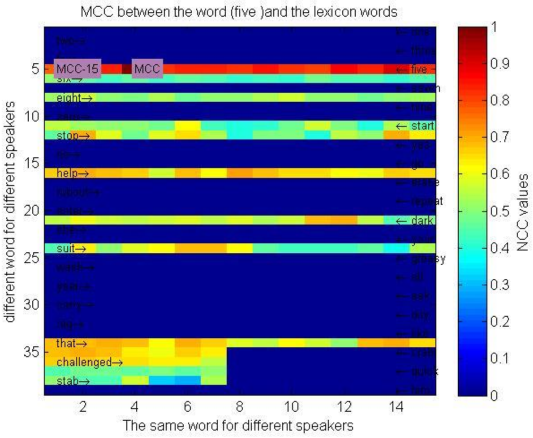
Figure 72 shows matching the SSI of the word' 'two' of index 13 with the lexicon words in the spatial domain by NCC based on the L1-1 category of the SSI pattern classification algorithm.

4.13.2 Test No.4

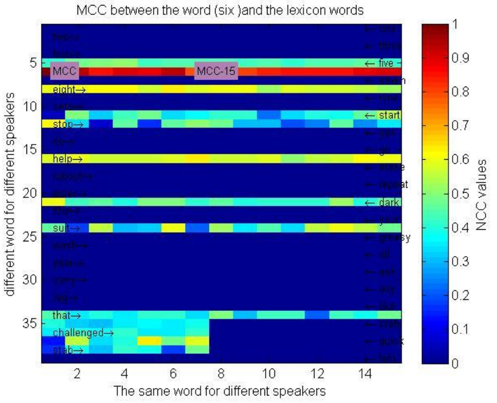
For improved reliability, a test of a random selection words from groups of code: BAB, A, BA, and AB were selected as an input image and matched with lexicon words. The input word was selected from 12/7 samples (the lexicon contains 15/7 samples of the same word uttered by different persons). The results of group BAB words are displayed in Figure 73 and Figure 74 and the result of groups A, BA, and AB are displayed in: Figure 75, Figure 76, and Figure 77 respectively. The remaining groups of codes: ABA, ABAB, and BABA have only one element. Therefore, there is no need for distinguishing them by matching. The test performs successful recognition for all words that is obvious by the high NCC similarity, represented by the red coloured row, which shows the results of matching the word input with other instances of the same word uttered by different persons.

Random words from group of code BAB_ group 1:(6 words)

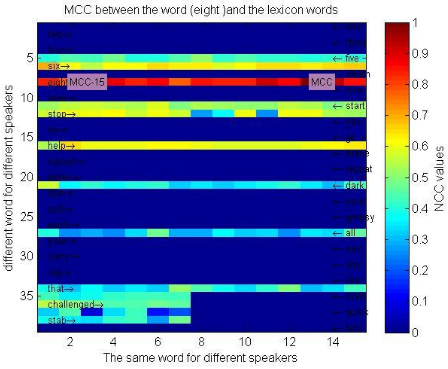
Input word 'five'



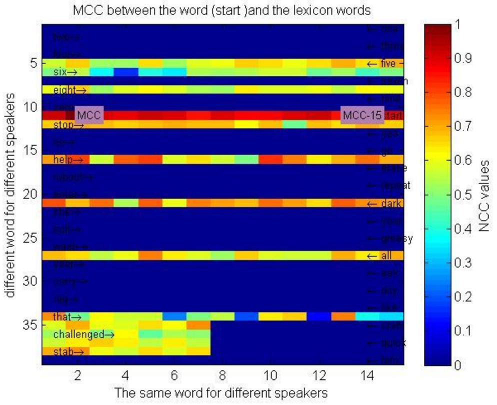
Input word 'six'



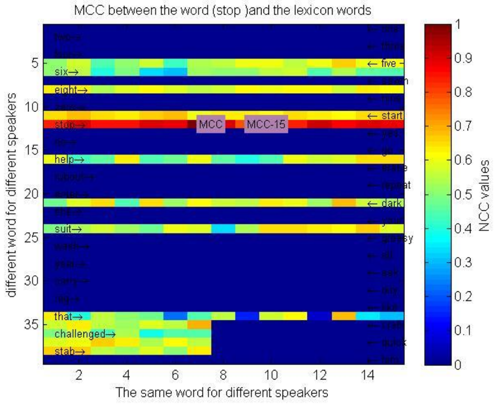
Input word 'eight'



Input word 'start'



Input word 'stop'



Input word 'help'

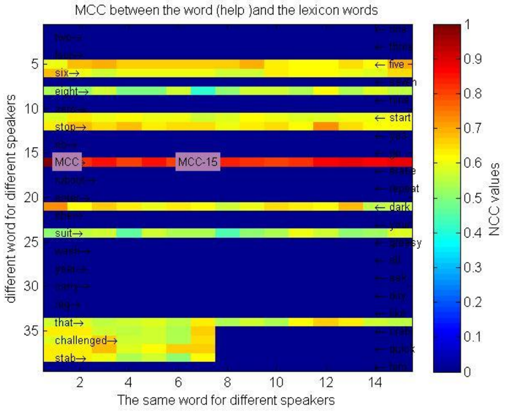


Figure 73 shows successful discrimination of random words from code **BAB_** group 1 by NCC matching of the SSI.

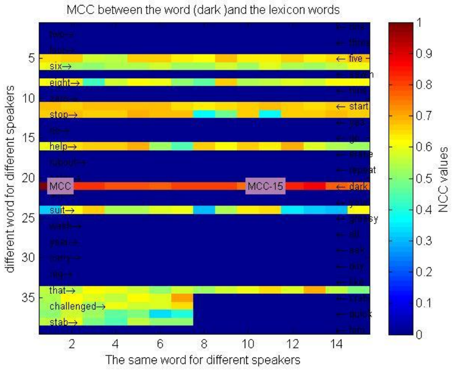
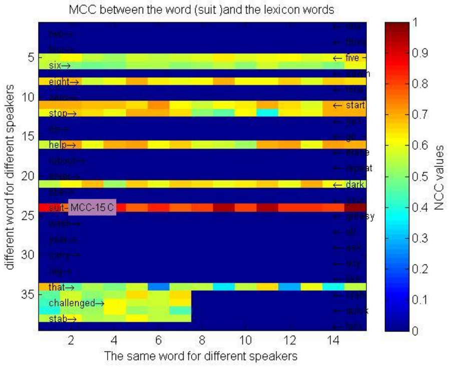
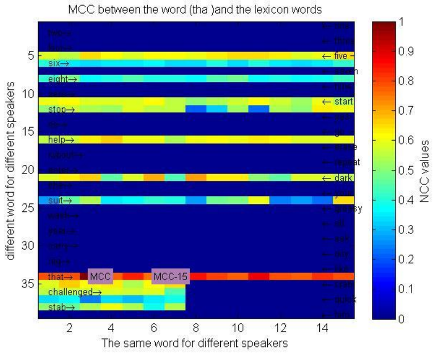
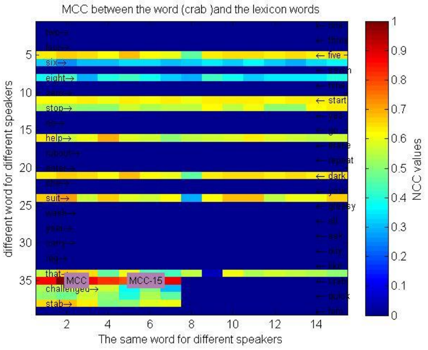
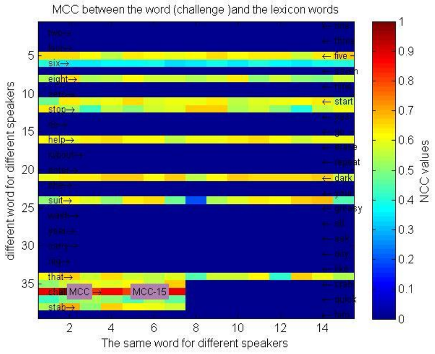
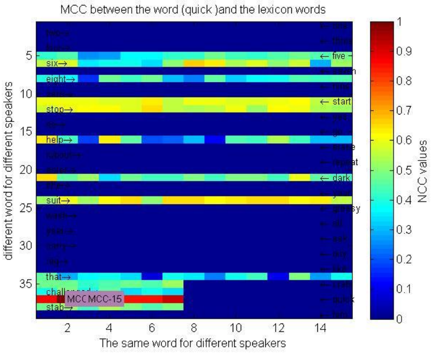
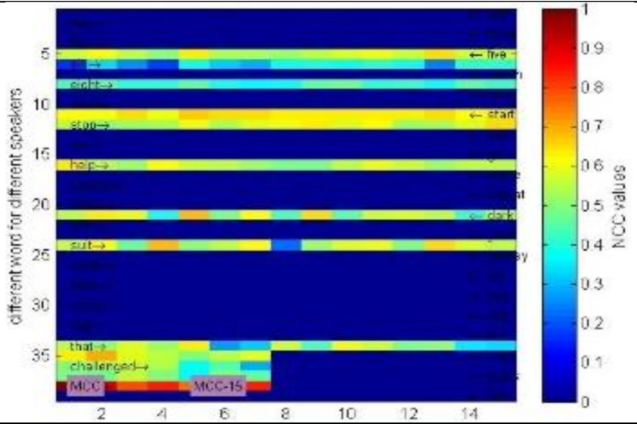
<table>
<tr><th colspan="2">Random words from group code BAB_ group 2: (7 words)</th></tr>
<tr><td>Input word ‘dark’</td><td>Input word ‘suit’</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>Input word ‘that’</td><td>Input word ‘carb’</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>Input word ‘challenge’</td><td>Input word ‘quick’</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2">Input word ‘stab’</td></tr>
<tr><td colspan="2">
</td></tr>
</table>

Figure 74 shows successful discrimination of random words from code BAB_ group 2 by NCC matching of the SSI.

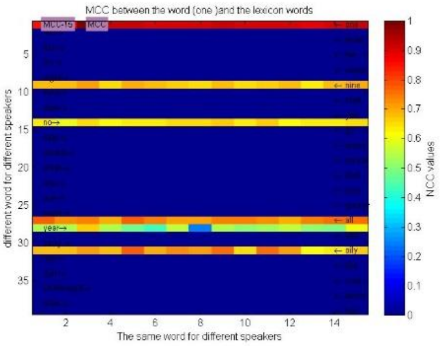
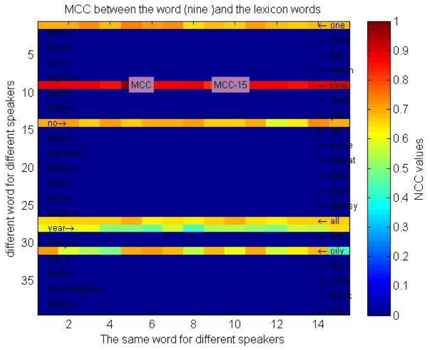
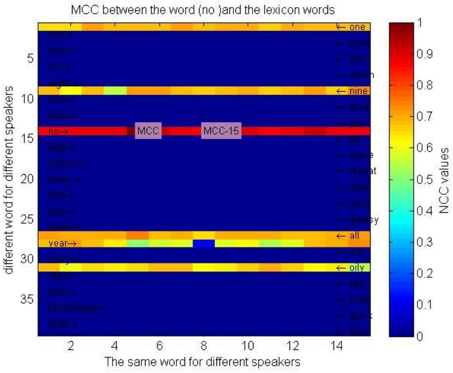
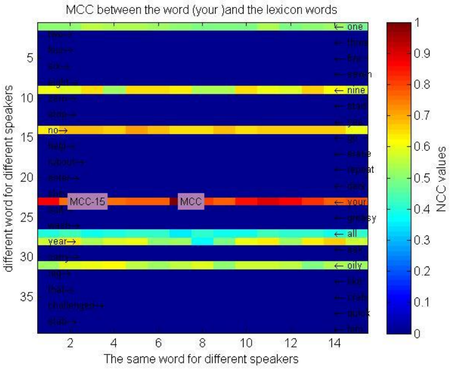
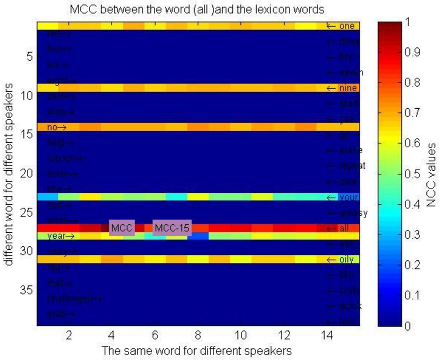
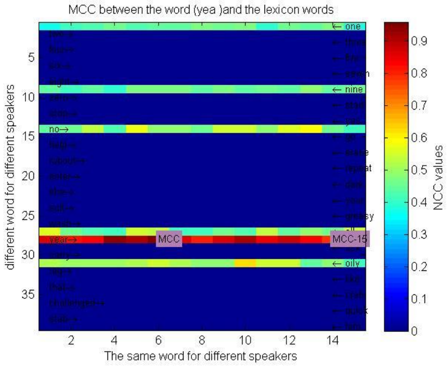
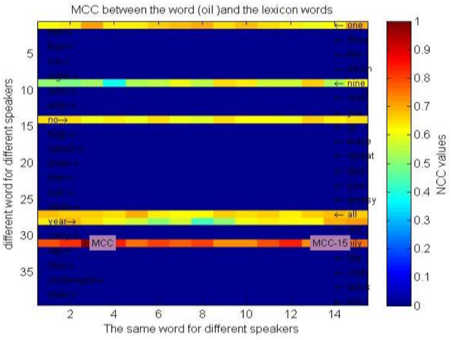
<table>
<tr><th colspan="2">Random words from code A (7 words)</th></tr>
<tr><td>Input word ‘one’</td><td>Input word ‘nine</td></tr>
<tr><td> </td><td> </td></tr>
<tr><td>Input word ‘no</td><td>Input word ‘your’</td></tr>
<tr><td> </td><td> </td></tr>
<tr><td>Input word ‘all’</td><td>Input word ‘year’</td></tr>
<tr><td> </td><td> </td></tr>
<tr><td colspan="2">Input word ‘oily’</td></tr>
<tr><td colspan="2"> </td></tr>
</table>

Figure 75 shows successful discrimination of random words from code **A** by NCC matching of the SSI.

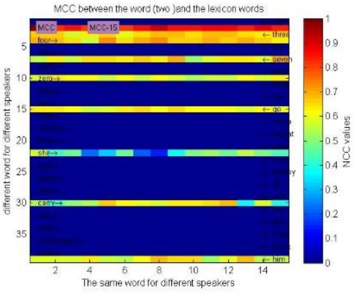
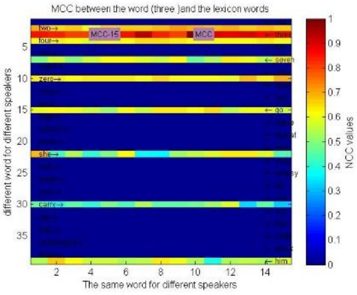
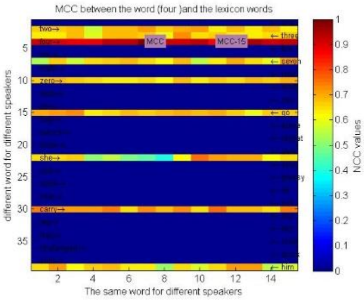
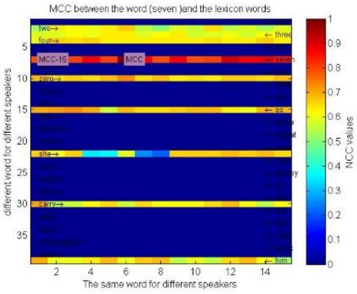
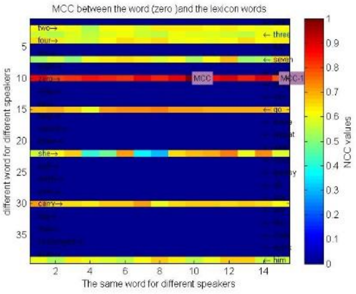
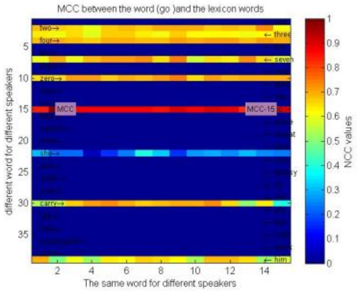
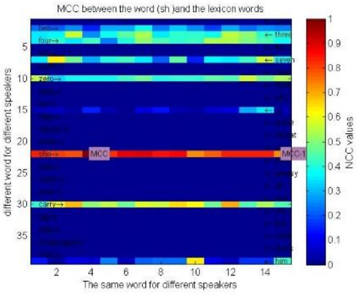
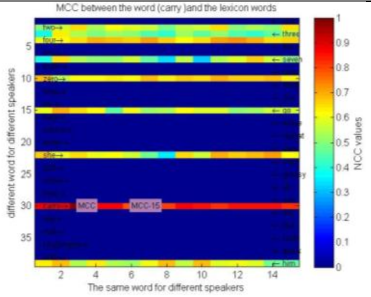
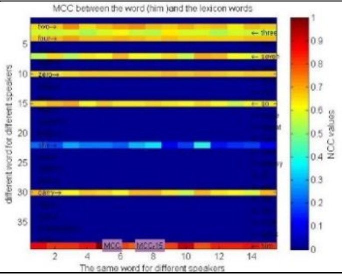
Random words from group of code **BA** (9 words)	
Input word ‘Two’	Input word ‘three
	
Input word ‘four’	Input word ‘seven’
	
Input word ‘zero’	Input word ‘go’
	
Input word ‘She’	Input word ‘carry’
	
Input word ‘him’	
	

Figure 76 shows successful discrimination of random words from code **BA** by NCC matching of the SSI.

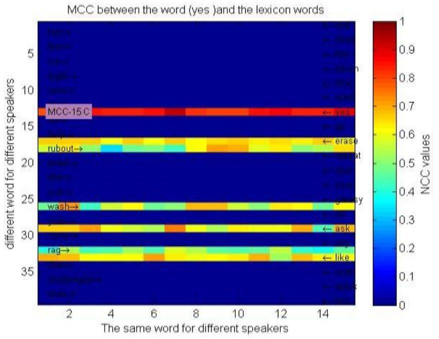
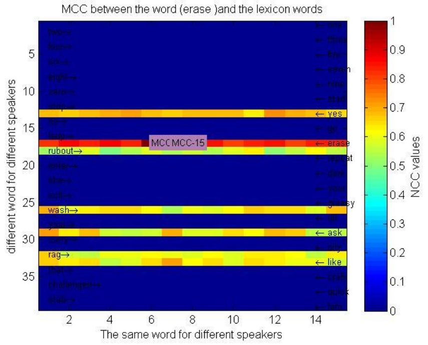
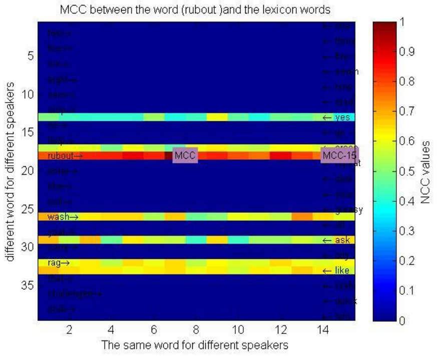
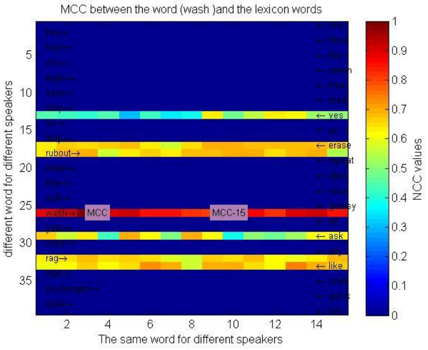
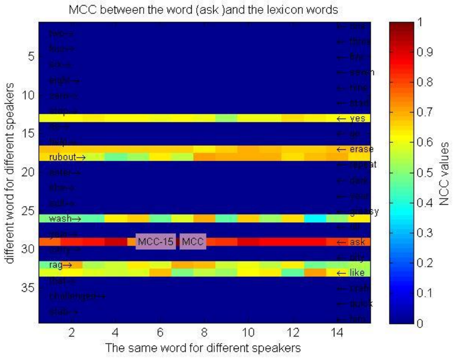
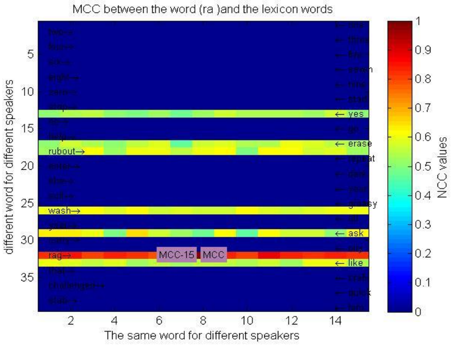
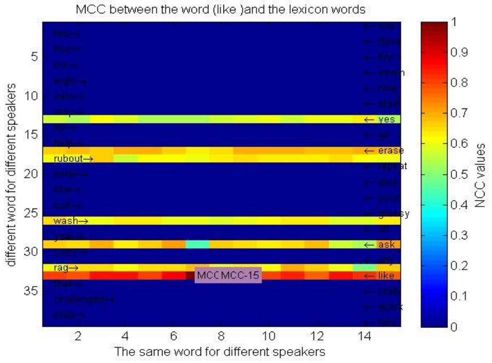
<table>
<tr><td colspan="2">Random words from group of code <b>AB</b> (7 words)</td></tr>
<tr><td>Input word ‘yes’</td><td>Input word ‘erase’</td></tr>
<tr><td> </td><td> </td></tr>
<tr><td>Input word ‘rubout’</td><td>Input word ‘wash’</td></tr>
<tr><td> </td><td> </td></tr>
<tr><td>Input word ‘ask’</td><td>Input word ‘rag’</td></tr>
<tr><td> </td><td> </td></tr>
<tr><td colspan="2">Input word ‘like’</td></tr>
<tr><td colspan="2"> </td></tr>
</table>

Figure 77 shows successful discrimination of random words from group of code AB by NCC matching of the SSI.

Discrimination of the word input has been made based on prior-information, which, is that the number of words which are the same is known i.e. 15 or 7 words. Consequently, we used the NCC_{15} or NCC_{07} to make the recognition decision if it is located in the same row of a word input in the lexicon which means the NCC matching of the SSI is able to distinguish the input word. The row colour is mostly red as shown in Figure 73 to Figure 77 which reflects the high NCC value. The average of NCC values of a word (values of the red coloured row) in their code sets (A, BA, AB, and BAB) is displayed in Figure 78 by the blue curve. The orange curve represents the NCC15 values, which are the minimum NCC values matching a word with the same word uttered by different persons. NCC_{16} is the maximum NCC value for matching of a word with other lexicon words, which are represented by the grey curve.

Practically, the recognition decision should be made by a global threshold, which can be used for A and B to make the recognition decision for all the lexicon words. The threshold should be less than the average of the NCC_{15} values at a distance far enough from NCC_{16} values to obtain high recognition reliability.

In any case, in our example test the NCC_{16} values do not exceed the NCC_{15} values. This means a global threshold can be defined. However, the procedure for NCC matching of the SSI that has been used in this example test does not show a big enough gap between the NCC_{15} and NCC_{16} values for high reliably recognition. On the other hand, this gap can be made bigger by optimising the parameters for the translation matching of SSIFSs and the thresholding parameters of the SSI separating it into the SSIFS and SSIRS patterns (i.e. Th-Ms). Figure 78 shows that for test No.4 the results indicate that the global threshold cannot be defined for whole lexicon word.

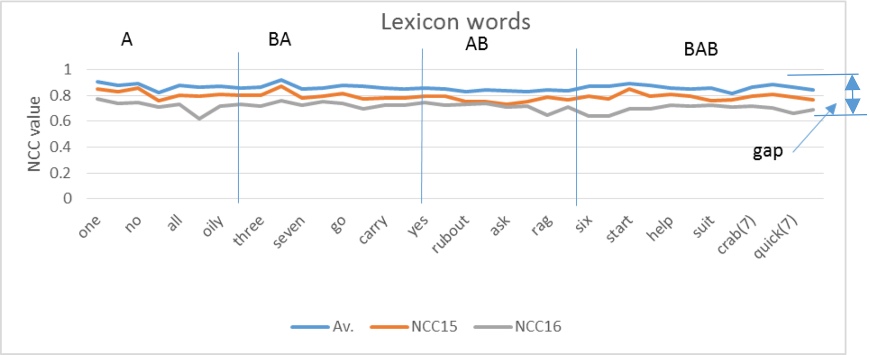


Figure 78 shows the threshold gap of the matching test based on the L1-1 category algorithm.

The local set codes are considered separately for defining the local threshold. Codes of sets A, BA, and BAB show that there is a possibility to define a threshold as shown in Figure 79 to Figure 81. However, the threshold is not sufficient for reliable application because the maximum NCC value can easily be changed slightly based on the starting point of the matching. In addition, for the code AB it is not possible to define a local threshold, as shown in Figure 82.

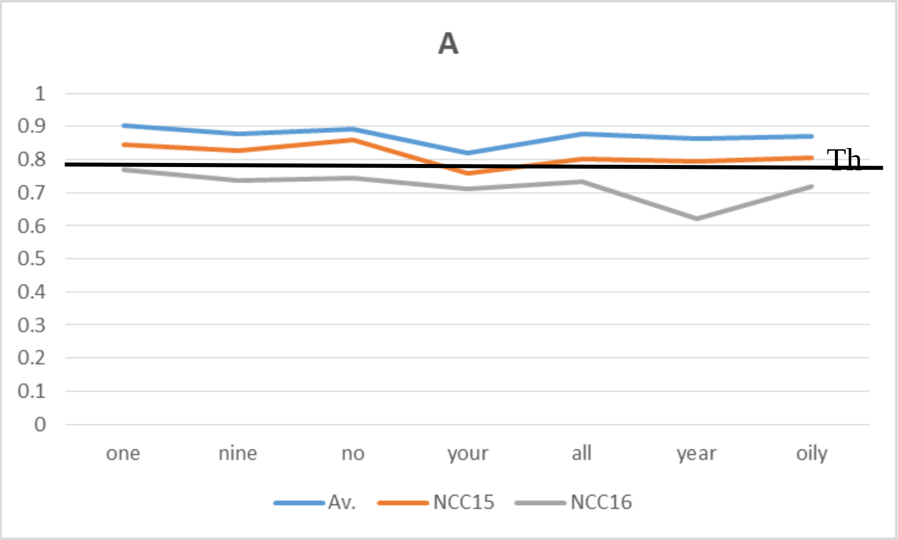


Figure 79 shows the local threshold for the group code A for the lexicon words of the study.

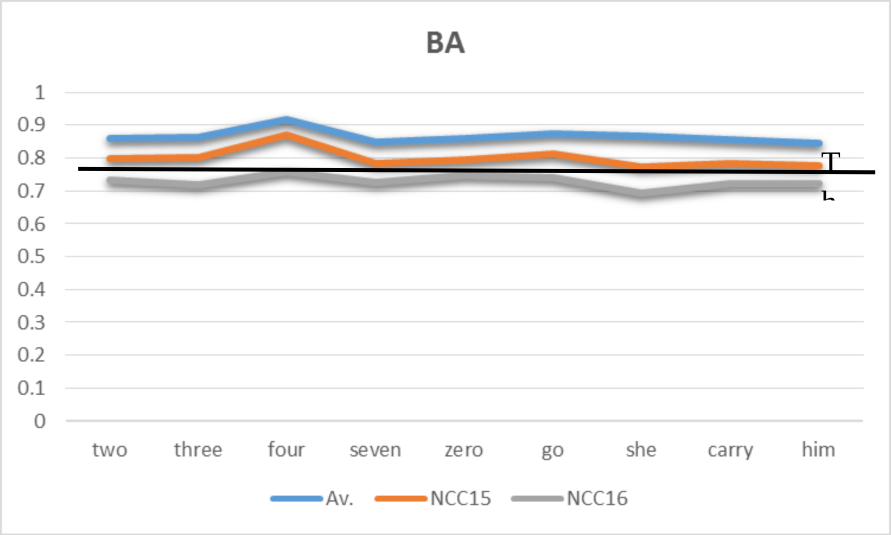


Figure 80 shows the local threshold for the group code BA for the lexicon words of the study.

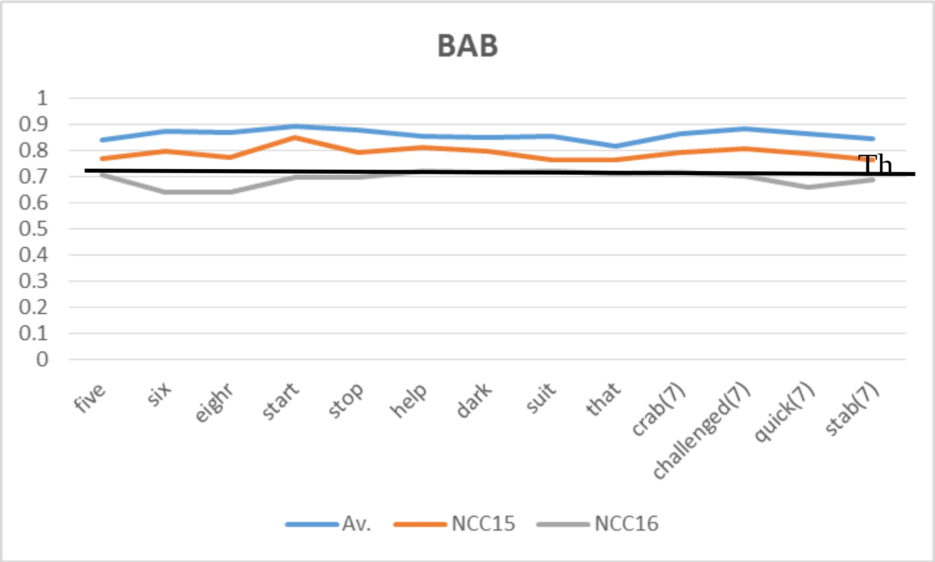


Figure 81.shows the local threshold for the group code BAB for the lexicon words of the study.

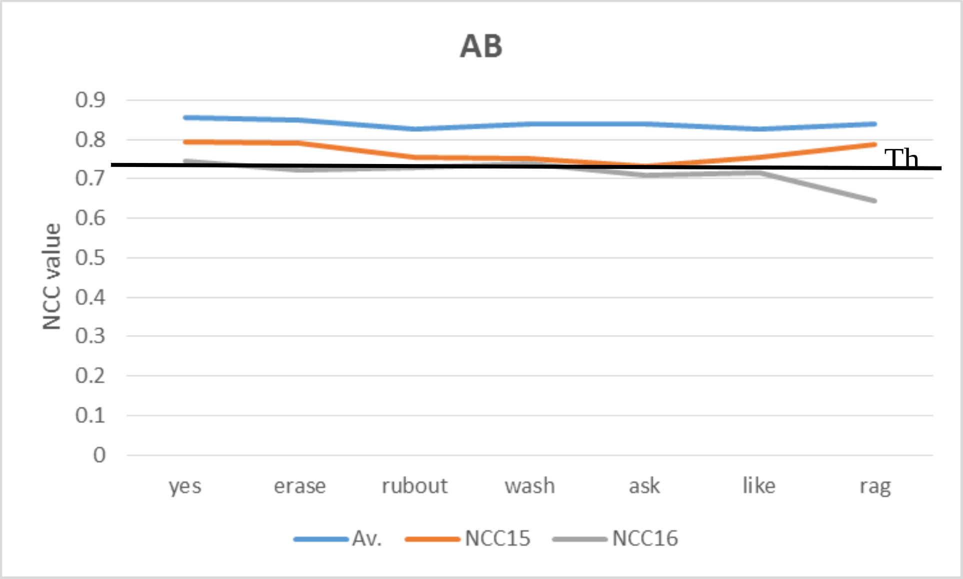


Figure 82 shows the local threshold for the group code AB for the lexicon words of the study.

The general algorithm based on the L1-1 category has been test only with NCC matching of the SSI, showing good results can be obtained. The expectation is that the MACH filter could perform better than NCC matching of the SSI. In addition, it is still possible to add a further enhancement to the performance of the NCC matching of the SSI.

4.14 Conclusion

The aim of this chapter is to show there is a promising approach to a word recognition by using the SSI patterns. Therefore, we tried to establish a general approach for SSI pattern recognition, which can be perfected with further work. On the other hand, we have used only the L1-1 category. Of course, the higher levels of SSI pattern classification can make the recognitions decision more reliable. The modularity of ASR is based on the recognition of training data sets by HMM or neural network methods. There has been an enormous amount of work during the last 85 years of research and the building of commercial applications. Therefore, integrating the SSI recognition with the current ASR is a necessary requirement. This is the aim of the next chapter.

CHAPTER FIVE

5 CHAPTER FIVE

Improving Hidden Markov model recognition, by integration with phoneme pattern recognition.

5.1 Introduction

Human speech production begins with an idea and through a series of neurological processes and muscular movements produces an acoustic sound pressure wave that is received by a listener's auditory system, processed and converted back to neurological signals.

To achieve this, a speaker forms an idea to convey, converts that idea into a linguistic structure by choosing appropriate words or phrases to represent that idea, utters the words or phrases based on learning grammatical rules associated with the particular language, and finally adds any additional local or global characteristics such as pitch intonation or stress to emphasize aspects important for overall meaning.

The availability of a small seed lexicon is assumed to represent a speech signal and then lean the pronunciations of new words directly from speech that is transcribed at word-level. The small seed lexicon are called 'pieces' of a speech signal which are matched to words in the recognition lexicon. The word-level is low-level recognition of speech and then there is the high level (language modelling) which is related to context dependency, and repetition.

The piece of the speech is called a phoneme (field phonemic) which allows for more pronunciations per word in the recognition lexicon. It is called phone (field phonetic) and is matched by training data to get better statistical models of acoustic variation in an objective way.

5.2 Converting speaker (articulatory phonetics) to listener (auditory phonetics)

As mentioned above, the theoretical unit of conveying linguistic meaning is a phoneme. Therefore, each phoneme has a unique set of articulatory gestures, which is a low level unit recognition of speech sound. Phonemes conveys the position and movement of the vocal tract articulators to produce a speech sound.

From the signal processing point of view, the frequency domain spectra derived from the acoustic waveform of speech can infer significant information about the speech signal. In addition, from a system modelling point of view, the articulators give the properties of the speech system filter. So each vocal tract shape (which produces a certain phoneme) can be characterised by a set of resonant frequencies, which are formants. The formants' frequency is a centre of frequencies. The type of excitation is one of the principal features of any speech sound. Thus, phonemes are practically measured by their frequency structure, the time waveform characteristic as acoustic sounds and then the unit is called a phone. Therefore, the motor program needed for performing a sequence of the phonemes produces a word or phrase.

For expression based recognising of speech (sentence level) a tonal pattern of pitch, syllable stresses and timing to a rhythmic speech pattern are used and are called prosodic features. The prosodic cues are intonation and stress. The prosodic features are said to be super-segmental which is extended more than the normal one phoneme segment [1]. The prosodic features are local and global characteristics which are important for the meaning of a sentence.

A speech sound production has another phenomenon which is called co-articulation, which is a change in the phoneme articulation and acoustics caused by the influence of another sound and requires movement of articulators in the glottal source and vocal tract. Technically, articulations of phonemes typically overlap each other in time, thereby causing sound patterns to be in transition most of the time.

The anticipatory co-articulation is categorised as right-to-left and left to right. The right-to-left is because a phoneme (the right) induces motion to the present phoneme and for the next phoneme (to the left). So, when an articulatory gesture does not influence the following phoneme, the given articulator may move toward a position more appropriate for the following phoneme. But when the articulator does conflict with the following phoneme, this will begin to move during phoneme production earlier than others in anticipation of the next phoneme. Thus, left-to-right occurs when some of the present phoneme features drift into the following phoneme. The anticipatory co-articulation suggests that syllables are the building blocks of words. Syllables occur when vowels have an optional initial and final margin with consonants. Syllables can influence the rhythm of a language, its prosody, its poetic meter and its stress patterns.

This suggests that the motor program needed for performing a sequence of sounds, syllables and words appears to anticipate the number of remaining speech units to be produced within a breath group, shortens those which simply needs to be produced, but retains the reachability of the consonant constrictions and the recognisability of the stressed syllables.

5.3 The gap between phonology and acoustic transcription

The aim of theoretical linguistics is to construct models that help us understand a language in some meaningful manner. Therefore, linguistic theory seeks to achieve explicit, falsifiable, predictive, and complete theories in mathematical models, including the formal descriptions of generative linguistics.

In models of phonology, the underlying form of the word is an abstract form that is postulated before any phonological rules have been applied to it. In, generative grammar, this is called the underlying representation [91]. A system of phonetic implementation relates surface phonological structures to measurable phonetic forms. These aspects of the linguistic system is studied theoretically in the dialects theoretical background, methods of measurement and phonological classification.

It is hypothesized, that the acoustic speech models are enough to quantify a wider range to cover speaking styles. Tactically, an acoustical speech model unit is an abstract (Acoustic feature) of a phonemic unit. Therefore, there is a gap between actual speech and acoustic speech representation. Actually, the gap is an error accumulation of speech representation unit, acoustical models and recognition algorithms. Speech engineers have been working to enhance speech recognition by working on better feature extractions, pronunciation modelling, acoustic modelling and noise handling [92].

5.4 Automatic speech Recognition (ASR)

5.4.1 Fundamentals of speech processing

The speech recognition models are hierarchically decomposed into different levels: the acoustic, phonetic and linguistic. With this decomposition, models of entire utterances can be classified into sharing sub-models (e.g. sentence into word models, and word models into sub-word models). There are a variety of types of sub-word model that are used including phoneme, bi-phone, syllable, or demi-syllable models.

The problem of how fundamental units (phones, syllables and, words) may be concatenated, in what order, in what context and with intended meaning, is more involved than simply programming the correct grammatical rules for the language. Therefore, the problem is approached with two broad categories: acoustic decoder (low level) then followed by linguistic decoder (high level), with the indicated direction of the flow of information given in Figure 83.

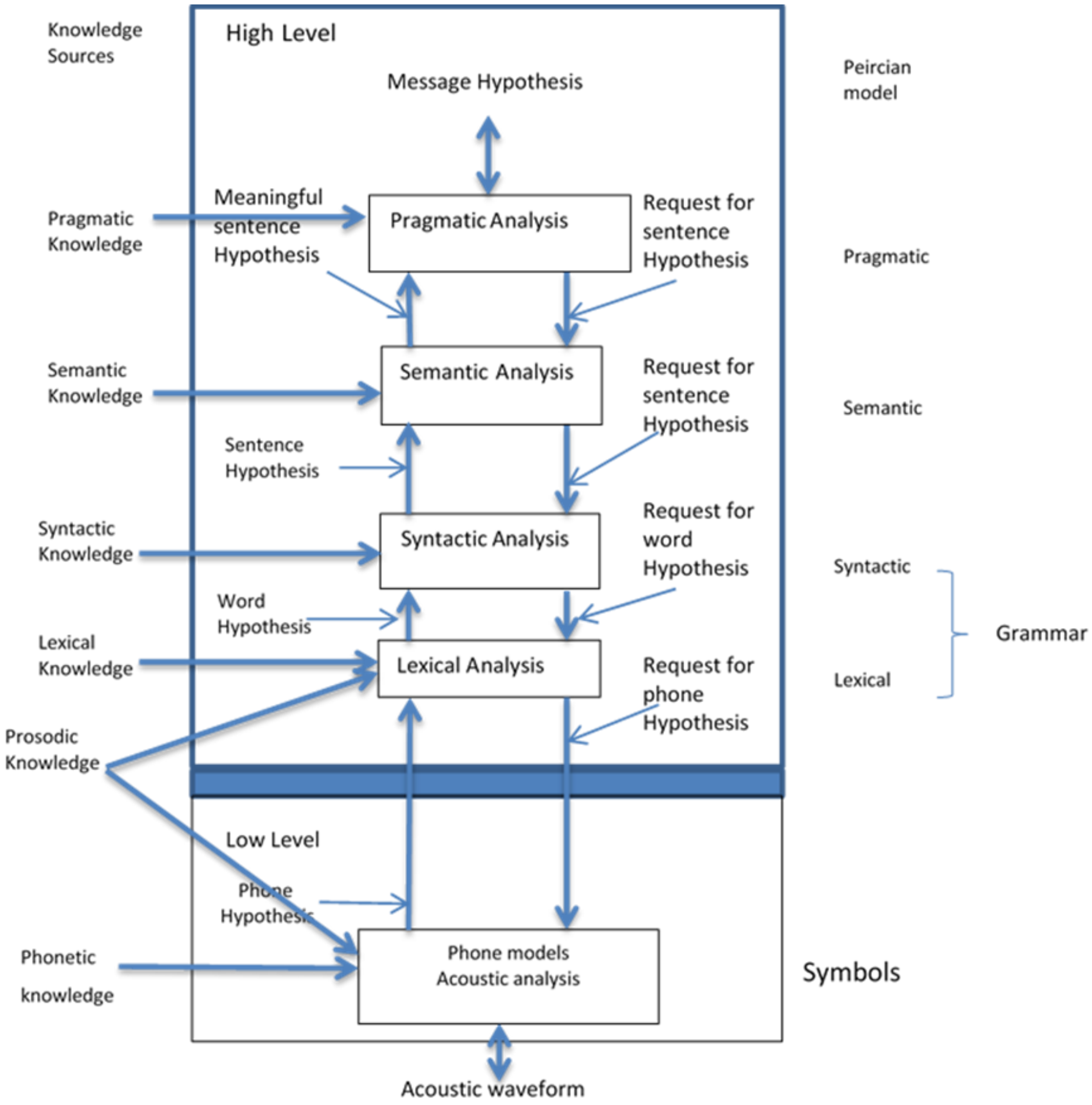


Figure 83 A general speech recogniser.

5.4.2 Speech processing engineering

From the speech processing engineering prospective, speech recognition is a special case of pattern recognition which is supervised by training and testing. The speech recognition types are isolated words, connected words, continuous speech, spontaneous speech and speaker recognition. An ASR with spontaneous speech ability is able to handle a variety of natural speech features such as the words running together. The ASR may be viewed as working in four stages: analysis (pre-processing), feature extraction, modelling and, testing (post-processing) as shown in Figure 84. The main difference between classes of speech recognition is the modelling stage.

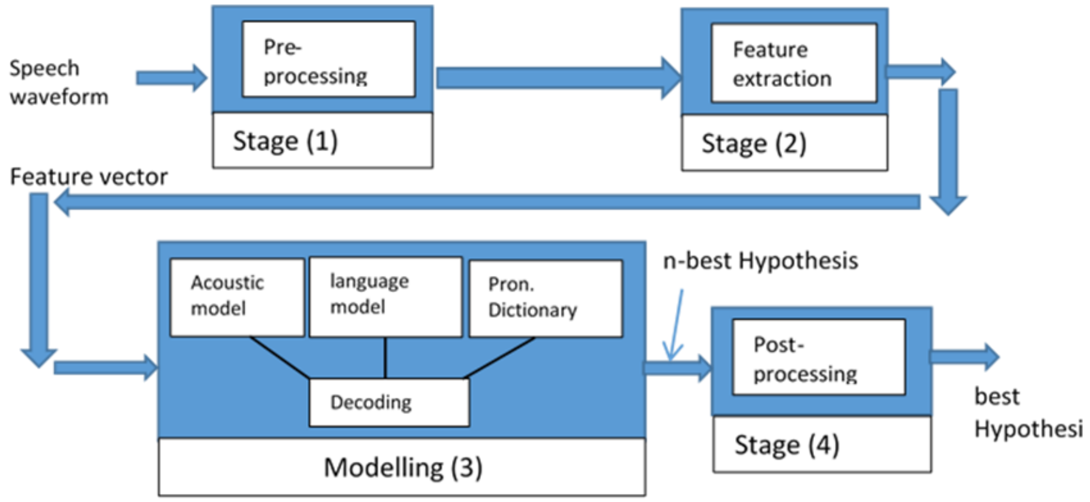


Figure 84 ASR system.

The training and test phases employ powerful statistical signal processing approaches. There are two basic classes of statistical signal processing methods which are: template matching (i.e. dynamic time warping) and a stochastic approach (i.e. hidden Markov models and artificial neural networks). These algorithms form the basis upon which almost all contemporary speech recognition algorithms using sequential computation rely. The HMM has been the basis of several successful commercial speech recognition systems, because it is amenable to computation on conventional sequential computing machines [1]. The HMM is a stochastic finite state automaton (machine) used to model a speech utterance. The utterance may be a word, a sub-word unit, or in principle, a complete sentence or paragraph.

From this point of view, there is no real loss of generality when using HMMs (which tends to model words). We selected words from continuous speech data to process them in HMM. These isolate words are considered as models of sub-word units (new units), models of words or models of phrases. Therefore, we are going to show the modelling by studying isolated word speech recognition (IWSR) which is fundamental to speech recognition systems [1]. The IWSR is based only on acoustic modelling, where there is no syntax or semantics to constrain the choice of words. The four stages of IWSR are shown in Figure 85 these are summarised below:

Analysis stage: at this stage, the speech is applied to a low pass filter and segmented by using the frame size and shifted to be in the range of 10-30 msec.

Feature extraction: Acoustic observations are extracted over time frames of uniform length, then a Mel-scale filter is applied to get for example ten coefficients chosen to be an acoustic feature vector. Those features are normalized in energy and are called Mel Frequency Cepstral Coefficient (MFCC).

Modelling: is a process of establishing statistical representations for the acoustic feature vector as states of the HMM. Researchers have proposed a variety of modifications and extensions for HMM based acoustic models to overcome their limitations. The HMM is a formal foundation for making probabilistic models of a liner sequence. The recognition processing can be done by recasting of the similarity between a recognition model and a segment of a speech signal as a probability.

Acoustic Model: The phonetic models describe the statistical structure of words. Usually, the decision trees are used to produce word pronunciation. The acoustic models describe the structure of sound for each in a probability distribution over a varying length sequence of feature vectors. Acoustic models typically used to represent the distribution of each feature vector include parametric mixture densities.

Testing stage: in general, it is a maximum of the likelihoods which are computed by the recognition model.

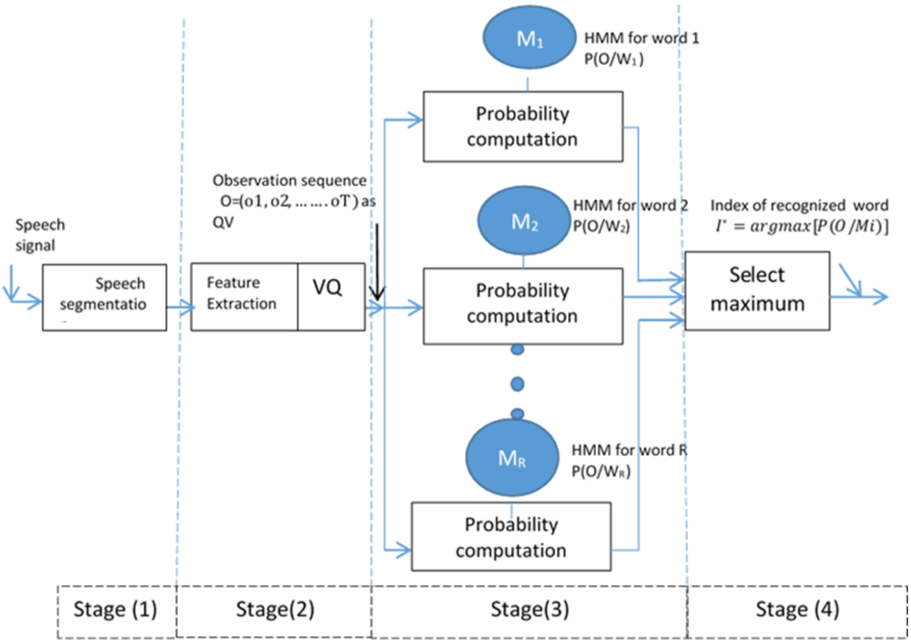


Figure 85 Isolated word speech recognition system.

The recognition of a word in the ISWR algorithm is accomplished in two steps: first is estimation of the observations of sequential features of a word speech waveform $[o1, o2, \dots oT]$. Observations of a word can be supervised (in training) to create discrete observations of state transitions; the second step is recognition of the observation by using HMM. Actually, the word we want to be recognised (observation of the same word as that trained) should give the same state transitions. For running matching in a consistent way and while including as much as possible the difference of word features during various utterances, vector quantisation (VQ) is used to compress word observations to a fixed length vector. Associated with VQ is a distortion penalty but there is an incentive to keep the codebook as small as possible without endangering recognition ability since a larger codebook implies more computation.

5.5 Probabilistic Model and the training problem

It is well known that using probability as a prediction (i.e. propositions) is known as Bayesian probability, but the question is: it is possible to train a process probability to get better results and how can that be done.

5.5.1 What does training process probability mean and how can a given class generate certain feature vectors

Suppose we have a set of words indexed by integers, $w = 1,2, \dots .R$ which are outcomes of the word random variable $\underline{w}$,which could be represented as a sequence of HMMs. The number of HMM states might equal the average number of phonemes within a word (e.g. 5 states). On the other hand, we have a feature vector modelled by the random vector $\underline{O}$ extracted from the utterance of the word to be recognized. Ideally, the given vector outcome $\underline{O} = O$.

The acoustic observations $(O; o1, o2, o3, \dots oT)$ are given, so the optimal word sequence $(\widehat{W})$ can be obtained from a conventional ASR engine as the probability below:

$$\widehat{W} = arg\,\underbrace{max}_{w}\, P\big([\underline{w} = w]|O\big) \qquad \text{Equation 15}$$

where, w is a class represent a sequence of possible phones or words in ASR.

However, in the learning process a given class will generate a certain feature vector, rather than the converse:

$$\widehat{W} = arg\,\underbrace{max}_{w}\, P\big(O|[\underline{w} = w]\big) \qquad \text{Equation 16}$$

By the definition of Bayes' rule we have:

$$\underbrace{P(W|O)}_{\substack{posterior\ prob.of\\ class[target]\ (W)\\ given\ predictor\\ [attribute]\ (O)}} = \frac{\overbrace{P(O|W)}^{\substack{likelihood\\ of\ predictor\\ a\ given\ class}}\ \overbrace{P(W)}^{\substack{class\\ prior\\ prob.}}}{\underbrace{P(O)}_{\substack{predictor\\ prior\\ prob.\\ [evidence]}}} \qquad \text{Equation 17}$$

and

$$P(W|O)\,P(O) = \; P(O|W)P(W)$$ Equation 18

However, the observation is given, so *P(O)* does not depend on probability P(W), therefore:

$$P([\underline{w} = w]|O) \; = \; P(O|[\underline{w} = w])P([\underline{w} = w])$$ Equation 19

Usually, the word probabilities are equal, so

$$P([\underline{w} = w]) = \frac{1}{R}\,, \quad w = 1,2, \dots, R,$$ Equation 20

Thus the maximization does not depend on P(w):

$$\widehat{W} = arg\,\underbrace{max}_{w}\,P([\underline{w} = w]|O) = \; arg\,\underbrace{max}_{w}\,P(O|[\underline{w} = w])$$ Equation 21

Then

$$\widehat{W} = arg\,\underbrace{max}_{w}\,P(O|[\underline{w} = w])$$ Equation 22

Equation 22 is the training process conditional probability.

5.5.2 The probability estimation of speech recognition

The probability estimation process of ASR is a two part process: acoustic modelling, followed by language modelling [93]:

$$P(W|O) = \quad \underbrace{\frac{\rho(O|W)}{\rho(O)}}_{\substack{\textbf{Acoustic modelling}, \\ \text{dependent probability}}} \quad \underbrace{P(W)}_{\substack{\textbf{language modelling}, \\ \text{the prior probabilities of} \\ \text{sentence models}}}$$ Equation 23

where ρ represents a probability density and *P* represents a probability [93].

Usually, acoustic modelling and language modelling are traded independently, and ρ(O) is assumed to be equal across the model [93].

5.6 Training the HMM

This section summarizes the HMM concept, implements the HMM for recognition, and shows how the HMM training improves speech recognition. The HMM is based on a random process called a Markov chain process, which describes a transition from one state to another in a state space.

5.6.1 Markov chain

In stochastic processes, the conditional independence probability is different from the unconditional independence probability (marginal probability). That is the conditional independence can imply some factorization of an existing joint distribution. That can yield huge computational savings (e.g. forward and backward algorithms) [94]. A stochastic process can be combined with a Markov chain t =1, 2,.. and the transition probabilities between chains are independent of time, so the Markov chain is said to be homogeneous or stationary.

Therefore, the joint probability of random variables can be calculated from a different set of random variables (i.e. factorisation of joint probability) by using only a chain rule. A graphical model is used to represent the relation between a set conditional independence factorisation of random variables which is called a Bayesian network.

Let us have a set of variables (s_1, s_2,…..,s_n) that can come in order as shown which is called the first order of the Markov chain.

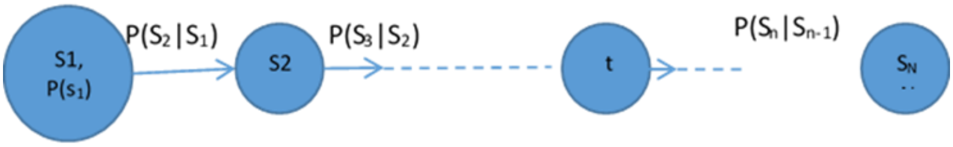


Figure 86 State diagram of first order Markov chain.

One of the important properties of the Markov process is called the Markov property which states that the future is conditionally independent on the past given the present [95]:

$$P(S_{t+1}|S_t, S_{t-1}, \dots, S_1) = P(S_{t+1}|S_t) \qquad \text{Equation 24}$$

Therefore, the joint probability of the first order Markov is the memoryless property of a stochastic process:

$$P(s_1, s_2, \dots, s_n) = P(s_1) \cdot P(s_2|s_1) \cdot P(s_3|s_2) \dots P(s_n|s_{n-1}) \qquad \text{Equation 25}$$

Actually, the joint probability can be factored in different ways of conditional probability factorisation to give a different order of the Markov chain. The probability of a second order Markov chain can thus be

$$\begin{aligned} P(s_1, s_2, \dots, s_n) & \\ &= P(s_1) \cdot P(s_2|s_1) P\left(s_3|s_{1,}s_2\right) \\ &\cdot P(s_3|s_2) P(s_4|s_{2,}s_3) \dots P(s_n|s_{n-1}) P(s_n|s_{n-1}, s_{n-2}) \end{aligned} \qquad \text{Equation 26}$$

Equation 26 can be represented as a state diagram as shown in Figure 87

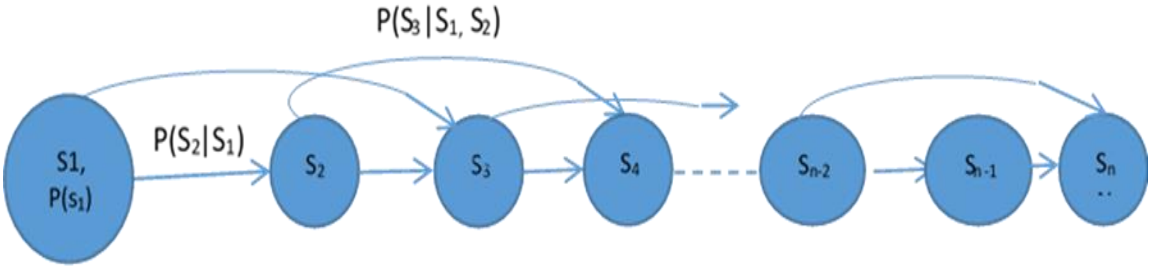


Figure 87 the second order Markov chain.

5.6.2 Elements of HMM

The Markov property is a basis of elementary HMM. The HMM is suitable for working with sequential data of temporal pattern (e.g. speech and music). A hidden Markov process is a stochastic process of two stages: an underlying process of states. The hidden state space is assumed to consist of one of N possible values, $s_1, \dots., s_T$ which is hidden from observation $o1, \dots. oT$ and the observation process (stochastic process) which is determined by the underlying process.

The HMM is the simplest dynamic Bayesian network[13]. Figure 88 shows the elements of a HMM. It should be apparent that the simple HMM that corresponds to the “hidden states” and the “observation process” is one in which each state corresponds to a specific state, and for which an observation probability is defined for each state. The choice of states is dictated by the state transition matrix of the HMM. The HMM is a powerful statistical tool for modelling generative sequences that can be characterised by an underlying process generating an observable sequence. In other words, to recognise the behaviour of a random variable (O) among a model (S), which is a set of random variables, a huge number of observations of that random variable are required to provide enough accuracy. The observations can be represented as the probability $P(O|S)$ of a track of a few of the states. Then, any new random variable observations (which require to be recognised) will give almost the same track states with is a high likelihood (max. of $P(o|s)$) of that random variable among a model of random variables.

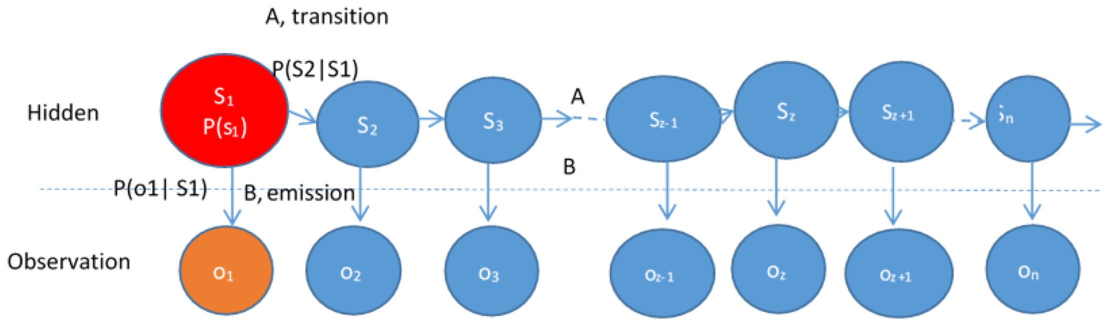


Figure 88 Elements of HMM.

[13] Bayesian network is a statistical model. A probabilistic graphical model represents that a set of random variables and their conditional dependencies via a directed acyclic graph (DAG).

The joint probability represented in Figure 88 can be factored into a conditional probability;

$$P\left(o_1, \dots o_T, s_{1,} \dots s_T\right) = P(O_1^T, S_1^T) \\ = P(s_1)P(o_1|s_1)\prod_{z=2}^{T} P(s_z|s_{z-1})\,P(o_z|s_z) \qquad \text{Equation 27}$$

Equation 27 can be rewritten as probabilities:

$$P\left(o_1, \dots o_T, s_{1,} \dots s_T\right) \\ = P(o_1|s_1)P(o_2|s_2)..P(o_T|s_{T-1}) \\ \times P(s_1)\;P(s_2|s_1)P(s_3|s_2)\dots P(s_T|s_{T-1}) \qquad \text{Equation 28}$$

For convenience we write the probability of the HMM:

$$M = (\pi, A, B) \qquad \text{Equation 29}$$

These probabilities can be classified as parameters as follows:

Transition probability (**A**) **:** $P(s_{z+1} = j|s_z = i) = a(j|i)$, $i, j \in (1, \dots m)$. The transition probability is a square matrix of size $\times S$, where, S is the number of states in the HMM.

For each of the n possible states, there is a set of emission probabilities governing the distribution of the observed variable at a particular time given the state of the hidden variable at that time. Usually, $(O, of\ K\ observation)$ is a probability density function (PDF). In this case, the emission is a PFD as well.

Emission probability($\boldsymbol{B}$): $(o_T = i|s_z = i) = b(o|i)$, $i \in (1, \dots m)$, The Emission probability is a matrix of size K×S .

Initial probability$(\boldsymbol{\pi})$**:**$(s1)$: how to choose initial estimates of the HMM parameters to maximize the likelihood function is not simple or straightforward to answer. By experience, initial estimates can be obtained in many ways; maximum likelihood segmentation of observations with averaging, segmentation of the observation sequences into states with averaging of observations within states; and segmental k-means segmentation with clustering [6]. $\pi(t) = [p(o_1 = 1), \dots, p(o_1 = S)]^T$, where xt is the state random process.

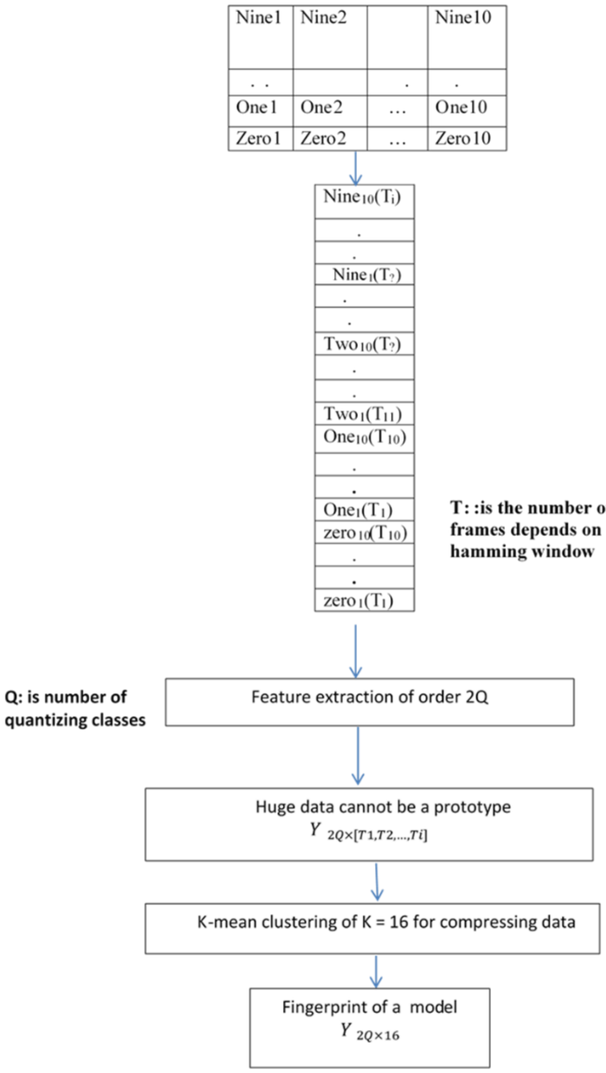
We can also add the number of states as a control parameter to a HMM model; (S) is number of hidden states of the HMM. In implementing HMMs. There are two schools of thought as to the number of states to use in each word model. One idea is to let the number of states correspond roughly to the number of phonemes within the word; hence appropriate models would have from 2 to 10 states. The other idea is to let the number of states correspond roughly to the average number of observations in a spoken version of the word, which is called the Bakis model [6, 96].

Actually, in choosing types of HMM parameters, the model (e.g. ergodic or left to right), type of observation symbols (discrete or continuous, signal or multi-mixture), and the number of states are made depending on the signal being modelled. There is no simple way making such choices [6].

5.7 Creating group observations of certain vocabulary systems

The voice signal of a word is a sequence temporal patterns (frames). There are many utterances possible of a word that produce different acoustic models. The patterns can be represented as a vector of probability density functions, which have different waveform length. Therefore, the feature vectors of waveforms of a word (codebook representation) have different lengths as well. To unite the length and to represent data in one vector, the K-mean algorithm (Vector quantization) is usually used. The vector quantization is originally a set of data symbols compressed to create the vector distribution. It can be called a prototype vector of a word. The prototype vector is a fingerprint word model and it can be used to sort out any probability density function of the word to create a word observation.

As an example, steps in creating a fingerprint model of words of ten digits (0 to 9) are shown in Figure 89.



Nine1
Nine2
Nine10
. .
.
.
One1
One2
...
One10
Zero1
Zero2
...
Zero10
$Nine_{10}(T_i)$
.
.
$Nine_1(T_?)$
.
.
$Two_{10}(T_?)$
.
.
$Two_1(T_{11})$
$One_{10}(T_{10})$
.
.
$One_1(T_1)$
$zero_{10}(T_{10})$
.
.
$zero_1(T_1)$
T: :is the number of frames depends on hamming window
Q: is number of quantizing classes
Feature extraction of order 2Q
Huge data cannot be a prototype
$Y_{2Q\times[T1,T2,\ldots,Ti]}$
K-mean clustering of K = 16 for compressing data
Fingerprint of a model
$Y_{2Q\times16}$

Figure 89 computation of fingerprint of aquatics model of digits (0:9).

An observation vector of any word of the ten digit model can be made by comparing each frame of features of a word with the fingerprint matrix of the vocabulary system. The observation is the position (column number) of the minimum Euclidean distance between the fingerprint matrix and vector frames of the word features which are matched one by one, with the fingerprint matrix placed as a column member in the observation vector. By repeating the same procedure for all model words, the group of observations of the ten digit vocabulary system can be collected as shown in Figure 90.

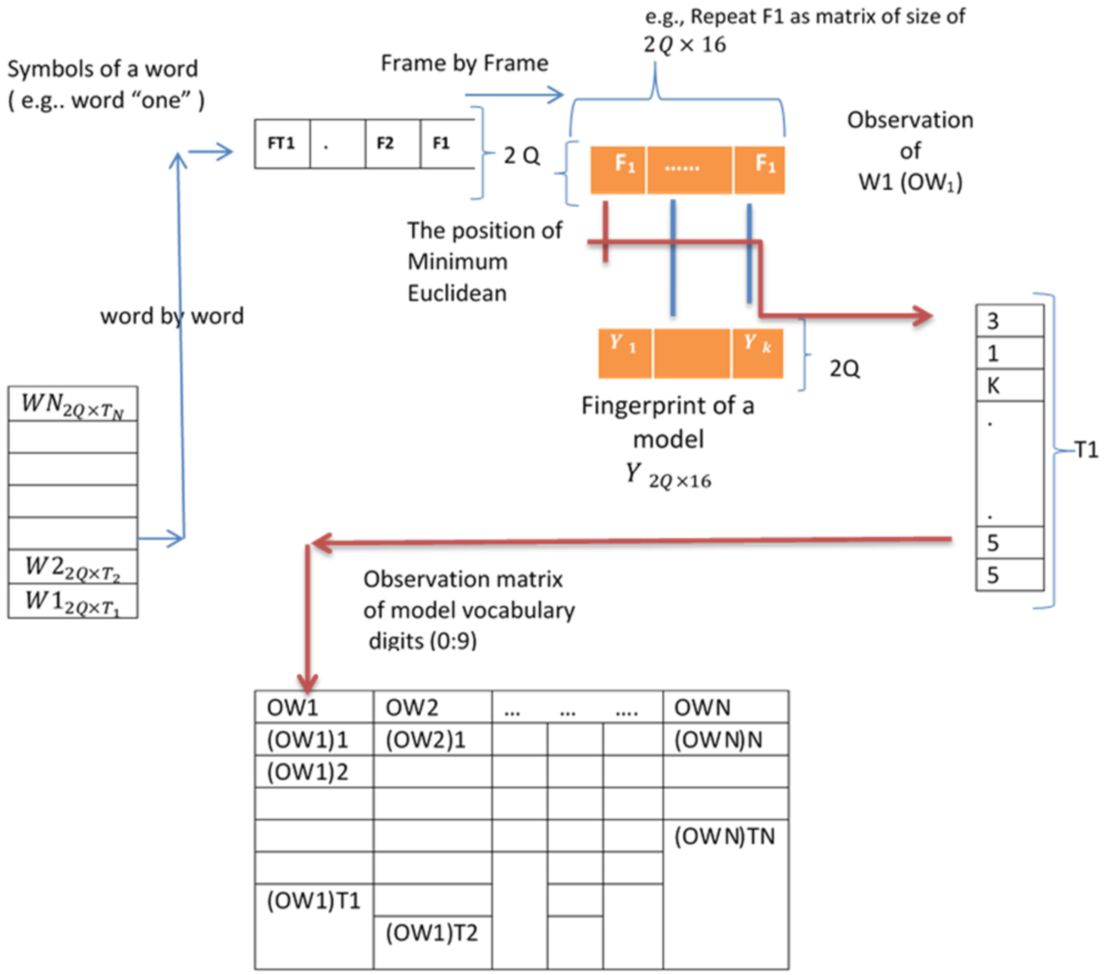


Figure 90 Computation of observation matrix of word digits (0:9).

5.8 HMM designed for isolated word recognition

For isolated word recognition with a distinct HMM designed for each word in the vocabulary, a left-right model HMM should be used that it is more appropriate than an ergodic model because the observation sequences (underlying the state sequence) associated with the model has the property that as time increases the state index increases [6]. We can then associate time with model states in a fairly straightforward manner.

First, let us define the elements of a discrete observation HMM;

$$\underbrace{M}_{A\ model} = (\underbrace{S}_{\substack{The\ number\\ of\ states\\ in\ the\ model}} , \underbrace{\pi(t)}_{\substack{The\ initial\\ state\\ distribution\\ probability\\ vector\ at\\ time\ t}} , \underbrace{A}_{\substack{Transition\\ matrix\\ of\ size\\ S\times S}} , \underbrace{B}_{\substack{Obsevation\\ matrix\\ of\ size\\ K\times S}} ,)) \qquad \text{Equation 30}$$

The HMM requires specification of the two model parameters, S and K (S is number of states, K is k-means clustering number of clusters) and the other variable represent three probability measures A, B, π (1).

Thus, the training problem is to adjust the model parameters to maximise the likelihood $\mathrm{\rho(O/W)}$, where this is the probability of the observation sequence $\mathrm{O_1^T} = \big(\,\mathrm{o(1), o(2), \ldots\ldots..\, o(T)}\big)$, given the model $\mathrm{W_1^T} = \big(\mathrm{w(1), w(2), \ldots\ldots w(T)}\big)$, which is fixed a state sequence.

The $\mathrm{\rho(O/W)}$ can be computed in terms of local joint densities at particular state $\mathrm{w(t)}$, and a word is represented by a particular state sequence; Consider a specific state sequence through the HMM of proper length T, say $\mathrm{I_1^T} = (\,\mathrm{i(1), i(2), \ldots\ldots..\, i(T)})$. The probability of the observation sequence being produced over this state sequence is

$$P(o|i,w) = b(o_1|i_1)b(o_2|i_2)\ldots b(o_T|i_T) \qquad \text{Equation 31}$$

The probability of the state sequence i is:

$$P(i|w) = \ P(o_1 = i_1)\ a(i_2|i_1) \ldots a(i_T|i_{T-1}) \qquad \text{Equation 32}$$

Therefore:

$$P(o,i|w) = b(o_1|i_1)b(o_2|i_2)\dots b(o_T|i_T) \times P(o_1 = i_1)\; a(i_2|i_1) \dots a(i_T|i_{T-1}) \qquad \text{Equation 33}$$

The $\mathrm{P(o,i|w)}$ represents any path in the HMM. In order to find the $\mathrm{P(o|w)}$, we must sum this result over all possible paths, because it represents mutually exclusive events:

$$P(O|W) = \sum_{all} P(o,i|w) \qquad \text{Equation 34}$$

The HHM is built in levels, if we assume that the set of V word HMMs as $\mathrm{w^v}, 1 \le \mathrm{v} \le \mathrm{V}$, then to find the optimum sequence of HHMs that matches we maximize the likelihood by using the Viterbi algorithm. Therefore, it is required to do a Viterbi match against O for each HMMw^{v} and at each level ℓ. So, we start at frame 1 of the observation interval on level 1, then retain each possible frame t [1].

The Equation 33 is a naive approach (direct computation) because it consumes a big number of computations (requires the calculation of $\mathrm{2TS^T}$,which for S=5, and T = 100, needs $2 \times 100 \times\ 5^{100} \approx 1.6 \times 10^{72}$ multiplies) [1].

To avoid this numerical problem, in practice, the computations of $\mathrm{\rho(O/W)}$ are divided into two recursions stages to reduce the computations, because the recursion is a reuse of earlier computations. In this case, the observation sequences ($\mathrm{O_1^T} = (\,\mathrm{o(1), o(2), \dots\dots.. o(T)}$)) are divided in two partial sequences ($\mathrm{O_1^t, O_{t=1}^T}$). The first recursion is the joint probability of having the forward partial sequence of $\mathrm{O_1^t}$ having arrived at state i at the tth step, and the second is the backward partial sequence $\mathrm{O_{t+1}^T}$ which given the state sequence emerges from i at time t. This is called the forward-backward algorithm (F-B algorithm) as shown in Figure 91.

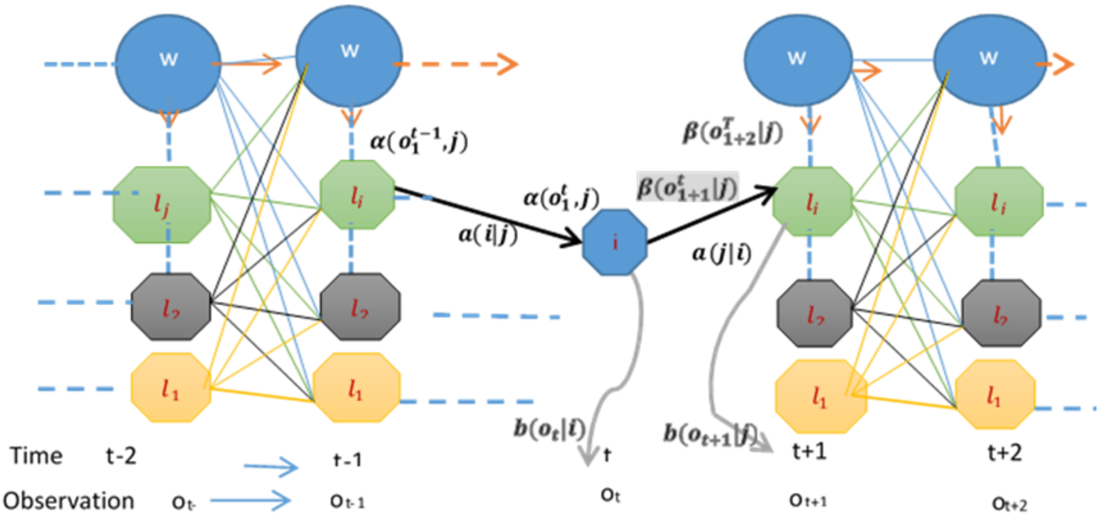


Figure 91 Bayesian network of forward -backward algorithm.

The forward algorithm for the HMM is the joint probability in a supposed state (i) which has the forward partial sequence of $\mathrm{O}_1^{\mathrm{t}}$, and for which the emission and transition probabilities are known, as illustrated in Figure 92 and described by Equation 35.

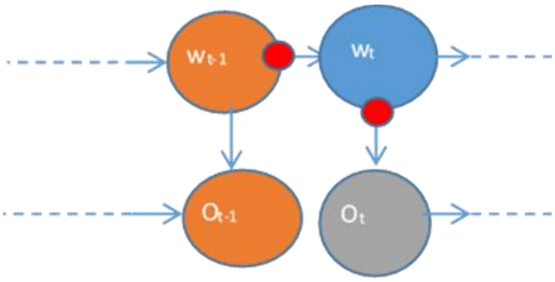


Figure 93 Bayesian network of forward algorithm.

$$\begin{aligned} P(W_t, O_t) &= P(w_1, \dots w_t, o_1, \dots o_t) \\ &= P(w_1, w_2, o_1, o_2) + P(w_3, w_4, o_3, o_4) + \cdots \\ &\quad + P(w_{t-1}, w_t, o_{t-1}, o_t) \\ &= \sum_{k=2}^{t} P(w_{k-1}, w_k, o_{k-1}, o_k) \end{aligned} \qquad \text{Equation 35}$$

This joint probability $\mathrm{P}(\mathrm{w}_{\mathrm{k}-1}, \mathrm{w}_{\mathrm{k}}, \mathrm{o}_{\mathrm{k}-1}, \mathrm{o}_{\mathrm{k}})$ can be factored as a combination of marginal probabilities and a joint probability:

$$P(w_t, o_t) = \sum_{k} P(o_t|w_t, w_{t-1}, o_{t-1}) P(w_t|w_{t-1}, o_{t-1}) P(w_{t-1}, o_{t-1})$$ Equation 36

But o_t is independent of w_{t-1}, o_{t-1} , and w_t is independent of o_{t-1} , therefore:

$$P(w_t, o_t) = \sum_{k,1..n} \underbrace{P(o_t|w_t)}_{\substack{Emission\ Prob.\\(known)}} \underbrace{P(w_t|w_{t-1})}_{\substack{Transition\ Prob.\\(known)}} P(w_{t-1}, o_{t-1})$$ Equation 37

As, we know both the emission and transition probabilities already, therefore the Equation 37 can be rewritten as:

$$\alpha(w_t) = P(w_t, o_t) = \sum_{k} b(t)a(t)\alpha(w_{t-1})$$ Equation 38

where α is forward HMM, and so we need to know the initial forward probability:

$$\alpha_1 = P(w_1, o_1) = P(w_1)P(o_1|w_1) = \pi(1)b(1)$$ Equation 39

Therefore, now we have all information to calculate α.

For the backward algorithm for the HMM the goal of is to compute $P(o_{t+1}, o_{t+2}, \dots, o_T|w_t)$ as illustrated in Figure 94 and described by Equation 40

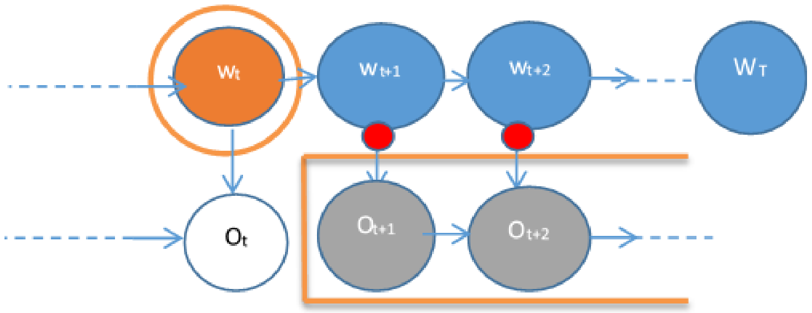


Figure 94 Bayesian network of backward algorithm.

$$\begin{aligned} P(o_{t+1}, o_{t+2}, \dots, o_T | w_t) &= P(o_{t+1}, w_{t+1} | w_t) + P(o_{t+2}, w_{t+2} | w_{t+1}) \\ &\quad + P(o_{t+3}, w_{t+3} | w_{t+2}) + \cdots \\ &= \sum_{k=t+1}^{T} P(o_{t+1}, w_{t+1} | w_t) \end{aligned}$$

Equation 40

$$\begin{aligned} P(o_{t+1}, o_{t+2}, \dots, o_T | w_t) = \sum_{k=t}^{T} P(o_{t+1}, w_{t+1} | w_t) &= P(o_{t+2} | w_{t+1}, w_t, o_{t+1}) \\ &\quad + P(o_{t+3} | w_{t+2}, w_{t+1}, o_{t+2}) \\ &\quad + P(o_{t+4} | w_{t+3}, w_{t+2}, o_{t+3}) + .. \end{aligned}$$

Equation 41

$$\begin{aligned} &\mathrm{P}(\mathrm{o}_{t+1}, \mathrm{o}_{t+2}, \dots, \mathrm{o}_T | \mathrm{w}_t) \\ &= \sum_{k} \mathrm{P}(\mathrm{o}_{t+2}, \mathrm{o}_{t+3}, \mathrm{o}_{t+4}, \dots, \mathrm{o}_T | \mathrm{w}_{t+1}, \mathrm{w}_t, \mathrm{o}_{t+1}) \\ &* \mathrm{P}(\mathrm{o}_{t+1} | \mathrm{w}_{t+1}, \mathrm{w}_t) \mathrm{P}(\mathrm{w}_{t+1} | \mathrm{w}_t) \end{aligned}$$

Equation 42

But $o_{t+2}, o_{t+3}, o_{t+4}, \dots, o_T | w_{t+1}$ are conditionally independent on w_t, o_{t+1}, and $o_{t+1} | w_{t+1}$ is conditionally independent on w_t, therefore:

$$\begin{aligned} &\underbrace{P(o_{t+1}, o_{t+2}, \dots, o_T | w_t)}_{\beta_t} \\ &= \sum_{k} \underbrace{P(o_{t+2}, o_{t+3}, o_{t+4}, \dots, o_T | w_{t+1})}_{\beta_{t+1}} \underbrace{P(o_{t+1} | w_{t+1})}_{\substack{\text{Emission Prob.} \\ \text{(known)}}} \underbrace{P(w_{t+1} | w_t)}_{\substack{\text{Transition Prob.} \\ \text{(known)}}} \end{aligned}$$

Equation 43

$$\beta(w_t) = \sum_{k} \beta(w_{t+1}) \underbrace{P(o_{t+1} | w_{t+1})}_{\substack{\text{Emission Prob.} \\ \text{(known)}}} \underbrace{P(w_{t+1} | w_t)}_{\substack{\text{Transition Prob.} \\ \text{(known)}}}$$

Equation 44

Then the likelihood ρ(O|W) for each word can be found:

$$\rho(O/W) = \sum_{J} \rho(o, i | w)^{\text{[14]}} = \sum_{J} \overbrace{\rho(o | i, w)}^{\substack{\text{obsevation} \\ \text{symbole Prob.}}} \times \overbrace{\rho(i | w)}^{\substack{\text{Transition} \\ \text{Prob.}}}$$

Equation 45

[14] Sum rule probability: $p(X) = \sum_Y p(X, Y)$, and product rule probability $p(X, Y) = p(Y/X)p(X)$

$$\rho(O/W) = \sum_{J} b(w_1|qi_1)\, b(w_2/i_2)..b(w_T/i_T) \times \underbrace{p(x_1 = i_1)}_{\substack{initial \\ Prob.}}\, a(i_2/i)\, a(i_3/i_2) \ldots a(i_T/i_{T-1})$$ Equation 46

$$\tilde{\alpha}(O_1^t, i) = \sum_{j=1}^{S} \tilde{\alpha}(O_1^{t-1}, i)\, a(i/j)\, b(O_t/i)$$ Equation 47

$$\tilde{\beta}(O_{t+1}^T, i) = \sum_{j=1}^{S} \tilde{\beta}(O_{t+2}^T, j)\, a(j/i)\, b(O_{t+1}/j)$$ Equation 48

It is important to note that this recursion $\tilde{\beta}(O_{t+1}^T, i)$ is initialized by defining O_{t+1}^T to be a partial sequence such that:

$$\tilde{\beta}(O_{t+1}^T, i) \stackrel{\text{def}}{=} \begin{cases} 1, if\ i\ is\ a\ legal\ final\ state \\ 0, otherwise \end{cases}$$ Equation 49

where the legal final state “is one at which a path through the model may end” [1]. This means that the ordinate of the lattice point at time T must be a state in the model.

The F-B algorithm has a disadvantage which is an underflow condition coming out of the large numbers of multiplications of numbers less than unity [1]. Therefore, the α(./.) and β(./.) must scale in each step with c_t, as follows:

$$c_t = \left(\sum_{i=1}^{S} \tilde{\alpha}(O_1^t, i) \right)^{-1}$$ Equation 50

$$\tilde{\alpha}(O_1^t, i) = c_t \cdot \tilde{\alpha}(O_1^t, i)$$ Equation 51

$$\tilde{\beta}(O_{t+1}^T, i) = c_t \cdot \tilde{\beta}(O_{t+1}^T, i)$$ Equation 52

Therefore, based on the scaled forward/backward recursions, the process can be started by creating random matrices of $\pi(1)$, A and B and the training observation, $O_1^T = o_1, o_2, \dots, o_T$, then computing a new mode of the HMM, $M = [\,S, \bar{\pi}(1), \bar{A}, \bar{B}, O_1^T]$ as:

$$\bar{a}(j/i) = \frac{\sum_{t=1}^{T-1} \tilde{\alpha}(O_1^T|i)\, a(j|i)\, b(O_{t+1}|j)\, \tilde{\beta}(O_{1+2}^T|j)}{\sum_{t=1}^{T-1} \tilde{\alpha}(O_1^t|i)\, \tilde{\beta}(O_{1+1}^T|i)} \qquad \text{Equation 53}$$

$$\bar{b}(k|j) = \frac{\sum_{ot=k,t=1}^{T} \tilde{\alpha}(O_1^t, j)\tilde{\beta}(O_{1+1}^T|j)}{\sum_{t=1}^{T} \tilde{\alpha}(O_1^t, i)\, \tilde{\beta}(O_{1+1}^T|j)} \qquad \text{Equation 54}$$

$$\bar{\pi}(1) = p(x_1 = i) = \frac{\tilde{\alpha}(O_1^t, i)\tilde{\beta}(O_2^T|i)}{c1} \qquad \text{Equation 55}$$

If the likelihood has increased such that $\rho_{new}(O/W) - \rho_{old}(O/W) \ \geq\ \varepsilon$, where ε is a given tolerance, then we re-estimate the model with $M_{old} = M_{new}$.

The speech recognition problem is able to transcribe the sequence of words (a sentence) corresponding to a spoken utterance.

The required likelihood can be obtained at any time in the lattice of Figure 91 and is described by:

$$\rho(O/W) = \sum_{i=1}^{S} \alpha(O_1^t, i)\beta(O_{1+1}^T|i) \qquad \text{Equation 56}$$

By using condition of Equation 52 (end of path model)

$$\rho(O/W) = \sum_{i=1}^{S} \alpha(O_1^T, i) = \left(\prod_{\tau=1}^{T} c_T\right)^{-1} \qquad \text{Equation 57}$$

$\rho(O/W)$ is could be a small value. Therefore, to avoid numerical problems, the log – likelihood measure is used instead [1]:

$$\log \rho(O/W) = -\sum_{\tau=1}^{T} \log c_T$$

Equation 58

5.9 Recognition using the HMM

For a given (but unknown) observation sequence and a given HMM trained on certain words (model), we can calculate the log-likelihood that the HMM produced a sequence.

Thus, assume we have a vocabulary of (R) words to be recognized and that each word is to be modelled by a distinct HMM (M), as shown in Figure 85. The measurement of the observation, which are converted into VQ of size (K), are followed by calculation of model likelihoods for all possible models P(O/W_R). Finally, this is followed by selection of the word whose model likelihood is highest.

5.10 Why speech recognition is expensive and time consuming

Researchers of speech recognition have used the correct pronunciation (i.e. none include alternative pronunciations) of a word which can be found in a lexicon. But the reality of spontaneous speech includes a variety of phenomena for a speech recognition task to cope with: false starts, human and nonhuman noises, new words and alternative pronunciations.

During the training, the phonetic units will be contaminated with inadequate acoustics. Then, the overall performance of the recognizer will degrade if the phonetic transcriptions in the dictionary do not match the actual occurrences in the database.

Therefore, creating dictionaries with alternative pronunciation and functional words has required more effort by speech recognition researchers who have suggested pronunciation models to modify the dictionary. Modifications are based on hand training (i.e. applying phonological rules to a given lexicon).

However, the hand training introduces the kinds of errors described below [97]. The correct phonetic transcription of a word is the aim of the experts. But in implementation of an acoustic model it is important to note that, firstly, the correct phonetic transcription of a word is not necessarily the most frequent transcription for a given task. Secondly, actual pronunciations can be very different from the "correct" pronunciation. In spontaneous speech and in dialects a lot of alternative pronunciations are used which are not always easy to predict.

With increasing number of basic phonetic units (usually between 40 and 100) and number of entries in the dictionary, it becomes more difficult to use the phonetic units consistently across dictionary entries.

The maintainer of the dictionary software can easily miss statistically relevant forms because it is hard to say which variants are statistically relevant for a given task. Therefore, correcting and modifying what is in existence in a lexicon of phonetic transcriptions of a word is a preoccupation of speech recognition experts.

The accuracy of automatic speech recognition relies on the accuracy of describing an alternate acoustic modelling as a different sequence of phonetic units using canonical pronunciations, since, in general, there are many utterances possible producing different acoustic models. It is unfeasible to produce a distinct model for each possible variation for a speech recognition system. Necessarily, there needs to be a distinction between idiosyncratic and systematic features of pronunciation. Thus, the acoustic modelling should include both the pronunciation variation (underlying distinctive information) and predictable features of each lexical item. Therefore, the speech recognition systems normally use handcrafted pronunciation lexicons designed by linguistic experts. Because of this speech recognition is expensive and time consuming. The accepted from of a handcrafted pronunciation lexicon is to assign and maximise a probability of the time sequence of states given a word sequence. An acoustic model may have been previously trained to synthesize the speech data.

There are many available commercial products that successfully recognize certain types of speech. Also, modelling techniques for accommodating and representing the variability in pronunciation have been developing. However, there still is a gap between the theoretical representation of speech (phonemes) and acoustic representation (phones), which affect badly the accuracy of the speech recognition systems.

5.11 Error acoustic modelling

The speech recognition systems that are based on the HMM aim to demonstrate acoustic-phonetic features of the elemental sounds of a selected language by determining how well the temporal variability of speech matches the state of each HMM by fitting a frame, i.e. a short temporal window, of coefficients that represent the acoustic input.

The problem of error acoustic modelling has been studied using various strategies to increase noise robustness by robust signal acquisition and feature compensation, model compensation and robust feature extraction. In other words, the problem requires contributions from the digital communications, signal processing, information theory, statistics and artificial intelligence communities. Technically, the problem is approached in two parts. The first is in the areas of signal processing and digital communications, which are applied to the analysis and synthesis of speech in order to represent the frames of speech (units) as abstract features (e.g. linear prediction analysis and cepstral analysis). The second is in the area of statistical and artificial intelligence, which allow a statistical mapping (each state of each HMM fitting in a frame) that is constructed from the abstract phonemic units to produce their context dependent realization as surface phonetic units. Therefore, the aim is by definition less abstract and less variable than for acoustic realizations.

Despite, the fact that HMMs have some limitations, the method has worked extremely well for certain types of speech recognition problems. A frame of speech in a word is influenced by other surrounding frames, which form the word. One of the HMMs limitations is the assumption that successive observations, $P\big(O_1^T;\ o(1),$ $o(2), \dots o(T)\big)$, forming frames of speech that make a word, are independent sequences. Therefore, the probability of a sequence of frames can be written as a product of probabilities of individual observations $P(O_1^T;\ o(1), o(2), .. o(T)) = \prod_{i=1}^{T} P(o_i)$. A second limitation that the probability of being in a given state at time *t* is only dependent on the state at time t-1 , which is inappropriate because speech sounds have dependencies often extending through several states. The assumption that distributions of individual observation parameters can be represented by a mixture of Gaussian or autoregressive densities is the third limitation [98].

Due to these limitations of the HMM models (i.e. imperfect statistical mapping), overfitting occurs which describes random errors or noise instead of the underling relationship. The problem is that maximum likelihood estimation (MLE) of the joint probability of training data gives some finite error rate of incorrectly decoded label frames. From the point view of machine learning, the problem is that of unobserved data (unseen data). The selection of a model according to its accuracy on the training dataset, instead of its selection accuracy on an unseen test dataset, results in a high chance it has lower accuracy on an unseen test dataset. This is due to the lack of the model generalization.

The error of embedded minimum classification is measured by competing HMMs (in the stage of the maximum likelihood estimation). To reduce this error, researchers have proposed an alternative, factorial method (a stream) amalgamated with Viterbi decoding, in order to make modifications to the standard HMM topology. The reason is to minimize error within the frames level [99, 100]. The result is a reduction in the classification error occurring between the likelihood incorrect paths (e.g. N-best paths) selected from HMM and the correct path.

A HMM error model has been made by employing multi–stream transformation, including factorial methods. The transformation can, for example, transfer the feature vectors of a data sequence into an observation space. When the data is transferred from the observation space into some normalized space, it is considered as independent and identically distributed data. The model is based on using a transformation stream in synchronism with the model stream (i.e. simple single state) which means it is an approximation to the discrete stream system without a dramatic increase in the number of model parameters. In 1999, The Hidden Markov Model Toolkit (HTK) was introduced as a portable toolkit for building and manipulating hidden Markov model [100]. An up-grade (HTK 3.4.1) has now been released. Gales [101] has suggested a 2-stream factorial model using a feature-space transformation stream. Xiong et al [102] investigate a generalization capability to improve robustness when the testing and training data are from different distributions, in the context of speech recognition. Taemin et al [99] minimized the number of incorrectly decoded labels in a frame by applying a smooth function that is arbitrarily close to the exact frame error rate and minimize it directly using a gradient-based optimization algorithm. Bing Hwang et al [103] differentiate the method of classifier design by way of distribution estimation and the discriminative method of minimizing classification error. The authors compare the traditional maximum likelihood method (based on the distribution estimation formulation) with a minimum classification error metric (MCE).

The natural way to minimize error detection in acoustic models is to reduce recognition errors. The aim is to improve the robustness of statistical speech transcription systems. We argue that it is possible to modify the traditional methods of MCE by adding a parallel layer. This layer we call a realization layer (RL) which can be employed to correct the performance of the HMM both in the training and recognition phase. The RL is thus a complementary layer. It can be integrated with any other methods of MCE and corrects the remaining statistical mapping error from MCE methods in the recognition phase. The RL is a word spectrogram matching technique. To clarify our argument, we initially explain image recognition in the next section and elaborate upon this in the subsequent section.

5.12 Testing the performance of HMMs by two dimensional matching of SSI of words

An IWSR can be used to demonstrate the avoidance of the overfitting problems of ASR. We have implemented the HMM procedure in 5.8, using the database of ten digits (0:9), i.e. test No.5. Each word contains ten samples and so a total is 100 samples were used in the training phase of the HMM model. If the same data as the training set is used to run the recognition phase, the HMM model fails to recognize 9 of the 100 word inputs.

After analysis of the results, we have determined where errors are made. The guesses of the HMM (word digits) are ranked in likelihood of ten HMM options. The results are depicted in Figure 12.

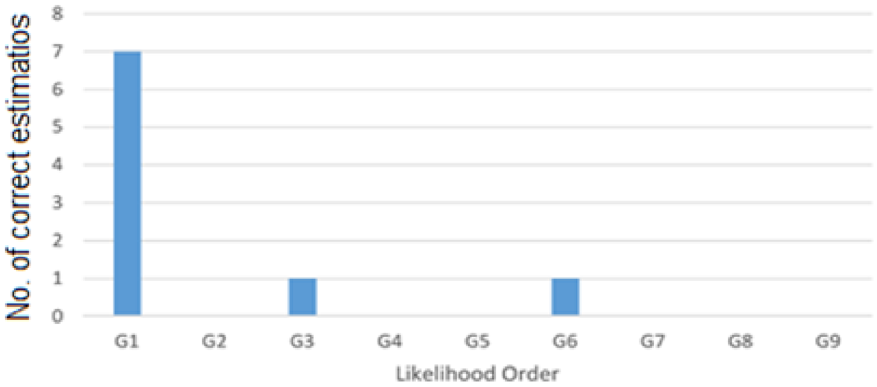
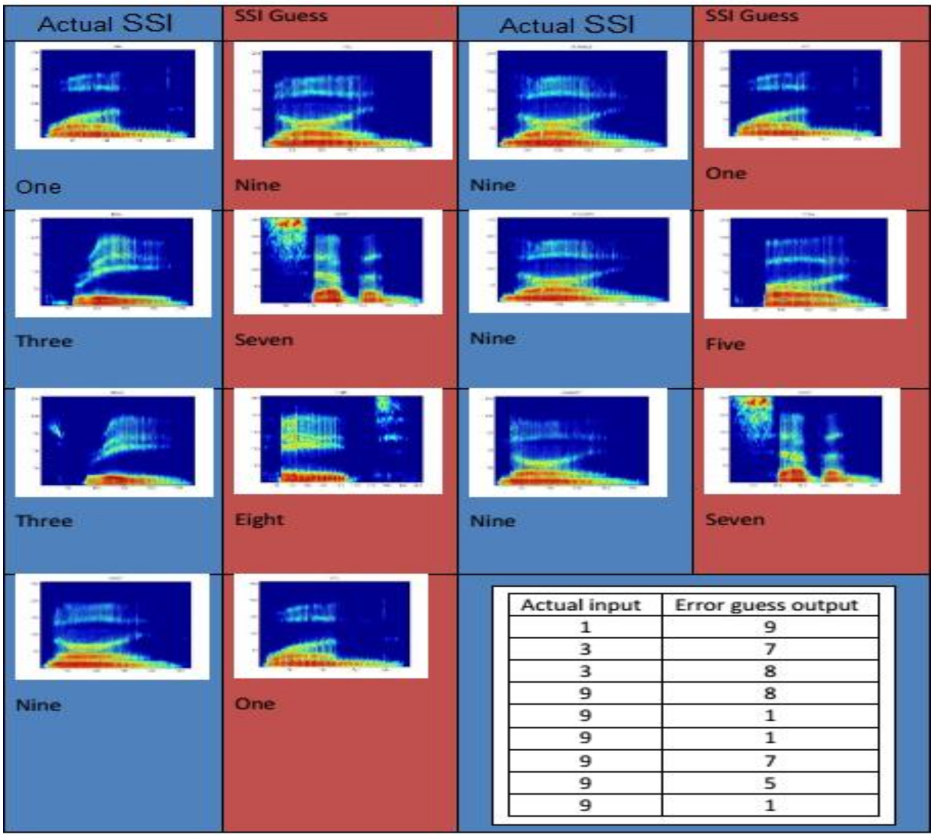


Figure 95 Chart showing where the right guesses (G) are located in likelihood options of the HMM.

The results of the test No.5 are shown in Figure 96, in which the errors of the HMM are obvious and can be corrected employing a 2-D template matching technique. As an example, the first error (as shown in Figure 96) is that the input of word digit “one” is recognized as the word of digit “nine”. However, there is a clear deference between the SSI of words for the digit one and nine when viewed as the 2-D SSIs shown in Figure 96.



Actual input	Error guess output
1	9
3	7
3	8
9	8
9	1
9	1
9	7
9	5
9	1

Figure 96 Illustration of how it is possible to detect errors in the guess of the HMM by employing the SSIs.

Therefore, adding a new layer to the recognition system to review the decision of the HMM by using the 2-D information provided by the SSIs of words can be effective in improving the error rate.

5.13 Realization layer (RL)

The RL is proposed to consist of three main parts: image recognition method, SSI library, and stack memory. The SSI library has SSIs of all the words of a model. The output values of the maxi mum likelihood stage of the HMM are mapped into primary keys of the SSI library and put in their order of maximization in stack memory (the last entry of which should be the highest likelihood).

The content of the stack memory is pushed one by one to the output buffer and calls the SSI of a word from the SSI library words. This processing is repeated until there is conformity to the HMM recognition results. Conformation can be achieved by matching the same word input of the HMM after it has been stored as a SSI temporarily in a buffer memory to be ready for matching with a 2-D image called from the SSI library as shown in Figure 97. If the matching result is negative (which occurs rarely since the first option is the highest likelihood from the HMM stage), then the next option in the stack memory will be called, and so on, until the SSI matching is positive and so the correct word recognition is achieved.

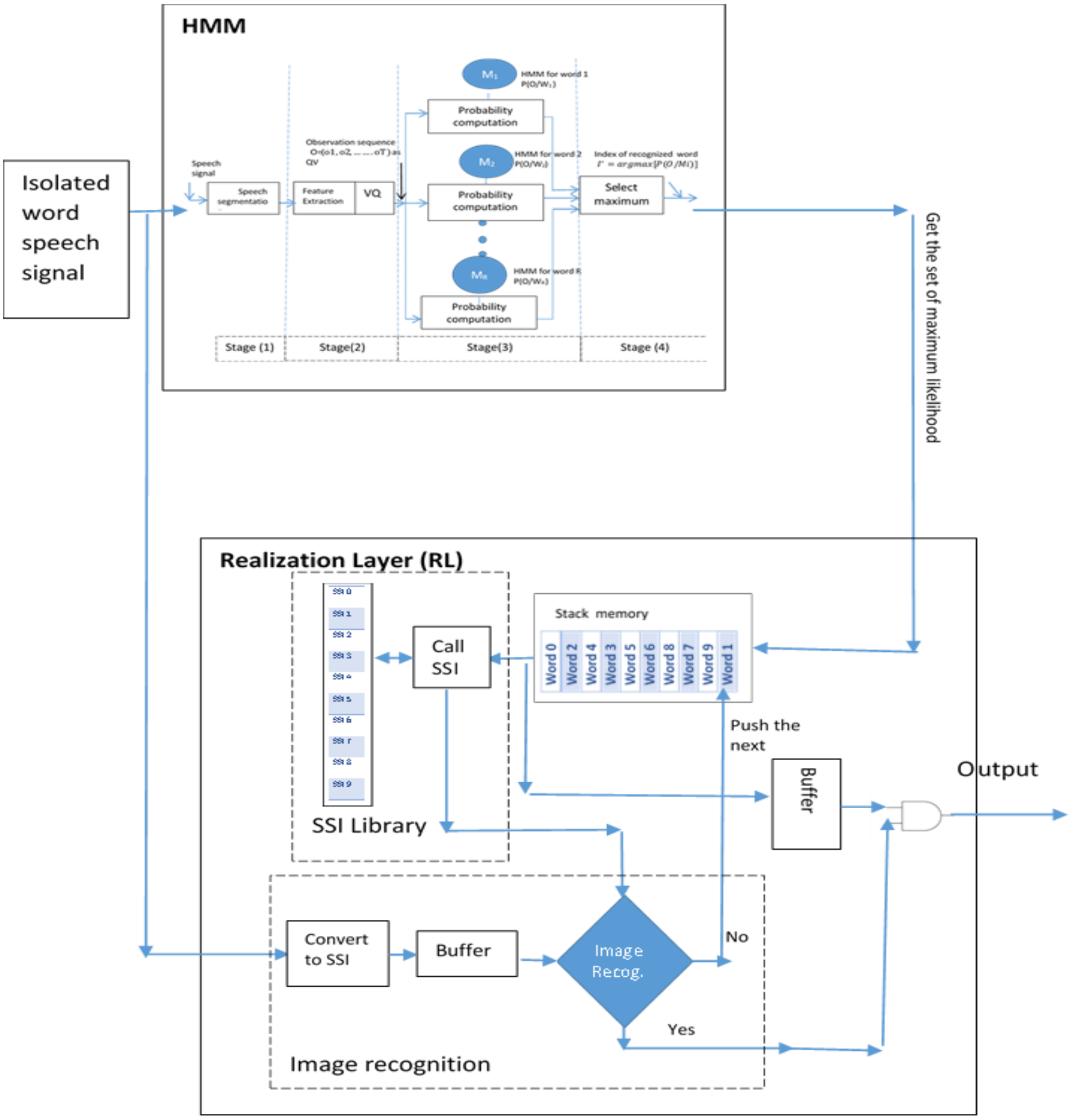


Figure 97 Organization of the RL to improve HMM performance.

As stated previously, the image recognition is performed by 2-D template matching. Pre-processing of the input by a HMM produces a ranking, which is revised by the template matching process to correct the erroneous output of the HMM. The results produced in the test No.5 described are shown Figure 98. It is apparent that the performance of the hybrid of the HMM/RL processing systems is perfect without any error, and the MCE has been reduced to zero.

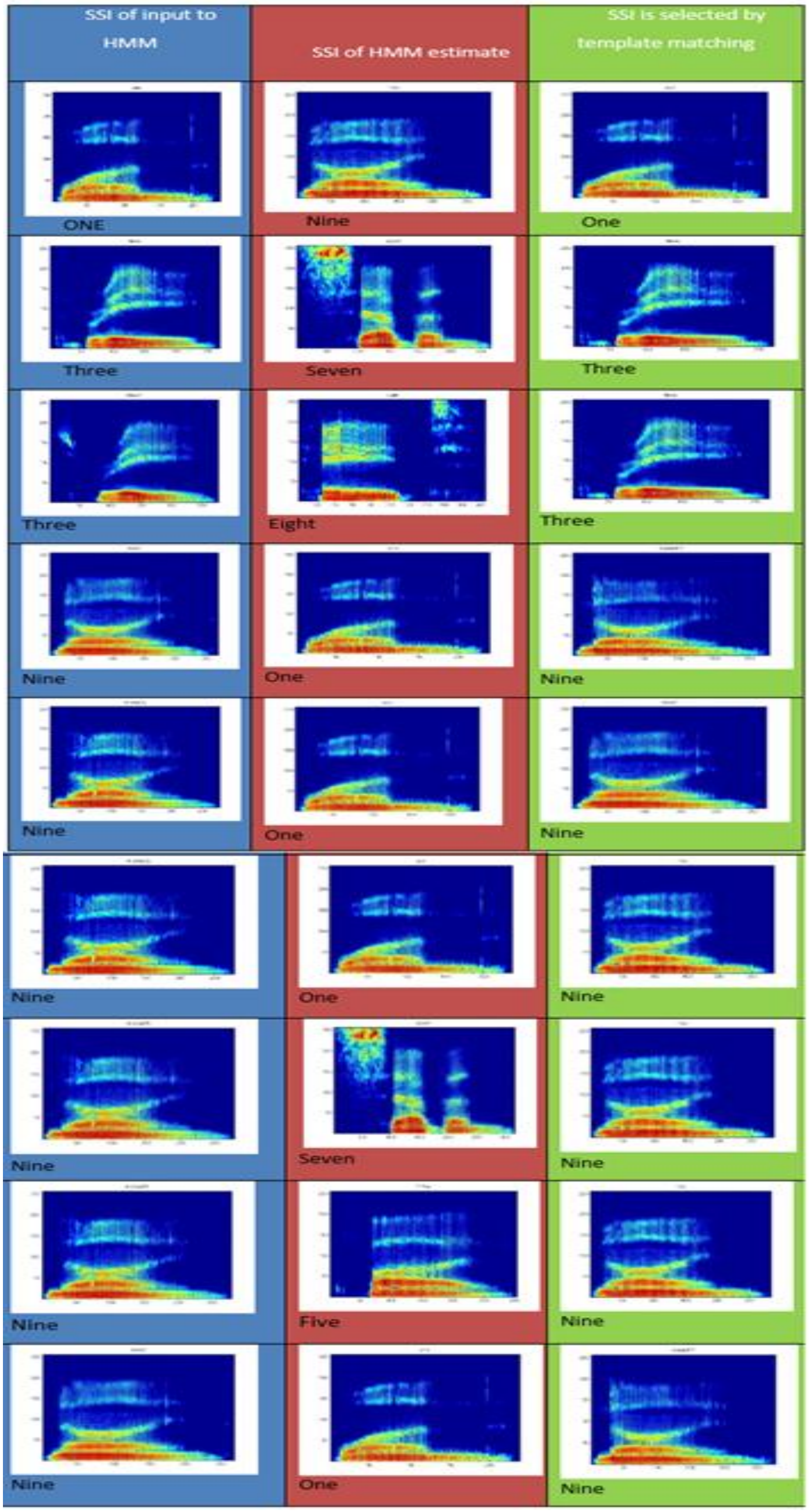


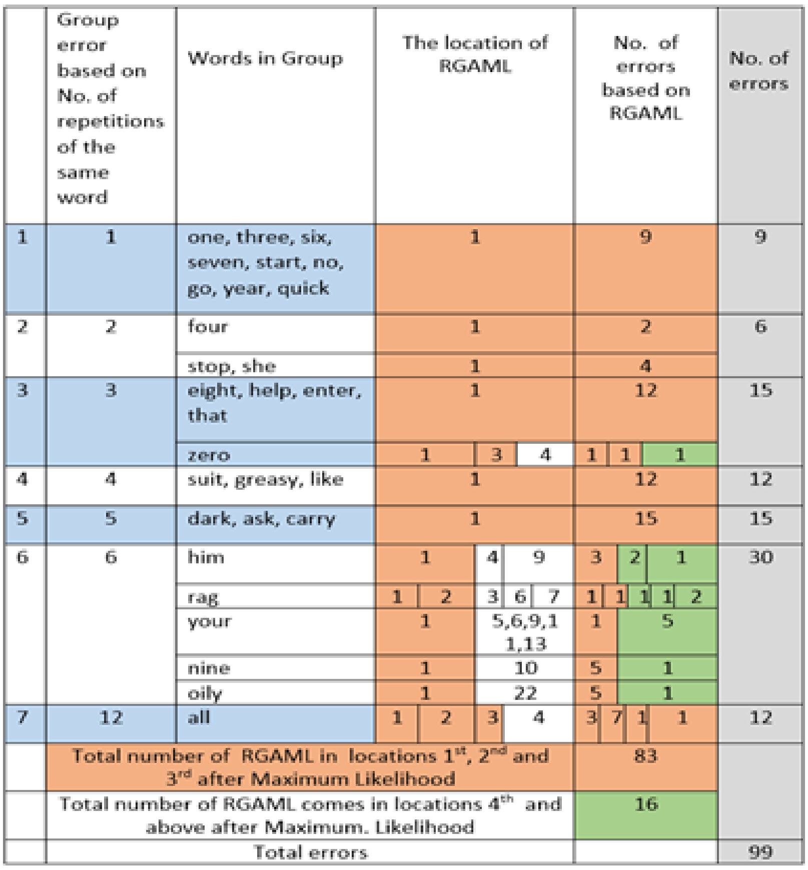
Figure 98 Illustration of the ability of 2-D template matching to correct mistakes in HMM recognition.

Looking for further confirmation for our model, test No. 6 was performed. We have used the lexicon defined in section 4.5. The data set consists of the same 10 digits and some other words extracted from TIMIT.

The HMM gives a statistical analysis. Therefore, each training of the Baum Welch algorithm with the same data gives a different stochastic transition matrix $A = (a_{ij})$,and so a different success of recognition since there are numerical errors in calculation of the K-mean classification and clustering error estimation in the training phase. This is why modifications based on hand training is one of way of improving HMM recognition.

The data of test No. 6 has been run in training phases first, then has been run on recognition tests several times, each run giving a different Baum-Welch matrix. If we define the recognition efficiency of the Baum-Welch matrix as: $1 - \frac{number\ of\ unrecognized\ words}{the\ total\ number\ of\ words\ \ tested}$, then the efficiency of the HMMs were between 0.8210 and 0.6546. We chose the Baum-Welch matrix with minimum errors to integrate with our RL (2-D template matching layer) for mitigating the HMM's errors. The test shows that the best HMM result has 99 errors for 553 word inputs. The 99 errors occurred in 7 groups depending upon: 1) the number of repetitions that fail to recognize the same word (as shown in Table 4); and 2) the location of the right guess is after the maximum likelihood (RGAML) in the likelihood vector of the HMM. The RGAML is the target for our RL to correct the recognition errors of the HMM. Therefore, the RGAML provides the input used to analyse the test results.

Table 4 Details of the HMM errors for test No.6.



	Group error based on No. of repetitions of the same word	Words in Group	The location of RGAML	No. of errors based on RGAML	No. of errors
1	1	one, three, six, seven, start, no, go, year, quick	1	9	9
2	2	four	1	2	6
		stop, she	1	4	
3	3	eight, help, enter, that	1	12	15
		zero	1 3 4	1 1 1	
4	4	suit, greasy, like	1	12	12
5	5	dark, ask, carry	1	15	15
6	6	him	1 4 9	3 2 1	30
		rag	1 2 3 6 7	1 1 1 1 2	
		your	1 5,6,9,11,13	1 5	
		nine	1 10	5 1	
		oily	1 22	5 1	
7	12	all	1 2 3 4	3 7 1 1	12
	Total number of RGAML in locations 1st, 2nd and 3rd after Maximum Likelihood			83	
	Total number of RGAML comes in locations 4th and above after Maximum. Likelihood			16	
	Total errors				99

Figure 99 shows the histogram of the RGAMLs in the HMM likelihood vector (the vector of lexicon words of the study case). The analysis shows that the HMM recognition has the same behaviour as in test No.5, most of RGAMLs being located in the first and second location in the likelihood vector, which accounts for around 84 % of the total errors occurring.

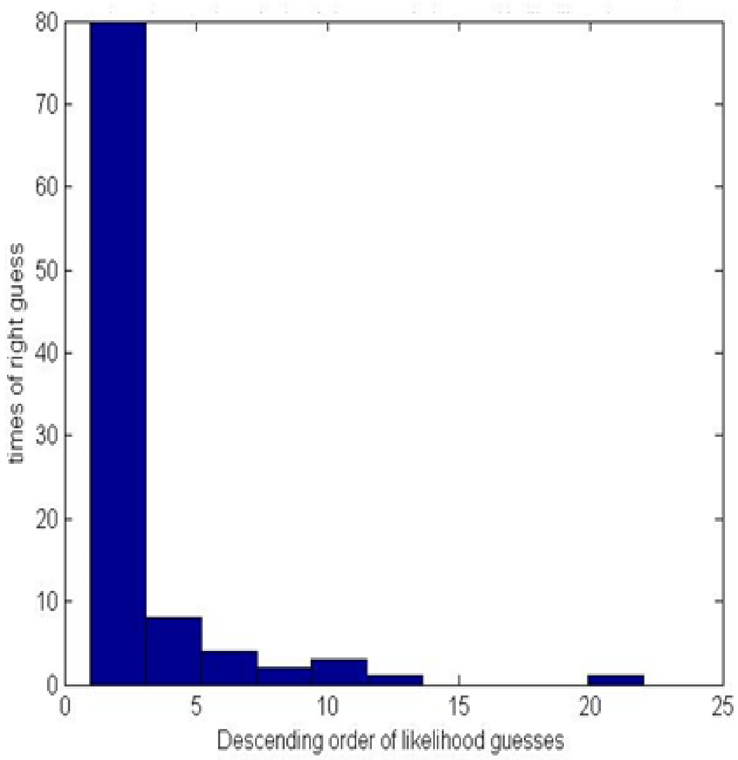


Figure 99 The histogram shows where the correct guesses are located in the likelihood vector of the HMM of the lexicon words in case study No.6.

The location error of the RGAML in the likelihood vector and the number of occurrences of the same error are counted as worst guesses of the HMM recognition. In this test, the worst guesses of the HMM are the words ‘oily’, ’your’, ’nine’, ’him’, ’rag’, and ’all’. The RGAML of ‘oily’ occurs once in location 22 and five times in the 1 location. The RGAML of ‘your’ occurs five times in locations 5, 6, 9, 11, 13 and once in the location 1. The RGAML of ‘nine’ occurs once in location 10 and five times in the location 1. The RGAML of ‘him’ occurs once in location 9, twice in the location 4 and three times in location 1. The RGAML of ‘rag’ occurs twice in location 7, and once in the locations 6, 3, 2, 1. Finally the RGAML of ‘all’ occurs twelve times, six of them at 2, three at 1, one at 3, and one in location 4 (after the maximum value of the likelihood vector of the HMM). The 99 errors are fed to the RL (for 2D template matching). The RL again demonstrates its ability to correct all HMM recognition errors. The RL corrections have been displayed in two categories dependent on the depth of the RGAML in the likelihood vector.

Firstly, the errors of the RGAML of depth equal to and greater than the 3rd location in the likelihood vector are illustrated in Figure 100 to Figure 104 to show the RL's path when finding the correct match for the HMM input. The actual SSI word (i.e. the input to the HMM) is indicated by a green rectangle and the RL detection by a black rectangle around the individual frame.

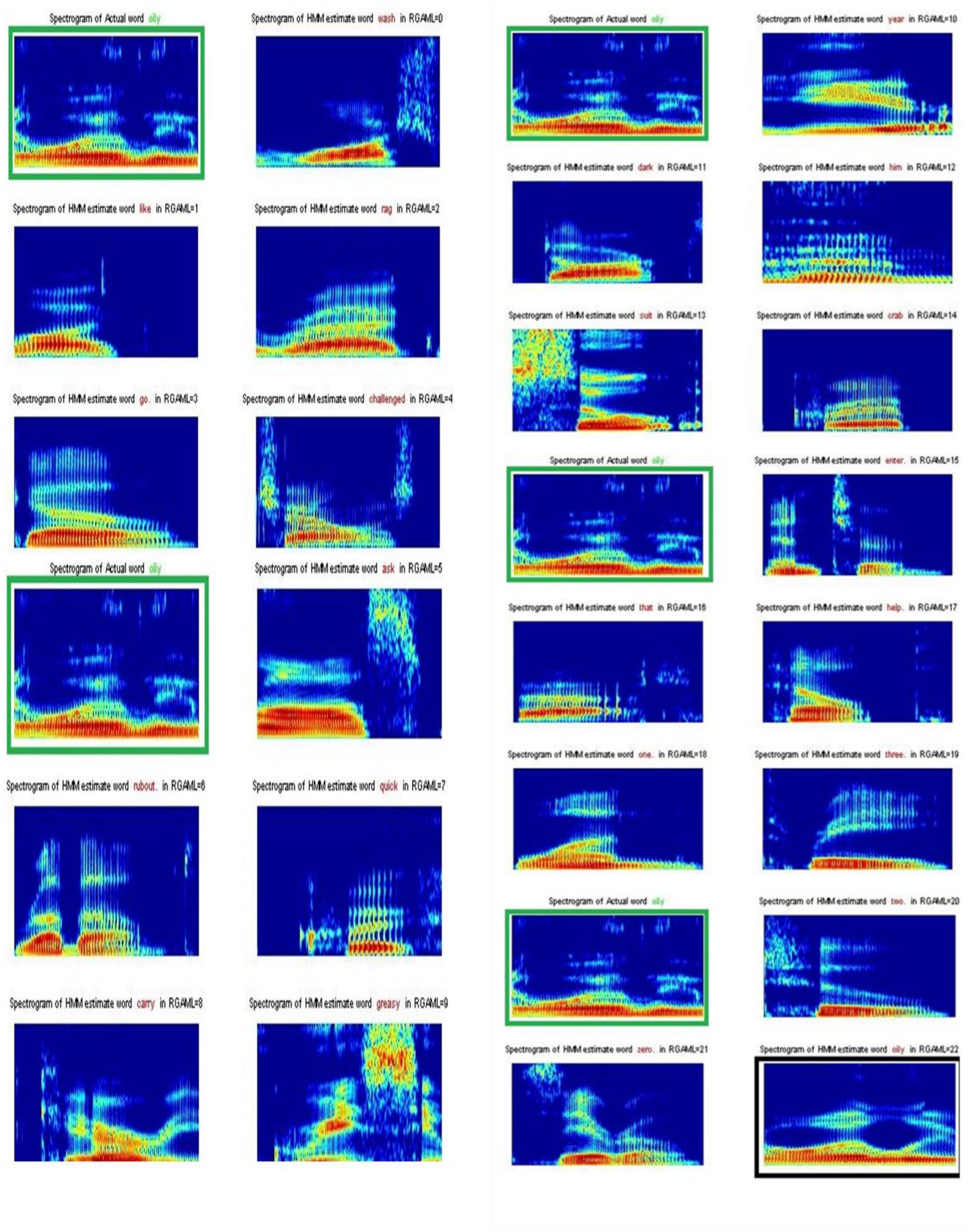


Figure 100 the RL path when detecting the correct SSI of the word 'oily' through the maximum likelihood vector of the HMM.

The RL detects the deeper error of the word ‘your’ in location RGAML =13 as shown in Figure 101.

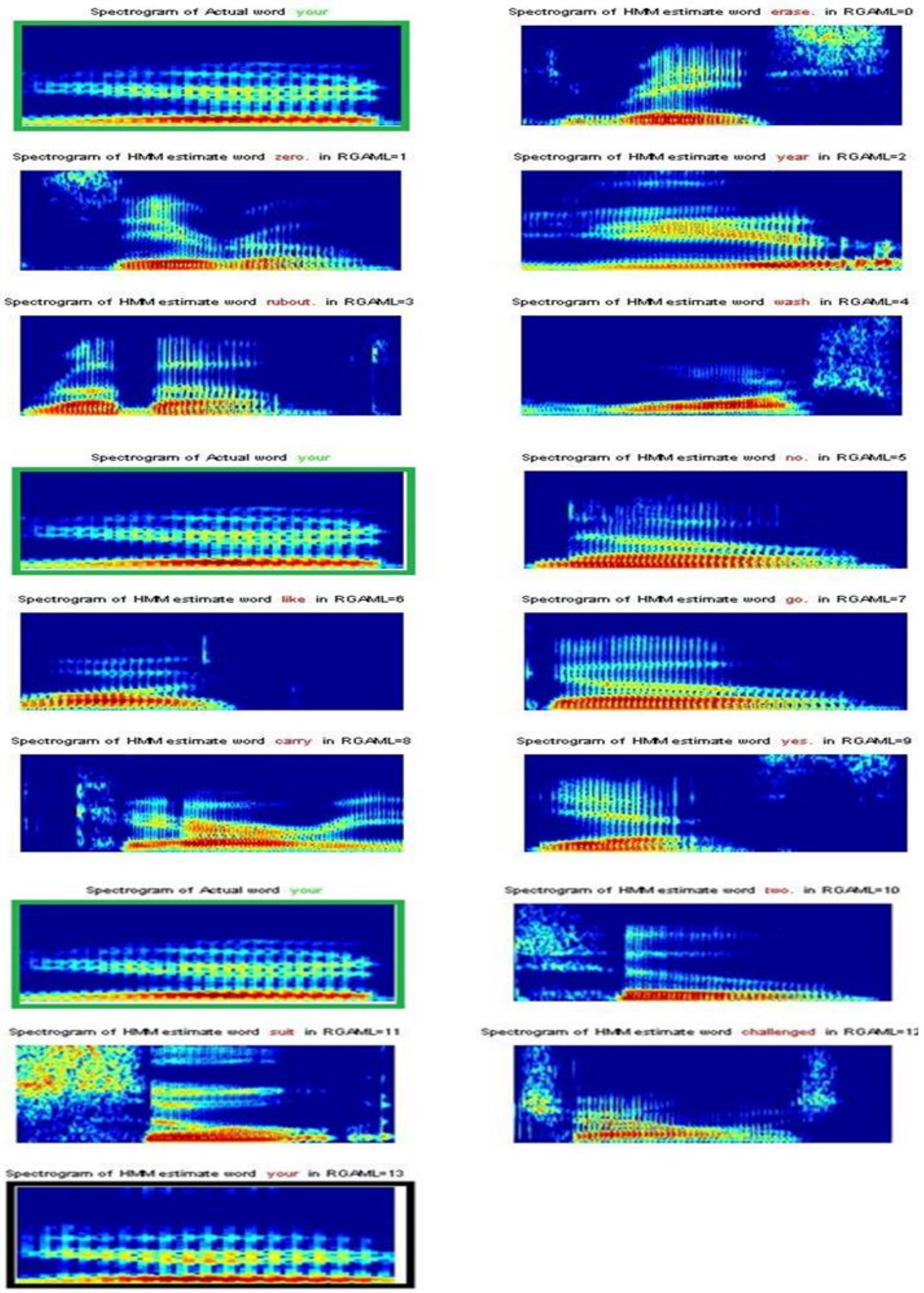


Figure 101 the RL path when detecting the correct SSI of the word ‘your’ through the maximum likelihood vector of the HMM.

The Figure 102 shows the RL detecting the deeper error of the word 'nine' in RGAML =10.

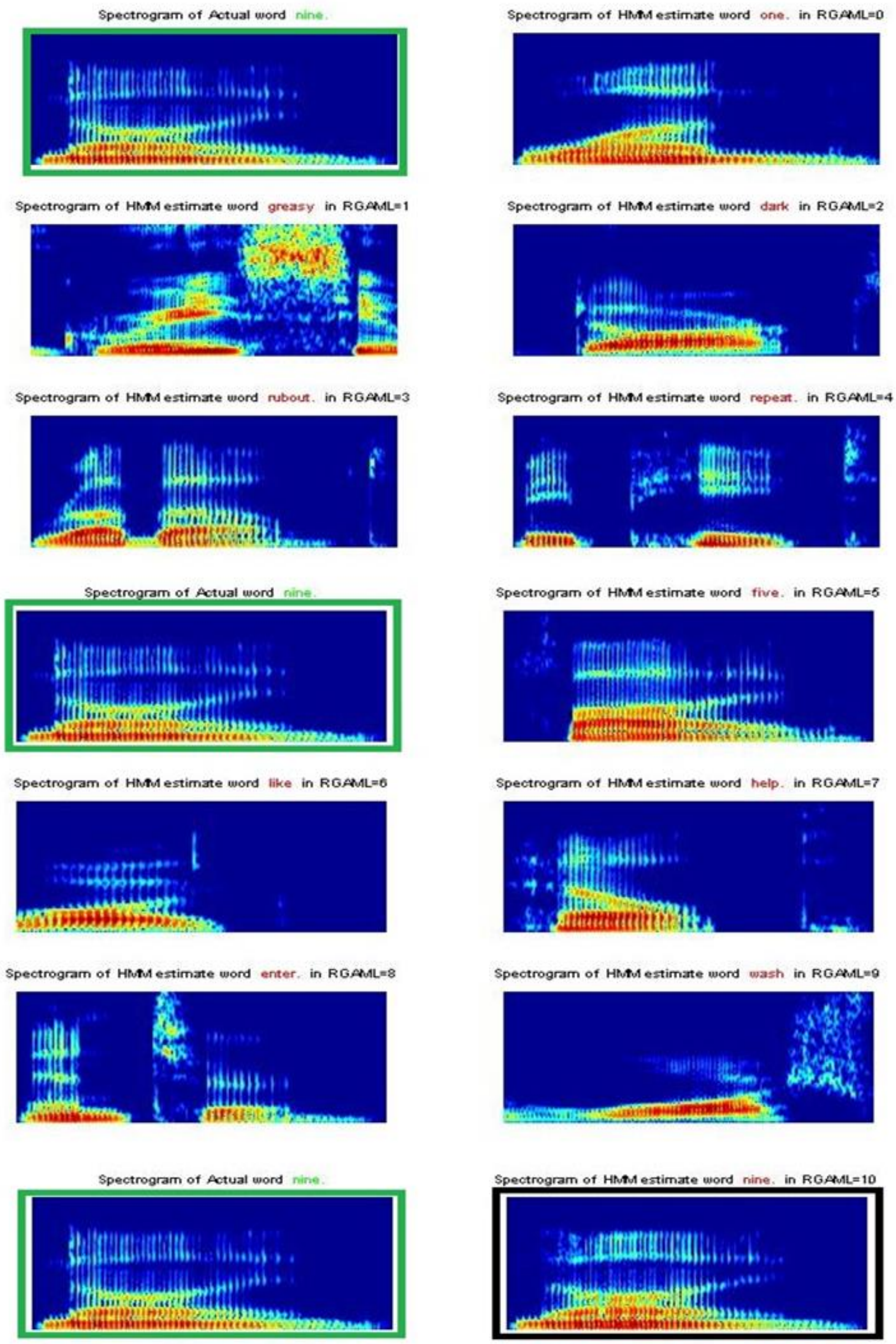


Figure 102 the RL path when detecting the correct SSI of the word 'nine' through the maximum likelihood vector of the HMM.

The Figure 103 shows the RL detecting the deeper error of the word ‘him’ in RGAML =9.

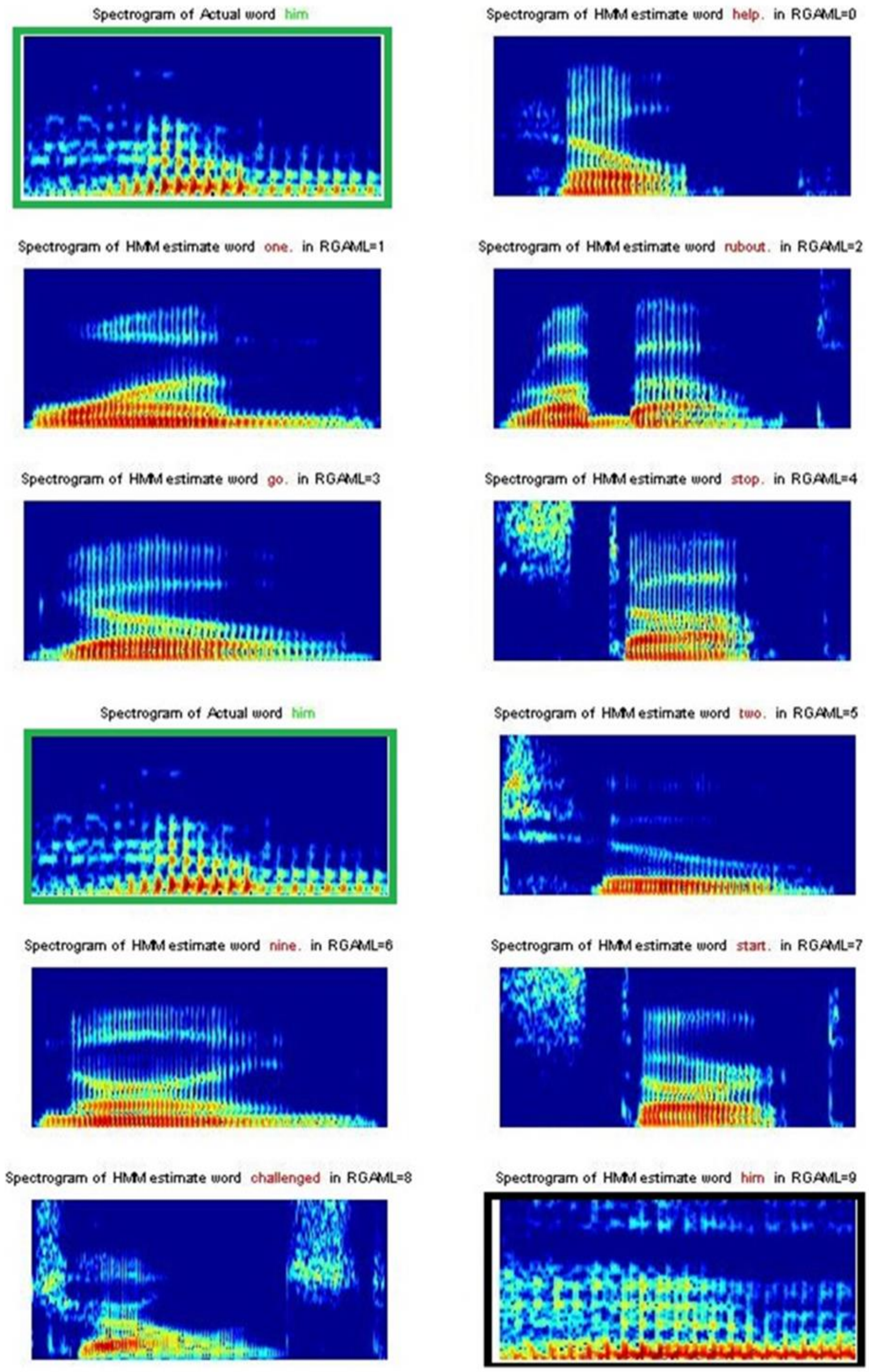


Figure 103 the RL path when detecting the correct SSI of the word ‘him’ through the maximum likelihood vector of the HMM.

The Figure 104 shows the RL detecting the deeper error of the word ‘rag in RGAML =7

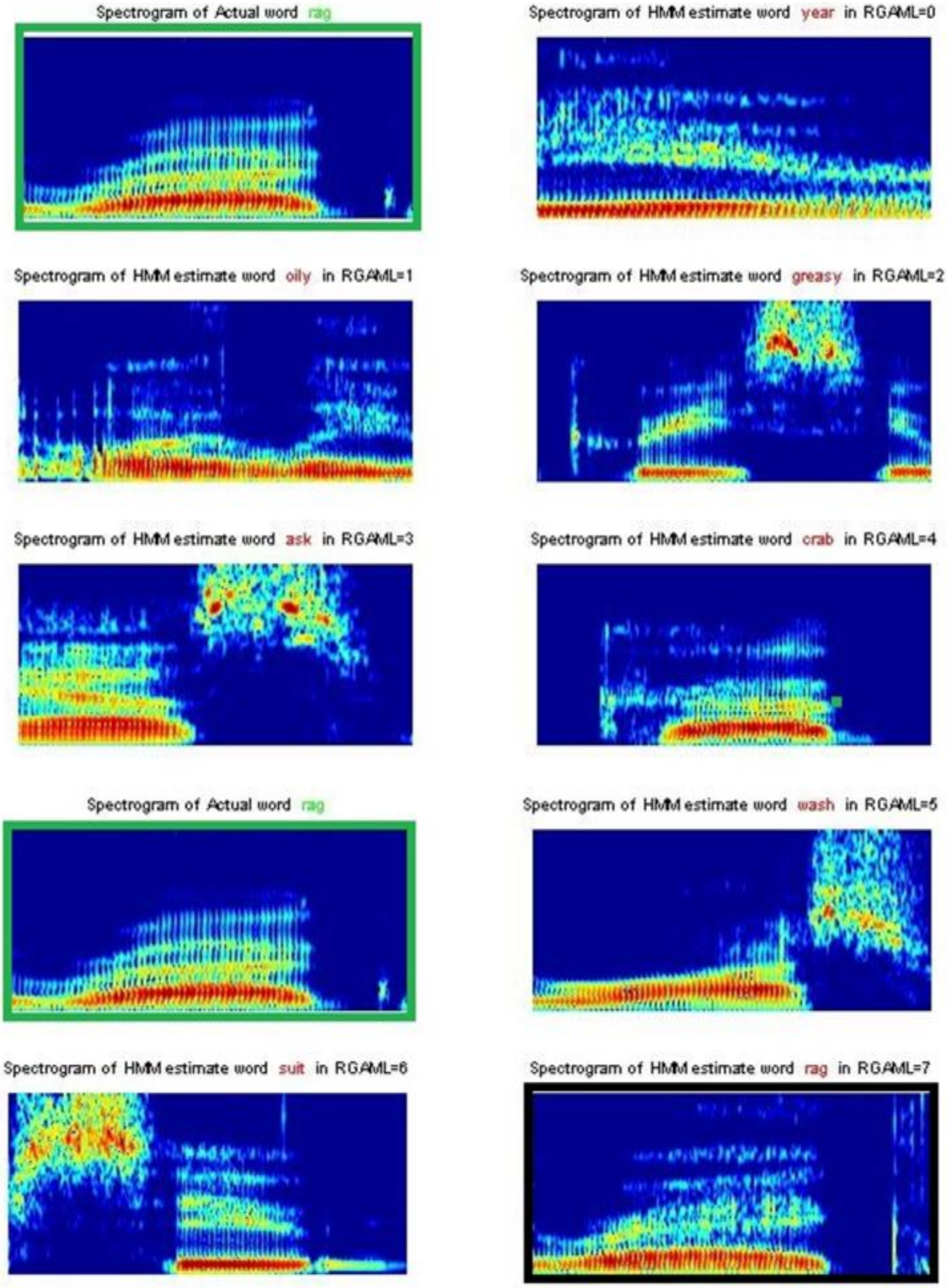


Figure 104 the RL path when detecting the correct SSI of the word ‘rag’ through the maximum likelihood vector of the HMM.

Secondly, all the other errors which are of RGAML depth equal to 1, 2 and 3 and are displayed in Figure 105 and Figure 106.The SSI of an actual word input to the HMM is displayed in the left hand column, in the middle column is shown the SSI of the HMM estimation, while the right hand column shows the RL detection. The pictorial illustrations of the RL show clearly that the RL can easily correct the HMM errors. On the other hand, there is a difficulty in matching the SSIs of the word ‘stop’ and ‘start’. This is because the start and end of both words generate the same patterns.

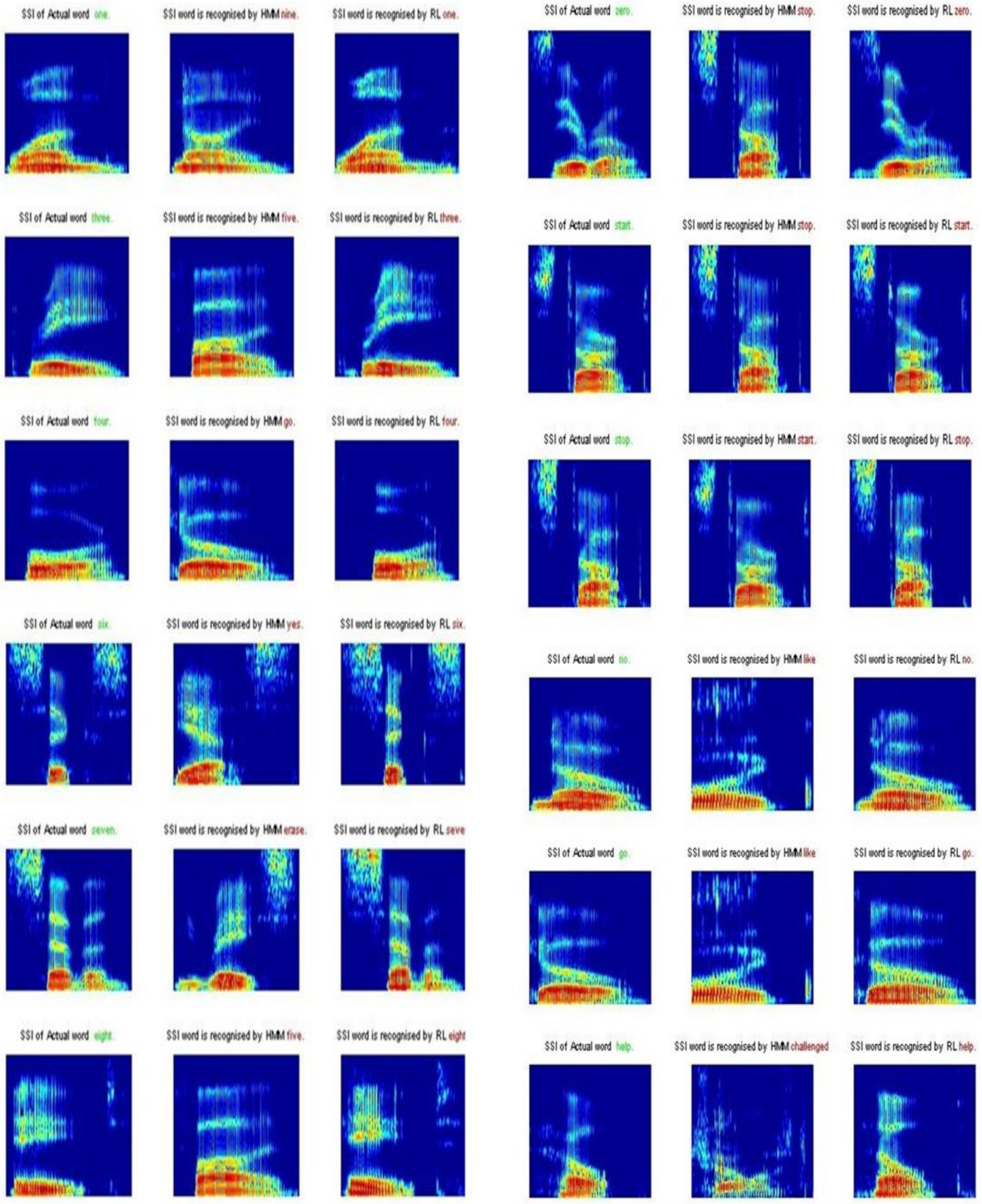


Figure 105 The RL when detecting the errors in the HMM with the RGAML depth equal to one and two (group 1).

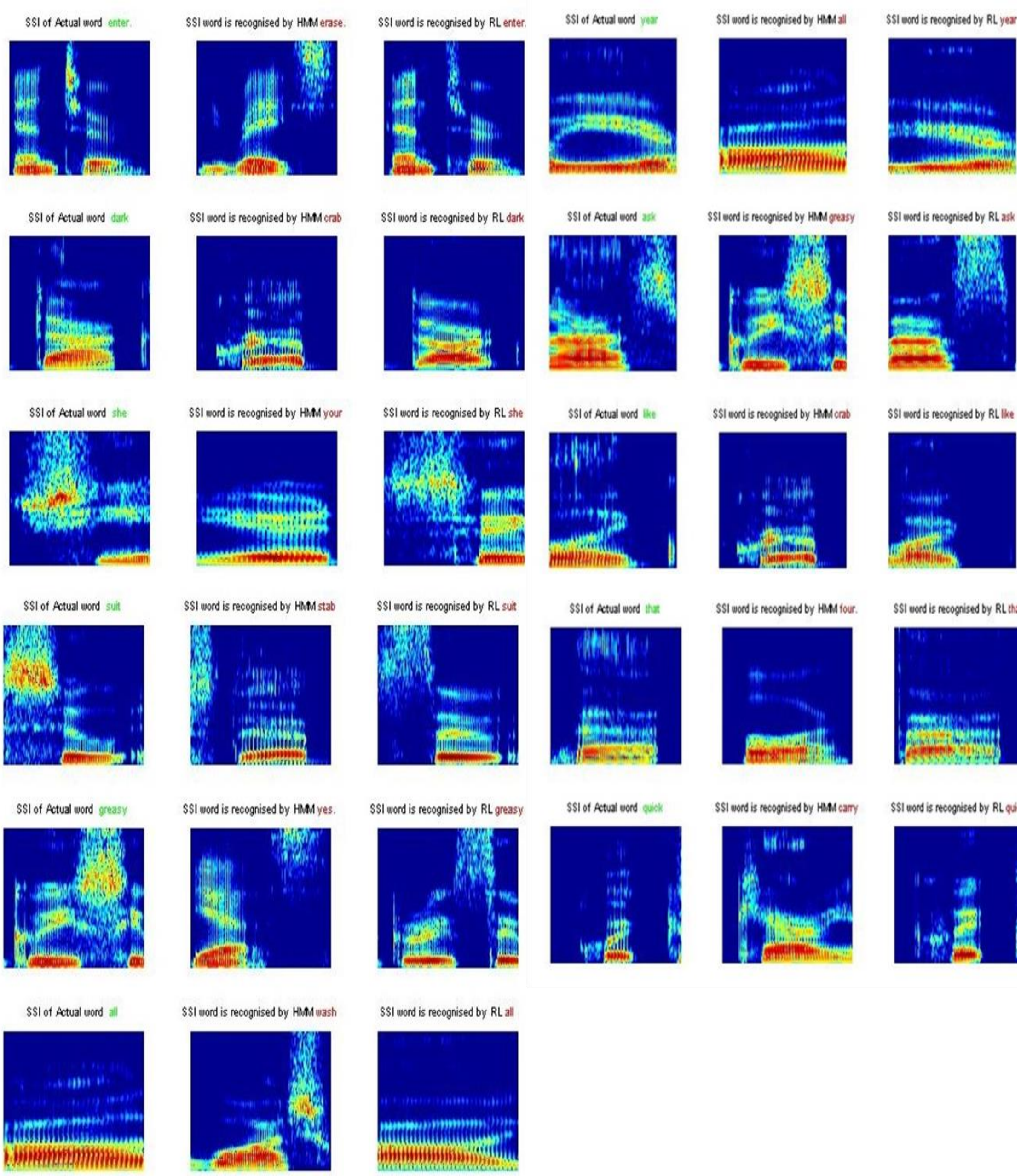


Figure 106 The RL when detecting the errors in the HMM with RGAML depth equal to one and two (group 2)

Robust feature extraction is one method for improving the performance of HMM recognitions in ASR. Methods of extracting features from the speech signal (e.g.

(LPC), Perceptual Linear Predictive Coefficients (PLP), (MFCC) all provide means of improving the HMM recognition ability. We have used LPC with in training features which is not an optimal way of reducing HMM speech recognition errors. Rather, in this chapter, we demonstrate that the 2-D template matching technique can improve upon the statistical decisions made HMM. Therefore, the selected optimum matching method of speech feature extraction can still be used in combination with our method.

5.14 Conclusion

The reason for using an IWSR is that this can be the ideal model to build on to create an ASR system. The low level of the ASR is almost similar to IWSR except there are long pauses between words in IWSR.

ASR is based on HMMs but, in the low level stages of ASR, a small gap in the statistical mapping causes a failure to produce a match of the acoustic speech signal features by the statistical process of the HMM and so some classification errors occur. Certainly, the errors in the low level recognition stage of ASR produces unavoidable errors in high level recognition.

It has been shown that each word has certain patterns of frequency transitions and can thus be recognized as a distinct image pattern. The SSIs allow for matching whole features of a word on one process which gives better recognition results than HMMs which are sequential data training. Therefore, we believe that the addition of the RL is a highly promising solution to compensate for the failure of HMMs in low level recognition. In addition, the same concept of employing SSIs can be used for whole sentences to reduce classification errors in HMM based high level recognition.

CHAPTER SIX

6 CHAPTER SIX

Conclusion and future work

6.1 Conclusion

The major focus of this thesis was to generate new speech recognition systems, based on matching the spectrograms of speech signals. Both the unsupervised training correlation of the 1 dimensional signal process and 2 dimensional correlation techniques were implemented and tested. The uniqueness of the study is the use of the 2 dimensional correlation, to improve current speech recognition systems and establish new speech recognition based on image processing. After more than seven decades of research in the field of speech recognition, a new approach may seem ambitious in the light of current information around speech recognition. This study is a comprehensive project of new perspectives on speech recognition. The comprehensive study includes: theorizing a new approach for speech recognition, supporting it with wider previous research, validating it with a lexicon of 533 words, and integrating it with the current HMM based speech recognising system. Consequently, we hope this study will draw attention of researchers for increasing research in the field of SSIs.

Chapter 2 focuses on the TFR of a speech signal. A speech wave signal has characteristics in both the time and frequency domain which is used for classifying speech (in any language) in fundamental units. So, recognising a speech segment needs to compare combined information of both the time and frequency features. A speech signal is a nonstationary signal, and the linear transformation of the speech signal into the TFR is an uncertainty process. The problem in the TFR is that the location in time or frequency domain (uncertainty principle) cannot be coherently transformed into a TFR. The small duration signals have inherently large bandwidth within the signal. The way to overcome this problem is by nonlinear (bilinear) representation. All the research on the bilinear TFR are on continuous signals, while the speech signal is a discrete signal. In the spectrogram, the time varying spectrum of the speech signal does not have to be accurate for recognising a word by SSI matching. However, the speech frequency features (frequency locations in the spectrum) are mostly important. Therefore, the spectrogram is a linear representation of the speech signal and the most powerful in speech analysis. The discrete WT is different from the STFT in the sense, that the window width can be changed in the WT to get high frequency resolution, so it is more useful for speaker identification. It can be used in future studies of the SSI for recognising the stress (louder, longer, and higher in pitch) and rhythm (two types of stresses; syllable stress in words and word stress in sentences) in a speech language.

The SSI is a simple and effective way for analysing a speech signal, used in this study. The SSI has some advantages in describing transitions and fluctuations of a speech signal and so it has been used for classifying speech into patterns. Secondly, the SSI has been addressed in many studies making it rich enough to support new approaches of classification speech unit's, as has been done in Chapter 3. Thirdly, the SSI is easy to implement in real time speech applications as it can be integrated with the current speech recognition applications, as described in Chapter 5.

Speech recognition problems occurring for uttered words can be classified into two: firstly, satisfying the visual speech segmentations of the speech in units, which is influenced by the co-articulatory and the start-end point of speech segments. Secondly, satisfying the speaker variations, which modifies the frequency locations in the spectrum and an uttered word duration in the TFR. The visual process of speech signal segmentation (phonemes) is related to understanding a writing system, which is important for verifying the recognition of the word level in ASR. Chapter 3 defines a new writing system based on the SSI patterns. In this writing system, it is not necessary to satisfy the phoneme interpretation but only the word level recognition by matching its SSI patterns. Also, in Chapter 3, boundaries for defining the redundancy of the code (the entropy) of this new writing system, that we have called the SIR-CODE, has been discussed.

The SIR-CODE refers to transferring the speech signal to an artificial domain (the SSI) that allows the classification of the speech signal into segments. This classification is based on the speech signal behaviour (transition and fluctuations of signal frequencies) in the new domain rather than the visual segmentation of the speech process into phonemes. This solves the issue of trying to fill the gap between theory and practice of phoneme recognition by matching the SSI patterns at the word level. Since, the new domain classifies the speech signal into segments, it is independent of the co-articulatory influences of speech sounds. Besides, these patterns have clear start-end points. This resolves the problems of endpoints in speech recognition, leading to a high recognition performance.

In this study, the phoneme properties have been used to classify the SIR-CODE to satisfy the mental process of understanding and speech recognition. The phonemes properties used to satisfy the L1-2 entropy of the SIR-CODE, is classified in the English language into five main types: FFP, SFFP, SAFP, LAFP, and G. Each type consists of classes based on the vowels and consonant behaviour of sounds, except the G Type. Hence, this can be applied to all spoken languages. The class is formed of different combinations of frequencies. These types are subsumed into two patterns under the L1-1. The L1-1 entropy has been verified by matching the SSI patterns in Chapter 4.

The SSI matching is suitable for satisfying the LI-1 entropy of finite words (small lexicon). While a natural active language can contain a huge number of words, this does not contradict the use of SSI matching. It is an independent way for demonstrating the SIR-CODE recognition ability (big lexicon). The missing phonemes in the SSI patterns can be a shortcoming for the visual process of speech segmentation of phonemes. On the other hand (in Chapter 4), the SSI matching shows that it contains underlying information that has the capability for speech recognition at the word level.

Using dynamic time warping on the SIR-CODE segments can help verify a huge word dictionary. This can provide an entropy measure to allow the SIR-CODE achieve the ASR, based on SSI matching only. The feature of the SIR-CODE segments is that it has clear endpoints, which makes the dynamic time warping process accurate for ASR. The higher level of SIR-CODE provides more entropy, in terms of a high recognition performance, but needs more sophisticated algorithms to reach that level of classification. We believe that the SIR-CODE L1-3 can be achieved by modifying the same methods used to classify the SIR-CODE L1-1. The higher level of SIR-CODE can be achieved by using Statistical Learning/Pattern Recognition on SSI patterns to estimate more features and encode them into symbols. Symbols should be enough to satisfy perfect coding of the speech signal of a language into the SIR-CODE.

For the purpose of establishing a link between the visual process of speech segmentation and the SSI pattern matching process, there is an initial introduction of image processing methods for recognising the SSI patterns in Chapter 3. Image processing methods have been discussed in more detail in Chapter 4. The results in Chapter 4 show that the SSI pattern matching can overcome the speaker variation problems easily. The speaker variations appear mostly as differences in the time dimension of the SSI, which is reflected as a challenge in the SSI pattern matching. This issue has been resolved in Section 3.12.3 by normalising the width of the two SSIFSs in the compared SSIs. The SSI width is modified based on the proportion of the normalised SSIFSs in the compared SSIs. The process of normalising is suggested as a part of the process, we called the pre-processing element of SSI matching. This process has shown very promising results in matching the same word, pronounced by different people. It works effectively with an SSI that has one pattern of SSIFS. Therefore, classifying the SSI of a word into SSIFS and SSIRS becomes necessary to normalize the SSIFS and further reduce the speaker variation for the SSI of a word, uttered by different people. This issue has been dealt with in Chapter 4. Chapter 4 also addresses the image scaling problem in image processing of the SSI and discusses improvements in the SSI matching for reliable recognition.

Some SSI patterns have regions that are formed by a noise-like region of pixel values. Therefore, there are no continuous lines or pixel values in the SSI patterns. Thus, finding the most likely matching of the SSI patterns is not straightforward, neither intensity-based (correlation intensity matching) nor feature-based (corresponding matched between points and lines or contours). The SSIFS is one of the two main patterns in the SSI and has more regularity than the SSIRS patterns. The SSIRS are formed of scattered regions appearing as a noise structure in the SSI with some more regular patterns. Therefore, the SSIRS is affected adversely in both the intensity and feature based matching between two SSIs of the same uttered word (as shown in Section 4.6).

The reason behind this unfavourable matching is that the random scattered regions of the SSIRS match with the other types of SSIFS patterns, located in the same area of the SSIs, which gives random false matching. In contrast, the SSIFS patterns are more solid and have stable values of matching for recognising the class of the SSIFS type. The parsing process of the SSI patterns has been used to compensate for the influence of the SSIRS pattern matching on the likely matching value.

The SSI parsing process that classifies the SSI into SSIFS and SSIRS has been implemented in Chapter 4. The SSI parsing provides more entropy of the SIR-CODE in terms of improvement in the SSI recognition. The SSI parsing restricts the SSIRS of the input SSI to match only with the same type of SSI to stop false matching. The SSI parsing can help effectively in processing with dynamic time warping for continuous speech recognitions by the SIR-CODE only.

In this study, the parsing process has been implemented by using an intensity-level slicing process. Instead of highlighting a specific range of gray level in the SSI over other levels, the process keeps a specific range from the normalised maximum value to a certain lower level of gray level in an SSI. We have called this the general algorithm for segmentation of the SSI. In this algorithm, the pixels that are less than the lower level are replaced by zeros. We call this new image as the SBSSI. The remaining pixels represent the highest PSD formants in an SSI. The SBSSI allows us to measure the properties of regions in the SSI. Indeed, the general algorithm segmentation of the SSI is similar to the process of getting rid of the noise in an SSI which has been discussed earlier in Chapter 3.

The general algorithm for segmentation has been used to classify the SSI into SSIFS and SSIRS regions rather than finding the precise start and end points of them. The general algorithm for segmentation is integrated either with spatial based matching or frequency based matching (using the MACH filter) to improve the SSI recognition. This process only matches the same order between the compared SSIs.

The general algorithm for segmentation shows an increase in the performance of the NCC matching (spatial matching) that successfully recognises whole words of the lexicon. However, defining a reliable threshold for the NCC matching for the whole lexicon was difficult. The correlation requires that the template and image have the same scale and orientation. The correlation value deteriorates when the template is not identical to the image. So the scale dependence of correlation has to be overcome. The correlation dependence on scale has been overcome precisely in Section 3.12.3 for the words consisting only of the SSIFS pattern because it is easier to find the start-end points for these. Therefore, defining both the order of the SSIFS and SSIRS and their precise start-end points is the strategy for updating the reliable threshold of the lexicon. In other words, this means applying the SIR-CODE-L2 implicitly on SSI patterns, which adds more entropy for a precise SSI recognition.

As mentioned earlier in Sections 4.6 and 3.8, nearly all lexical words have an SSIFS. The SSIFS in an SSI has a unique pattern which is the fingerprint of any word. Therefore, we may combine the results of individual matching of the SSIFS and SSIRS for the compared SSIs. This extra process can be achieved with less computations to give high recognition performance. In this study, expensive computations and transformations for achieving robust scaling in the SSI have been avoided for real-time application purposes.

The SSI segmentations are used to review one of the well-known and reliable speech databases, the TIMIT. We have found errors on defining the start and end points of some words in this database by using the SSI patterns matching. The study illustrates some examples of TIMIT's errors in defining the start and end points. This also provides evidence that the SIR-CODE segments are more powerful in recognising uttered words.

With this acceptable recognition ability of the SSIs, by using the general algorithm and spatial matching, we integrate the SSIs recogniser with traditional speech recognition (HMM). Indeed, continuous speech can be represented as a chain of phones with ambiguous start-end points of words. We comparing methods of dynamic time warping to predict the phone sequences for a recognised word to overcome the start-end point ambiguities. The problem around speaker variation is dealt by representing acoustic speech of a word within trained statistical models for different speakers. Acoustic speech models of lexicon words should be quantifiable enough to cover a wide range of speaking styles. Therefore, researchers have tried to fill the gap between actual speech and acoustic speech representations. One of the good features of the HMM is that it can incorporate a stochastic model into other stochastic models from several hierarchical knowledge sources. Thus, it can compare acoustic probability of unknown utterances generated by each word's model with trained known utterances. However, the problem of parameter estimation with these models is that they are very sensitive to background noise. Due to the HMM's limitations and the influence of background noise, the models (i.e. imperfect statistical mapping) suffer from overfitting, which appears as random errors or noise instead of the underling relationship. In case of any error in dynamic correlation of the chain of phones for a word, the accuracy of recognising the word is affected adversely. Further, this also affects the higher levels (the sentence level) in ASR.

In Chapter 5, we have suggested a realization layer on top of the traditional speech recognition layer (based on HMM) to check all sequential phones of a word in one go matching. The role of the realisation layer is to call the word according to the maximum likelihood of the HMM to match with the input word. Therefore, this results in either confirming the HMM result, suggesting successful recognition, or the input word will go through the SSI recogniser to select the right identifier of the input word. The recognition test of the 533 words of the study lexicon was successful. This integration of the two types of speech recognisers i.e. wave speech signal analysis and SSI matching, exploits the underlying information in SSI patterns to overcome the shortcoming of representing speech signals in acoustic models perfectly. Consequently, the ASR performance will be more accurate, and reliable in noisy environments which is still the biggest challenge in the application of ASR. We have tested a limited lexicon capacity of words with the SSI recogniser, which is sufficient to test for all the possibilities of the SSI patterns by the SSI recogniser. So, we believe that this test will be a foregone conclusion for a large lexicon too.

This study is based on a new package of speech system recognition integrated with the HMM. This paves way for future work in the field, summarised in the next section.

6.2 Future work

While this thesis has demonstrated improved speaker independent word recognition, many opportunities for extending the scope of this thesis remain. This section highlights some of these directions:

6.2.1 Speech representation as an image

In our work, speech signals are represented in the TFR by the SAFP. The discrete WT spectrogram can also be very useful in the SSI recogniser as it is much better in displaying frequency resolution. Thus it is effective in recognising the stress and rhythm in a speech language. Both stress and rhythm have not been studied in this thesis, which are important elements to understand the structure of the sentence.

6.2.2 Adapting speech databases of a huge lexicon for SSI patterns

In our study, the SSI pattern speech representations are classified based on phonemes and tested with a small lexicon by SSI pattern matching. For more reliability, it is important to expand the test of classifying the SSI pattern speech representations to a bigger lexicon by the SSI pattern recogniser.

Our study elaborates extensively on speech representations as SSI patterns. The mapping of English phonemes in to five SSI patterns have been discussed in this study. However, we are yet to explore the causes of the nested patterns. The nested patterns could consist of either SSIFS or SSIRS, or both. This creates an ambiguity as to which phonemes create what pattern. This can be achieved by incorporating a human interface to classify the nested patterns and resolve the ambiguity in these patterns.

This information provided by expert spectrogram readers will help to adapt the traditional database to our classification of speech signals in the SSI patterns. Thus, this database will comprise the precise start-end points of all the SSI patterns for a huge lexicon of the English language.

6.2.3 The SIR-CODE-L1-2 and L1-3 recogniser

Using SIR-CODE_L1-1, we classified patterns of any word as SSIFS and SSIRS. At SIR-CODE_L1-2, the SSIFS is further divided into two patterns, the FFP and SFFP. Similarly, the SSIRS can be divided into LAFP and SAFP and the fifth pattern is G (silent duration). We have demonstrated the classification of the SSIFS into an FFP and SFFP in the study. We did not extend this to classify the SSIRS patterns, which, can be achieved by the same strategy used to classify the SSIFS patterns. The entropy in the SIR-CODE_L1-2 will enhance the speech recognition in terms of word level and words in a sentence structure for the ASR system.

The next important stage is to define the start-end points of all the SSI patterns precisely to reach SIR-CODE_L1-3 which can then provide a highly precise recognition of a speech signal at word level and sentence structure level (e.g. mora), and also provide statistical information (duration, frequency, etc.) of the L1-3 patterns.

6.2.4 Continuous speech recognition by SIR-CODE

Our study used isolated words for SSI pattern speech representation for word level recognition. We need to implement this process for continuous speech. For this, we would require statistical information to define the average window slide length to capture the SSI patterns, through either the STFT or discrete WT representation. This will encourage wider applications of the SIR-CODE. More research is needed to integrate the SIR-CODE with the traditional techniques of speech recognition to enhance the existing methods that are struggling with the issue of speech recognition in a noisy environment.

Reference

7 Reference

[1] R. Deller, H. L. Hansen, and G. Proakis, *Discrite Time Processing of Speech Signals*. New York: IEEE Press Classic Reissue, 2000.

[2] K. H. Davis, R. Biddulph, and S. Balashek, "Automatic Recognition of Spoken Digits," *The Journal of the Acoustical Society of America,* vol. 24, pp. 637-642, 1952.

[3] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*: Prentice Hall, 1993.

[4] T. Sakai, "Automatic Mapping of Acoustic Features into Phonemic Labels," in *Spoken Language Generation and Understanding: Proceedings of the NATO Advanced Study Institute held at Bonas, France, June 26 – July 7, 1979*, J. C. Simon, Ed., ed Dordrecht: Springer Netherlands, 1980, pp. 147-189.

[5] L. R. Rabiner, "Speaker-Independent Recognition of Isolated Words Using Clustering Techniques," *IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING,* pp. 336-349, 1979.

[6] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *PROCEEDINGS OF THE IEEE,* vol. 77, pp. 257-286, 1989.

[7] Y. Steve, "A review of large-vocabulary continuous-speech," *IEEE Signal Processing Magazine,* vol. 13, p. 45, 1996.

[8] C. Wheddon and R. Linggard, *Speech and Language Processing*. Suffolk: Chapman and Hall, 1990.

[9] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 28, pp. 357-366, 1980.

[10] H. Hermansky, S. Tibrewala, and M. Pavel, "Towards ASR on partially corrupted speech," in *Proc.ntl. Conf. on Speech and Language Processing*, Tunisia, 1996.

[11] J. M. Borst, "The Use of Spectrograms for Speech Analysis and Synthesis," vol. 4, 1956.

[12] D. B. Pisoni, H. C. Nusbaum, P. A. Luce, and L. M. Slowiaczek, "Speech perception, word recognition and the structure of the lexicon," *Speech Communication,* vol. 4, pp. 75-95, 1985/08/01 1985.

[13] J. Holmes and W. Holmes, *Speech Synthesis and Recognition*. London: Taylor & Francis, 2001.

[14] B. Pinkowski, "Multiscale fourier descriptors for classifying semivowels in spectrograms," *Pattern Recognition,* vol. 30, p. 9, 1993.

[15] B. Pinkowski, "Principal component analysis of speech spectrogram images," *Pattern Recognition,* vol. 30, pp. 777-787, 1997.

[16] M. Dai, P. Baylou, and M. Najim, "An efficient algorithm for computation of shape moments from run-length codes or chain codes," London, 1997, pp. 57-69.

[17] K. FUKUNAGA and W. L. G. KOONTZ, "Application of the Karhunen-Loeve Expansion to Feature Selection and Ordering," *IEEE TRANSACTIONS ON COMPUTERS,,* vol. C-19, pp. 311-318, 1970.

[18] R. Steinberg and D. O'Shaughnessy, "Segmentation of a speech spectrogram using mathematical morphology," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 2008, pp. 1637-1640.

[19] J. P. Haton, *Automatic Speech Analysis and Recognition: Proceedings of the NATO Advanced Study Institute held at Bonas, France, June 29–July 10, 1981*: Springer Netherlands, 2012.

[20] D. Klatt and K. N. Stevens, "On the automatic recognition of continuous speech:Implications from a spectrogram-reading experiment," *Audio and Electroacoustics, IEEE Transactions on,* vol. 21, pp. 210-217, 1973.

[21] L. Cohen, *Time-frequency analysis: theory and applications*: Prentice-Hall, Inc., 1995.

[22] D. Gabor, "Theory of communication. Part 1: The analysis of information," *Electrical Engineers - Part III: Radio and Communication Engineering, Journal of the Institution of,* vol. 93, pp. 429-441, 1946.

[23] L. Cohen, "Time-frequency distributions-a review," *PROCEEDINGS OF THE IEEE,* vol. 77, pp. 941-981, 1989.

[24] F. Hlawatsch and G. F. Boudreaux-bartels, "Linear and quadratic time-frequency signal representations," *Signal Processing Magazine, IEEE,* vol. 9, pp. 21-67, 1992.

[25] O. Rioul and M. Vetterli, "Wavelets and signal processing," *Signal Processing Magazine, IEEE,* vol. 8, pp. 14-38, 1991.

[26] Y. Sheng, "Wavelet Transform," in *Transforms and Applications Handbook*, A. D. Poularikas, Ed., Second ed: CRC Press LLC, 2010.

[27] L. Cohen, "What is a multicomponent signal?," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, 1992, pp. 113-116 vol.5.

[28] R. K. H. K. D. L. Y. Lacy, "The sound spectrograph," *TIlE JOURNAL OF THE ACOUSTICAL SOCIETY OF AMERICA,* vol. 18, p. 31, 1946.

[29] L. Cohen and T. E. Posch, "Positive time-frequency distribution functions," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 33, pp. 31-38, 1985.

[30] L. Cohen, "On a fundamental property of the Wigner distribution," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 35, pp. 559-561, 1987.

[31] L. Cohen, "The uncertainty principle in signal analysis," in *Time-Frequency and Time-Scale Analysis, 1994., Proceedings of the IEEE-SP International Symposium on*, 1994, pp. 182-185.

[32] X. Jun and P. Flandrin, "Multitaper Time-Frequency Reassignment for Nonstationary Spectrum Estimation and Chirp Enhancement," *Signal Processing, IEEE Transactions on,* vol. 55, pp. 2851-2860, 2007.

[33] S. Qian and C. Dapang, "Joint time-frequency analysis," *Signal Processing Magazine, IEEE,* vol. 16, pp. 52-67, 1999.

[34] E. Wigner, "On the Quantum Correction For Thermodynamic Equilibrium," *Physical Review,* vol. 40, pp. 749-759, 1932.

[35] L. Cohen, "Generalized Phase-Space Distribution Functions," *Journal of Mathematical Physics,* vol. 7, pp. 781-786, 1966.

[36] T. A. C. M. C. W. F. G. Mecklenbrauker, "The Wigner distribution-a tool for time-frequency signal analysis; part Ill: relations with other time-frequency signal transformations," *Philips Journal of Research,* vol. 35, p. 17, 1980.

[37] R. L. Allen and D. Mills, *Signal Analysis: Time, Frequency, Scale, and Structure*: Wiley, 2004.

[38] C. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 24, pp. 320-327, 1976.

[39] S. Stein, "Algorithms for ambiguity function processing," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 29, pp. 588-599, 1981.

[40] H. I. Choi and W. J. Williams, "Improved time-frequency representation of multicomponent signals using exponential kernels," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 37, pp. 862-871, 1989.

[41] Y. Zhao, L. E. Atlas, and R. J. Marks, II, "The use of cone-shaped kernels for generalized time-frequency representations of nonstationary signals," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 38, pp. 1084-1091, 1990.

[42] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *Information Theory, IEEE Transactions on,* vol. 36, pp. 961-1005, 1990.

[43] J. Ning and J. Peng, "Repression of the Cross-Term Interference Based on Emd and Cohen's Class Distribution," in *Testing and Diagnosis, 2009. ICTD 2009. IEEE Circuits and Systems International Conference on*, 2009, pp. 1-4.

[44] B. S. E. A. Skrondal, *The Cambridge Dictionary of Statistics*, Fourth ed. United Kingdom at the University Press of Cambridge, 2010.

[45] F. Peyrin and R. Prost, "A unified definition for the discrete-time, discrete-frequency, and discrete-time/Frequency Wigner distributions," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 34, pp. 858-867, 1986.

[46] W. Martin and P. Flandrin, "Wigner-Ville spectral analysis of nonstationary processes," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 33, pp. 1461-1470, 1985.

[47] D. Dragoman, "Applications of the Wigner Distribution Function in Signal Processing," *EURASIP Journal on Advances in Signal Processing,* vol. 2005, p. 264967, 2005.

[48] L. R. Rabiner and R. W. Shafer, *Digital processing of speech signals* vol. null, 1978.

[49] J. B. A. L. R. RABINER, "A unified approach to short-time Fourier analysis and synthesis," *PROCEEDINGS OF THE IEEE,* vol. 65, p. 7, NOVEMBER 1977

[50] J. L. Flanagan, *Speech analysis; synthesis and perception, by James L. Flanagan*. Berlin, New York: Springer-Verlag, 1972.

[51] A. Janssen, "A note on "Positive time-frequency distributions"," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 35, pp. 701-703, 1987.

[52] A. V. Oppenheim, "Speech spectrograms using the fast Fourier transform," *Spectrum, IEEE,* vol. 7, pp. 57-62, 1970.

[53] M. R. Portnoff, "Time-frequency representation of digital signals and systems based on short-time Fourier analysis," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 28, pp. 55-69, 1980.

[54] B. E. A. Saleh and N. Subotic, "Time-variant filtering of signals in the mixed time frequency domain," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 33, pp. 1479-1485, 1985.

[55] J. B. Allen, "Short term spectral analysis, synthesis, and modification by discrete Fourier transform," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 25, pp. 235-238, 1977.

[56] M. J. Bastiaans, "A Sampling Theorem For The Complex Spectrogram, And Gabor's Expansion Of A Signal In Gaussian Elementary Signals," *Optical Engineering,* vol. 20, pp. 204597-204597-, 1981.

[57] J. Wexler and S. Raz, "Discrete Gabor expansions," *Signal Processing,* vol. 21, pp. 207-220, 1990.

[58] S. Qian and D. Chen, "Discrete Gabor transform," *Signal Processing, IEEE Transactions on,* vol. 41, pp. 2429-2438, 1993.

[59] L. Shidong and D. M. Healy, Jr., "A parametric class of discrete Gabor expansions," *Signal Processing, IEEE Transactions on,* vol. 44, pp. 201-211, 1996.

[60] A. Akan, V. Shakhmurov, and Y. Cekic, "A fractional Gabor transform," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on,* 2001, pp. 3529-3532 vol.6.

[61] E. R. Caianiello and G. Chollet, *Speech processing, recognition, and artificial neural networks: proceedings of the 3rd International School on Neural Nets "Eduardo R. Caianiello"*: Springer, 1999.

[62] P. Ladefoged, *A Course in Phonetics* vol. II. New Youk: Harcourt Brace Jovanovich, 1975.

[63] J. C. Catford, *A Practical Introduction to Phonetics*: Oxford University Press, Incorporated, 2001.

[64] H. Bussmann, K. Kazzazi, and G. Trauth, *Routledge Dictionary of Language and Linguistics*: Taylor & Francis, 1998.

[65] C. E. Stilp, "The redundancy of phonemes in sentential context a)," *The Journal of the Acoustical Society of America,* vol. 130, pp. EL323-EL328, 2011.

[66] A. K. Halberstadt, "Heterogeneous acoustic measurements and multiple classifiers for speech recognition," Massachusetts Institute of Technology, 1999.

[67] E. Sejdi, I. Djurovi, and J. Jiang, "Time-frequency feature representation using energy concentration: An overview of recent advances," *Digit. Signal Process.,* vol. 19, pp. 153-183, 2009.

[68] M. Dickinson, C. Brew, and D. Meurers, *Language and Computers*: Wiley, 2012.

[69] V. W. Zue and R. A. Cole, "Experiments on spectrogram reading," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '79.,* 1979, pp. 116-119.

[70] Z. Xinhui, D. Garcia-Romero, R. Duraiswami, C. Espy-Wilson, and S. Shamma, "Linear versus mel frequency cepstral coefficients for speaker recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on,* 2011, pp. 559-564.

[71] Raphael Steinberg and Douglas O'Shaughnessy, "Segmentation of speech spectrogram using Mathematical Morphology," presented at the ICASSP, 2008.

[72] P. Khunarsal, C. Lursinsap, and T. Raicharoen, "Singing voice recognition based on matching of spectrogram pattern," in *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, 2009, pp. 1595-1599.

[73] N. S. Dey, R. Mohanty, and K. L. Chugh, "Speech and Speaker Recognition System Using Artificial Neural Networks and Hidden Markov Model," in *Communication Systems and Network Technologies (CSNT), 2012 International Conference on*, 2012, pp. 311-315.

[74] K. Kirchhoff, "Syllable-level desynchronisation of phonetic features for speech recognition," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, 1996, pp. 2274-2276 vol.4.

[75] S. King, T. Stephenson, S. Isard, P. Taylor, and A. Strachan, "SPEECH RECOGNITION VIA PHONETICALLY FEATURED SYLLABLES," presented at the 5th International Conference on Spoken Language Processing (ICSLP '98), 1998.

[76] R. Rasipuram, M. Razavi, and M. Magimai-Doss, "Integrated pronunciation learning for automatic speech recognition using probabilistic lexical modeling," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, 2015, pp. 5176-5180.

[77] Z. Horváth, D. Csuka, K. Vargova, A. Kovács, A. T. Molnár, P. Gulácsi-Bárdos, S. Leé, L. Varga, R. G. Kiss, I. Préda, and G. Füst, "Elevated C1rC1sC1inh levels independently predict atherosclerotic coronary heart disease," *Molecular Immunology*, vol. 54, pp. 8-13, 2013.

[78] P. Ladefoged and K. Johnson, *A Course in Phonetics*: Cengage Learning, 2014.

[79] K. Hayward, *Experimental Phonetics: An Introduction*: Taylor & Francis, 2014.

[80] A. Marchal, *From Speech Physiology to Linguistic Phonetics*: Wiley, 2010.

[81] A. Mohammed, Y. Rupert, and C. Chris, " Defining properties of speech spectrogram images to allow effective pre-processing prior to pattern recognition," *SPIE*, vol. 8748:Optical Pattern Recognition XXIV, p. 11, 29 April 2013.

[82] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*: Prentice-Hall, Inc., 2006.

[83] A. Mallawaarachchi, S. H. Ong, M. Chitre, and E. Taylor, "Spectrogram denoising and automated extraction of the fundamental frequency variation of dolphin whistles," *The Journal of the Acoustical Society of America*, vol. 124, pp. 1159-70, Aug 2008.

[84] W. B. Hussein, "Spectrogram Enhancement By Edge Detection Approach Applied To Bioacoustics Calls Classification," *Signal & Image Processing : An International Journal*, vol. 3, pp. 1-20, 2012.

[85] B. E. D. Kingsbury, N. Morgan, and S. Greenberg, "Robust speech recognition using the modulation spectrogram," *Speech Communication*, p. 15, 1998.

[86] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue. (1993). *TIMIT Acoustic-Phonetic Continuous Speech Corpus*. Available: http://hdl.handle.net/11272/2PJSR Linguistic Data Consortium

[87] F. King-Sun and B. K. Bhargava, "Tree Systems for Syntactic Pattern Recognition," *IEEE Transactions on Computers*, vol. C-22, pp. 1087-1099, 1973.

[88] P. H. Hennings-Yeomans, B. V. K. V. Kumar, and M. Savvides, "Palmprint Classification Using Multiple Advanced Correlation Filters and Palm-Specific Segmentation," *Information Forensics and Security, IEEE Transactions on,* vol. 2, pp. 613-622, 2007.

[89] H. Zhou and T. H. Chao. Bellingham, Wash.: SPIE - The International Society for Optical Engineering, 1999.

[90] J. Saunders, "Real-time discrimination of broadcast speech/music," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, 1996, pp. 993-996 vol. 2.

[91] D. Crystal, *Dictionary of Linguistics and Phonetics*: Wiley, 2011.

[92] E. McDermott, T. J. Hazen, J. Le Roux, A. Nakamura, and S. Katagiri, "Discriminative Training for Large-Vocabulary Speech Recognition Using Minimum Classification Error," *Audio, Speech, and Language Processing, IEEE Transactions on,* vol. 15, pp. 203-223, 2007.

[93] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco, "Connectionist probability estimators in HMM speech recognition," *Speech and Audio Processing, IEEE Transactions on,* vol. 2, pp. 161-174, 1994.

[94] S. M. Aji and R. J. McEliece, "The generalized distributive law," *Information Theory, IEEE Transactions on,* vol. 46, pp. 325-343, 2000.

[95] Z. Ghahramani, "An introduction to hidden Markov models and Bayesian networks," in *Hidden Markov models,* ed: World Scientific Publishing Co., Inc., 2002, pp. 9-42.

[96] R. Bakis, "Continuous speech word recognition via centisecond acoustic states," *IEE Proceedings,* vol. 64, pp. 532-536, April 1975.

[97] T. Slobada and A. Waibel, "Dictionary learning for spontaneous speech recognition," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, 1996, pp. 2328-2331 vol.4.

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

[99] C. Taemin, K. KiBeom, and J. P. Bello, "A Minimum Frame Error Criterion for Hidden Markov Model Training," in *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, 2012, pp. 363-368.

[100] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, *The HTK Book*. Cambridge , UK: Cambridge University Engineering Department, 2006.

[101] M. J. F. Gales, "The HMM error model," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on,* 2002, pp. I-937-I-940.

[102] X. Xiong, L. Jinyu, C. Eng Siong, L. Haizhou, and L. Chin-Hui, "A study on hidden Markov model's generalization capability for speech recognition," in *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*, 2009, pp. 255-260.

[103] J. Biing-Hwang, H. Wu, and L. Chin-Hui, "Minimum classification error rate methods for speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 5, pp. 257-265, 1997.