Addis Ababa University
College of Natural Sciences

Audio-Visual Speech Recognition Using Lip Movement for Amharic Language

Befkadu Belete

A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

October, 2017

Addis Ababa University
College of Natural Sciences

Befkadu Belete
Adviser*: Yaregal Assabie (PhD)*

This is to certify that the thesis prepared by *Befkadu Belete,* titled: *Audio-Visual Speech Recognition Using Lip Movement for Amharic Language* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Name	Signature	Date
Advisor: *Yaregal Assabie (PhD)*		
Examiner: Dida Midekso (PhD)		
Examiner: *Solomon Atnafu (PhD*		

Abstract

Automatic Speech Recognition (ASR) is a technology that allows a computer to identify the words that a person speaks into a microphone or telephone and convert it to a written text. In recent years, there have been many advances in automatic speech reading system with the inclusion of visual speech features to improve recognition accuracy under noisy conditions. By identifying lip movements and characterizing their associations with speech sounds, the performance of speech recognition systems can be improved, particularly when operating in noisy environments.

The aim of this study is to design and develop automatic audio-visual Amharic speech recognition using lip reading. In this study, for face and mouth detection we use Viola-Jones object recognizer called haarcascade face detection and haarcascade mouth detection respectively, after the mouth detection ROI is extracted. Extracted ROI is used as an input for visual feature extraction. DWT is used for visual feature extraction and LDA is used to reduce visual feature vector. For audio feature extraction, we use MFCC. Integration of audio and visual features are done by decision fusion. As a result of this, we used three classifiers. The first one is the HMM classifier for audio only speech recognition, the second one is HMM classifier for visual only speech recognition and the third one is CHHM for audio- visual integration.

In this study, we used our own data corpus called AAVC. We evaluated our audio-visual recognition system with two different sets: speaker dependent and speaker independent. We used those two evaluation sets for both phone (vowels) and isolated word recognition. For speaker dependent dataset, we found an overall word recognition of 60.42% for visual only, 65.31% for audio only and 70.1 % for audio-visual. We also found an overall vowels (phone) recognition of 71.45% for visual only, 76.34% for audio only and 83.92 % for audio-visual speech. For speaker independent dataset, we got an overall word recognition of 61% for visual only, 63.54% for audio only and 67.08% for audio-visual. The overall vowel (phone) recognition on the speaker independent dataset is 68.04% for visual only, 71.96% for audio only and 76.79 % for audio-visual speech.

Keywords: Amharic; Lip-reading; visemes; appearance-based feature; DWT; AAVC

Dedication

To My Beloved Sister Belaynesh Belete

Acknowledgment

First of all, I thank God for being with me all the time, not only during this research, but also in my whole life. It is all His kindness that brings this happiness to my life.

I would like to express my very great appreciation to my adviser Dr. Yaregal Assabie for his valuable and constructive suggestions during the development of this research work. His willingness to give his time so generously has been very much appreciated.

Special thanks should be given to my beloved sister Belaynesh Belete, to my beloved brothers Minilik Belete and Tena Belete and my best friends Tsigereda Gizachew and Yaynshet Medhn for supporting and encouraging me to finish my thesis. I would also like to extend my thanks to my students at Addis Ababa Science and Technology University who participated in data gathering. Furthermore, gratitude goes to all friends and classmates who were encouraging me to complete my research work.

I would also like to extend my gratitude to Addis Ababa Science and Technology University for sponsoring my study. Since this thesis is the cumulative result of the two years of learning, I would like to thank all the staff members of the Department of Computer Science, Addis Ababa University involved in the process as well as my classmates for working together in different projects and assignments in harmony.

Finally, I wish to thank my brothers Tena Belete and Eingdawerk Eshete and all family members for their support and encouragement throughout my study.

Contents

List of Figures

List of Tables

List of Algorithm

Acronyms and Abbreviations

AAVC	Amharic Audio-Visual data corpus
ASRs	Automatic speech recognitions
AVRRSD	Audio-visual recognition ratio of speaker dependent
ARRSD	Audio Recognition ratio of speaker dependent
AV-ASR	Audio-visual automatic speech recognition
AVSR	Audio-visual speech recognition
AVSP	Audio-visual speech processing
CHMM	Coupled Hidden Markov Model
CV-syllables	Consonant-Vowel syllables
DBNs	Dynamic Bayesian Networks
HCI	Human computer interface
HMM	Hidden Markov Model
LPCs	Linear prediction coefficients
MFCCs	Mel-frequency cepstral coefficients
NN	Neural Networks
ROI	Region of interest
SD	Speakers Dependent
SNR	Signal to noise ratio
SI	Speakers Independent
VRRSD	Visual Recognition ratio of speaker dependent
VSR	Visual speech recognition

Chapter One: Introduction

1.1 Background

Automatic speech recognition (ASR) is a technology that allows a computer to identify the words that a person speaks into a microphone or telephone and convert it to a written text. It is viewed as an integral part of future human computer interfaces that are envisioned to use speech, among other means, to achieve natural, pervasive, and ubiquitous computing. However, although ASR has witnessed significant progress in well-defined applications like dictation and medium vocabulary transaction processing tasks in relatively controlled environments, its performance has yet to reach the level required for speech to become a truly pervasive user interface. Indeed, even in "clean" acoustic environments, state-of-the-art ASR system performance lags human speech perception by up to an order of magnitude, whereas its lack of robustness to channel and environment noise continues to be a major hindrance [1].

Today's trend is to make communication and interaction between humans and their artificial partners easier and more natural. Speech recognition technology has reached a maximum of performance and good recipes for building speech recognizers have been written. However, the major problems of background noise and reverberations due to the environment are still insurmountable. Therefore, inspecting other sources, other than sound, for complementary information which could alleviate these problems, is a necessity [2].

It is well known that both human speech production and perception are bimodal process in nature. Visual observation of the lips, teeth and tongue offers important information about the place of pronunciation articulation. A human listener can use visual cues, such as lip and tongue movements, to enhance the level of speech understanding. The process of using visual modality is often referred to as lip-reading which is to make sense of what someone is saying by watching the movement of his lips [3]. A Visual speech recognition (VSR) system refers to a system which utilizes the visual information of the movement of the speech articulators such as the *lips, teeth* and somehow *tongue* of the speaker. The advantages are that such a system is not sensitive to ambient noise and change in acoustic conditions, does not require the user to make a sound, and provides the user with a natural feel of speech and dexterity of the mouth [4].

Speech command based systems are useful as a natural interface for users to interact and control computers. Such systems provide more flexibility as compared to the conventional interfaces such as keyboard and mouse. However, most of these systems are based on audio signals and are sensitive to signal strength, ambient noise and acoustic conditions [4]. To overcome this limitation, speech data that is orthogonal to the audio signals such as visual speech information can be used. The systems that combine the audio and visual modalities to identify utterances are known as audio-visual speech recognition (AVSR) system.

There are two ways of approaching phonetics. One approach studies the physiological mechanisms of speech production. This is known as *articulatory phonetics*. The other, known as *acoustic phonetics*, is concerned with measuring and analyzing the physical properties of the sound waves we produce when we speak. According to articulatory phonetics, organs of articulation are divided into movable articulators and stationary articulators. Movable articulator is the articulator that does all or most of the moving during a speech gesture. The movable articulator is usually the lower lip, some part of the tongue and jaws. A stationary articulator is the articulator that makes little or no movement during a speech gesture. Stationary articulators include the upper lip, the upper teeth, the various parts of the upper surface of the oral cavity, and the back wall of the pharynx [5, 6]. Those articulators movement dose not affected by noise. Thus, visual speech information from the speaker's mouth region will be improve noise robustness of automatic speech recognizers

Speech in most languages is produced by the lungs forcing air through the vocal chords located in the larynx. The vocal chords are two muscular folds that are usually apart for breathing, but can be brought close together and then vibrate in the airstream from the lungs. The vibration is controlled by the tension of the chords and modulates the airstream. This is the process of phonation, and the sounds produced are voiced. Sounds made using an unrestricted, unmodulated airstream are unvoiced [7]. Figure 1.1 shows a simplified diagram of the speech articulators [7].

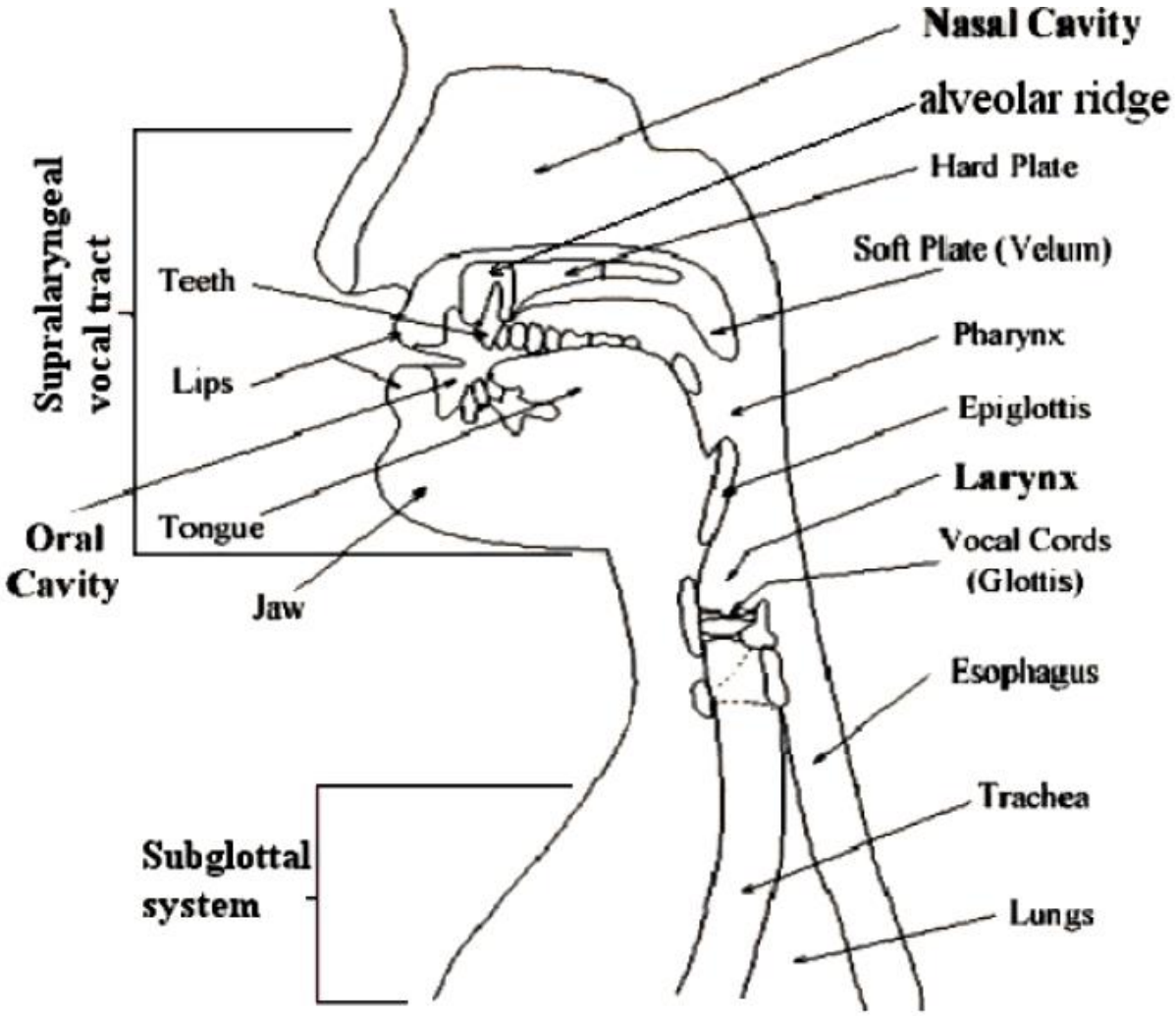


Figure 1. 1: *Principal Features of the Vocal Tract*

Above the larynx is the vocal tract, the first stage of which is the pharynx (back of the throat) which can be tightened to change speech sounds. From the pharynx, the airflow may be redirected either into the nose and mouth, or just the mouth by closing the velum (soft palate). Sounds made with the velum open are nasal and with the velum closed oral. The shape and configuration of the vocal tract further filter the speech sound. As shown in Figure 1.1 [7], the sounds produced can be classified according to the place and manner of their articulation.

The place of articulation describes which articulators are used, and is classified as one of the following [7].

- **Bilabial** between both lips. For example, ᑎ/Ba.

- **Labiodental** between lower lip and upper front teeth. For example, ፈ/Fe.
- **Alveolar** between the tongue tip or blade and alveolar ridge. For example, ተ/Te.
- **Palatal** between the front of the tongue and the hard palate. For example, ዩ/U.
- **Velar** back of the tongue and the soft palate. For example ግ/g.
- **Palato-alveolar** tongue blade and back of the alveolar ridge. For example, ሸ/ She.
- **Labiovelar constants are** doubly articulated at the velum and the lips. For Example, ጓ/g^w.
- **Glottal consonants** are consonants using the glottis as their primary articulation. For example, ሀ/h.

1.2 Motivation

Recently, researches on automatic lip-reading using the video sequence of the speaker's mouth have attracted significant interest. Automatic lip-reading under noisy environments is very effective in compensation for the decrease of speech recognition rate with an audio-only speech recognition (ASR) system. The bimodal based on audio-visual information is an important part of the human-computer interface (HCI). We allow more weighting value to visual data than to audio one under a bad SNR (signal to noise ratio) but, on the contrary, more to audio data than to visual one under a clean SNR (signal to noise ratio) [8]. Under noisy circumstances, this bimodal approach has been a good alternative showing superior recognition rate to audio-only ASR system.

In general, the nature of human speech is bimodal. Speech observed by a person depends on audio features, as well as on visual features like lip synchronization or facial expressions. Visual features of speech can compensate for a possible loss in acoustic features of speech due to noisy environments. This combination of auditory and visual speech recognition is more accurate than audio only or visual only features. Perception of speech can be enhanced with use of multiple information sources like audio and video features of speech [9].

The above facts have motivated us to work on automatic recognition of visual speech, formally known as automatic lip-reading, or speech-reading. Work in this field aims at improving ASR by exploiting the visual modality of the speaker's mouth region in addition to the traditional audio modality, leading to audio-visual automatic speech recognition (AVASR) systems.

1.3 Statement of the Problem

Audio features are still the main contribution and play a more important role, than visual features. However, in some cases, it is difficult to extract useful information from the audio. There are many applications in which it is necessary to recognize speech under extremely adverse acoustic environments. Detecting a person's speech from a distance or through a glass window, understanding a person speaking among a very noisy crowd of people. In these applications, the performance of traditional speech recognition is very limited.

To our best of knowledge, few researches have been attempted to develop speech recognition for Amharic language. Solomon Berhanu [10] developed an isolated consonant vowel syllable Amharic recognition system followed by Kinfe Tadesse [11] who developed a sub-word based (phones, triphones, and CV-syllables) isolated Amharic word recognition systems. Martha Yifru [12] also developed a speaker independent, small vocabulary, isolated Amharic word recognizer to be used as an interface to command Microsoft word. Molalgne Girmaw [13] also developed an HMM-based small vocabulary speaker dependent continuous speech recognizers using two different tools: HTK and the MSSTATE Toolkits and compared their performance. Solomon Teferra [14] is the other researcher in this area who explored various possibilities for developing a large vocabulary speaker independent continuous speech recognition system for Amharic. Solomon Teferra [14] reported that he proved Amharic syllables models can be used as a competitive alternative to triphone models. He also incorporated the five dialects of Amharic language (Gonder, Gojam, Shewa, Menz and Wello) and developed a speech corpus**.** However, all researchers considered only audio in their work. Thus, performance of these systems (i.e., audio only speech recognition) is heavily dependent on a match between training and test conditions. The performance of automatic speech recognition (ASR) systems degrades heavily in the presence of noise, compromising their use in real world scenarios.

Lots of researches have been done on audio-visual speech recognition for different languages to enhance the performance of the speech recognition system such as English [3], Japanese [15], Dutch [8], etc. However, to our best knowledge, research on audio-visual speech recognition using lip-reading has not been carried out for Amharic language or for any other Ethiopian local language.

1.4 Objectives

General Objective

The general objective of this thesis is to develop audio-visual speech recognition by using lip reading for Amharic language.

Specific Objectives

To achieve the general objective of the thesis, the following specific objectives are identified.

- Study the phonological and viseme structure of Amharic Language.
- Collect audio-visual data for training and testing.
- Extract audio and visual speech features.
- Design or adopt a system that recognizes Amharic speech from audio and visual features.
- Design visual speech recognition for Amharic.
- Design or adopt audio speech recognition for Amharic.
- Integrate audio and visual speech recognitions.
- Develop a prototype of the system.
- Train and test the system.

1.5 Methodology

In order to achieve the general and specific objectives mentioned above, we will use the following methods.

Literature Review

Exhaustive literature review will be carried out by examining the different types of approaches, methods and tools used in implementing the visual-based (lip-reading) speech recognition system. Amharic language related literatures will also be reviewed to better understand the phonetic and viseme features of the language.

Data Collection

Relevant data which is needed to conduct the research will be collected. Various data collection strategies will be followed to acquire the required data. The dataset useful for

training and evaluating the system. Video data will be recorded by web camera in a typical office environment. All participant in the data collection can speak Amharic language. Participants were asked to read the word in their own natural style and with no instruction regarding pronunciation

Design and Implementation

To accomplish the study, Python programming language will be used. Python is selected because it supports Unicode encoding, and Python has a free Python libraries package, for constructing matrices and for all other operations associated with matrix, and also Python has free open libraries packages for audio and visual processing. We will also use OpenCV for visual processing because we can easily integrate this package with Python.

Evaluation

In order to evaluate the performance of the system, a prototype system will be developed for the Amharic audio-visual speech recognizer that can recognize independent speakers. We will select and set different types of evaluation techniques to test the recognition performance of the proposed system.

1.6 Scope and Limitations

This study, focuses on developing audio-visual speech recognition system for Amharic language using lip-reading by integrating audio and visual features. In this thesis, we focus on isolated words and phones of the Amharic language. The system does not consider continuous speech, facial expressions, hand and body gestures. Furthermore, the system does not consider side-view mouth visual features.

1.7 Application of Results

Lip-reading can be seen both as a complementary process to speech recognition and as a stand -alone process. The application for lip reading as a stand-alone application are diverse: multimedia phones for the hearing impaired, mobile phone models interface spaces (e.g., at the time of this writing , phone models that use lip-reading are already being designed) person identification, recovery of speech from deteriorated or mute movie clips, and perhaps the most promoted applications, security by video surveillance (e.g., security cameras that are

recovering what is being said without the need of a microphone, and more importantly for a large distance) and used to improve sign-language recognition.

1.8 Organization of the Rest of the Thesis

The rest of this thesis work is organized as follows: Chapter 2 gives an overview of AVSR systems and reviews existing approaches described in the literature. Chapter 3 discusses the related works on automatic audio only speech recognition, visual only speech recognition and audio-visual speech recognition. Chapter 4 presents the design of the proposed system along with the functions of each of the components. Chapter 5 elaborates the implementation and evaluation of the proposed system. Chapter 6 concludes the thesis by highlighting the contribution of the research, recommendations, and future works.

Chapter Two: Literature Review

2.1 Introduction

This chapter details the background and the terminology used in audio-visual speech recognition system. Section 2.2 discusses the basic of phonetics and phonology. In Section 2.3, the Amharic phonetics, phonology and phonemes are discussed. Section 2.4 discusses lip-reading or speech reading. Section 2.5 discusses phonetics of visual speech. Section 2.6 discusses the basic architecture of AVSR system. Section 2.7 discusses audio feature extraction. Section 2.8 discusses audio the basics of visual front-end and visual features extraction. Section 2.9 discusses classifiers for audio-visual speech recognition. Section 2.10 discusses multimodal fusion. This Chapter also talks about the basics of HMMs and the different types of classifications of HMMs and types of speech recognition. In Section 2.11 we discuss the type of speech recognition.

2.2 Phonetics and Phonology

The human perception of the world is inherently multi-sensory since the information provided is multimodal. In addition to the auditory information, there is visual speech information provided by the facial movements as a result of moving the articulators during speech production [16, 17]. The use of visual speech information has introduced new challenges in the field of ASR. These are robust face and mouth detection, extraction and tracking of a visual region of interest (ROI), extraction of informative visual features from the ROI, the integration of audio and visual modalities and the provision of suitable classifiers [18].

In order to understand the link between the audio signal and the corresponding visual signal that can be detected on the mouth/lips of the speaker, we need to have some understanding of how speech is produced. Phonetics and phonology are the two fields of grammar which deal with the study of the sounds of human language. Phonetics studies the actual speech sounds of the language including the way how the sound is produced, transmitted, and perceived. Phonology on the other hand is the systematic study of how speech sounds are organized to form sound systems. Phonetics is related to the science of acoustics in that it uses much of the techniques used by acoustics in the analysis of sound [7].

We know from phonetics and acoustics that according to their acoustic features, speech sounds can be categorized into consonants and vowels. These are subcategorized into the smallest units of a language called phones. Phones are one of the most common sub-word recognition units used in the development of automatic speech recognizers [19]. There are three sub-disciplines of phonetics that study the different feature of speech sounds [19]. These are:

- **Articulatory phonetics:** The study of the production of speech sounds. It is the study of articulators in the process of the production of speech sound. The muscles change the shape of the articulators enabling them to modify the flow of air that passes from the chest through the mouth and nostrils into the atmosphere.
- **Acoustics phonetics:** The study and analysis of the physical production and transmission of speech sounds. Speech sounds, like sounds in general, are transmitted through the air as small, rapid variations in air pressure that speared in longitudinal waves from the speaker‘s mouth and can be heard recorded, visualized and measured. Differences between individual speech sounds are directly reflected as differences in either one or several or all of the sound parameters like tone, stress duration, pitch, loudness and quality of the speech waves. By dealing with the study and description of the acoustics properties of individual speech sounds, acoustic phonetics is the immediate like between articulatory phonetics and speech perception. It is important for applications in the fields of signals processing and speech technology, like ASRS.
- **Auditory phonetics:** The study of the perception of sounds. Just as articulatory phonetics involves the understanding of the anatomy of the human speaking system, auditory phonetics involves the understanding of the human hearing system. This means, auditory phonetics deals with the understanding of the anatomy and physiology of the human ear and brain.

2.3 Amharic Phonetics

Articulatory phonetics shows that characteristics of sound are determined by the positions of the various articulators in the vocal tract and the state of the vocal cords. Three aspects could be mentioned here: voicing, manner and place of articulation. According to the voicing aspect, we can classify sounds into voiced and unvoiced (voiceless). In view of the manner of articulation, we can have the following classes of sound: stops, fricatives and approximants.

The places of articulation include labial, dental, palatal, velar and glottal as shown in Figure 1.1[6]. There are a set of 37 phones in Amharic language among which seven of them are vowels and thirty of them are consonants [20, 21]. The major categories of Amharic phones are described below.

2.3.1 Amharic Consonants

Table 2.1 shows the general classification of Amharic consonants based on their manner of articulation, voicing, and place of articulation [20, 21].

Table 2.1: *Amharic Consonants*

Manner of articulation	**Voicing**	**Place of Articulation**						
		Bilabial	Labiodental	Alveolar	Palatal	Velar	Labiovelar	Glottal
Stops	Voiced	ብ[b]		ድ[d]		ግ[g]	ጓ[g^{w}]	
	Voiceless	ፕ[p]		ት[t]		ክ[k]	ኳ[k^{w}]	ዕ [ʔ]
	Ejective or Glottalized	ጵ[p’]		ጥ[t’]		ቅ[k’]	ቋ[k’^{w}]	
Fricatives	Voiced		ቭ[v]	ዝ[z]	ዥ[ž]			
	Voiceless		ፍ[f]	ስ[s]	ሽ[š]			ህ[h], ኋ[h^{w}]
	Ejective			ጽ[s’]				
Affricates	Voiced				ጅ[j]			
	Voiceless				ች [c]			
	Ejective or Glottalized				ጭ [c’]			
Nasals		ም[m]		ን[n]	ኝ[N]			
Liquids				ል[l]	ር[r]			
Glides		ው[w]			ይ[y]			

2.3.2 Amharic Vowels

Vowels are open sounds, made largely by shaping the vocal tract rather than by interfering with the flow of air stream [19] .Vowels are most usefully described in terms of the *position of the tongue* as they are articulated [19]. A vowel articulated with the body of the tongue relatively forward is classified as a *front vowel*; one made with the body of the tongue relatively high is a *high vowel*. Vowels produced with the body of the tongue neither high nor low are called *mid vowels* [19]. Vowels produced with the tongue body front are called *front vowels* while those made with the tongue body back are called *back vowels*. Vowels accompanied by lip rounding as in (ኡ and ኦ) are called *rounded vowels* while the other vowels are called *unrounded vowels.*

In general, Amharic vowels (ኧ, ኡ, ኢ, ኣ, ኤ, እ and ኦ) are categorized in to rounded (ኡ and ኦ) and unrounded (ኧ, ኡ, ኢ, ኤ and እ) based on their manner of articulation. The other categorization is based on the height of the tongue and part of the tongue which depends on their production as indicated in Table 2.2[20, 21].

Table 2.2: *Amharic Vowels*

	Front/Unrounded	**Central/Unrounded**	**Back/Rounded**
High	ኢ [i]	እ[I]	ኡ[u]
Mid	ኤ [e]	ኧ[A]	ኦ[o]
Low		ኣ [a]	

2.4 Speech Reading

Speech reading, or as it is commonly referred to lip-reading or visual speech recognition (VSR), using visual cues obtained from the images of the mouth, lips, chin and any other related part of the face to understand speech. Speech reading is simply a special case of audio-visual speech recognition where all emphasis is placed on the visual, not the acoustic speech modality. A person skilled in speech reading is able to infer the meaning of spoken sentences by looking at the configuration and the motion of visible articulators of the speaker. Although sometimes referred to as lip-reading, speech information does not stem solely from labial

configurations as the tongue and teeth position also act as additional sources of information. It is, however, largely agreed that most information pertaining to visual speech does stem from the mouth region of interest (ROI) [15].

In addition to speech reading, there are many other ways in which humans can use their sight to assist in aural communication. Visible speech provides a supplemental information source that is useful when the listener has trouble comprehending the acoustic speech. Furthermore, listeners may also have trouble comprehending the acoustic speech in situations where they lack familiarity with the speaker, such as listening to a foreign language or an accented talker. When noisy environments are encountered, visual information can lead to significant improvement in recognition by humans. The complementing and supplemental nature of visual speech can be used in such speech processing applications as automated speech recognition, enhancement and coding under acoustically noisy conditions [15].

In general, lip-reading or visual information has been shown to be useful for improving the accuracy of speech recognition in both humans and machines. These improvements are the result of the complementary nature of the audio and visual modalities [3, 15]. For example, many sounds that are confusable by ear are easily distinguishable by eye, such as ‘ [n] and ‘ [m]. The improvements from adding the visual modality are often more pronounced in noisy conditions [15].

2.5 Phonetics of Visual Speech

Lip-reading is not as essential as hearing for most humans. However, for deaf people or when the environment is very noisy, it becomes important and possibly the only way for speech to be perceived or comprehended. It is also useful for security purposes such as in the case of surveillance when it is difficult to capture the voice of a person whose face is captured on a camera from a distance. Humans usually deal with the problem of noise by using visual information.

It is also known that some hearing impaired and deaf persons can reach almost perfect speech perception by only seeing the speaker’s face, or particularly the region of the mouth [2]. For everyone, visual information complements the audio signal not only when noise is present, but also in clean environments [3, 8, 9]. The reason why the visual modality is important is

that it offers complementary information about the place of articulation. This is because the articulators (lips, teeth, tongue) are visible. Seeing them can help distinguish for example ፕ[p] from ክ[k], ብ[b] from ድ[d] , ጵ[p’] from ጥ[t’] or ም[m] from ን[n], since all these pairs are easy to confuse from audio only because these pairs are the same manner of articulation and voicing as discussed in Section 2.3.1.

Verbal communication uses cues from both the visual and acoustic modalities to convey messages. Traditional information processing has usually focused on one media type. Human speech is produced by the vibration of the vocal cord and the configuration of the vocal tract that is composed of articulatory organs, including the nasal cavity, tongue, teeth, velum and lips. Using these articulatory organs, together with the muscles that generate facial expressions, a speaker produces speech. Since some of these articulators are visible, there is an inherent relationship between acoustic and visible speech. The bimodal nature of human speech can be most aptly demonstrated in the McGurk effect [3]. The McGurk effect demonstrates that when humans are presented with conflicting acoustic and visual stimuli, the perceived sound may not exist in either modality.

The aim of audio-visual speech processing (AVSP) is to take advantage of the redundancies that exist between the acoustic and visual properties of speech in order to process speech for recognition in an optimal manner. AVSP is a multidisciplinary field which requires skills in conventional speech processing, facial analysis, computer vision, human perception as well as the vast subject of image processing in order to capture facial artifacts and acoustic speech for use in processing. AVSP deals with the simultaneous analysis of corresponding speech and image information and their application to the field of speech processing [22].

The basic unit of acoustic speech is called the phoneme. Phonemes are the smallest segment of sound that convey useful linguistic information. Phonologically, each language is made of these basic units and each language or dialect consists of a set of phonemes [14, 23]. As discussed earlier in Section 2.3 for Amharic, consisting of 37 phonemes, is commonly used to classify phonemes [20, 21]. These basic audio sounds are used in most speech recognition systems to provide a set of basic units for recognition; these can then be combined to form the words and sentences using the additional information stored in the lexicon and language models. Similarly, in the visual domain, the basic unit of mouth movements is called a viseme

[2, 3]. Visemes are distinguishable segments obtained from videos of speakers. They represent particular oral or facial shapes, as well as the positions and movements adopted during speech utterances.

The human speech production system produces phonemes to construct a word and, by hearing these phonemes, we understand the spoken word. If we watch the lips at the same time, we can see the visemes to understand the spoken word.

They may coincide with the generation of one or more phonemes and are derived either manually by human observation of visual speech or automatically by the clustering of visual speech data. A number of phoneme-to-viseme mappings have been derived by researchers [24], but, unlike phonemes, there is no standard set of visemes for a given language [4]. Strictly speaking, instead of a still image, a viseme can be a sequence of several images that capture the movements of the mouth. However, most visemes can be approximated by stationary images [25]. Many acoustic sounds are visually ambiguous such that different phonemes can be grouped as the same viseme. Therefore there is a many to one mapping between phonemes and visemes. By the same token there are many visemes that are acoustically ambiguous. An example of this can be seen in the acoustic domain when people spell words on the phone, expressions such as '𑀩𑁆 [bi]' or '𑀤𑁆 [di]' are often used to clarify such acoustic confusion. These confusion sets in the auditory modality are usually distinguishable in the visual modality. This highlights the bimodal nature of speech and the fact that to properly understand what is being said information is required from both modalities. The extra information contained in the visual modality can be used to improve standard speech processing applications such as speech and speaker recognition. The bimodal nature of speech is highlighted especially well in the McGurk effect [2, 3]. For example when a person hears the sound 𑀧 [ba], but watches the sound 𑀖 [ga], the person may not perceive either 𑀧 [ba] or 𑀖 [ga]. Something close to a 𑀥 [da] is usually perceived. The McGurk effect highlights the requirement for both acoustic and visual cues in the perception of speech [15].

2.6 Front-end of Audio-Visual Speech Recognition

Before being applied to the recognizer for training or recognition purposes, audio and visual streams need to be preprocessed to remove data irrelevant to speech and to enhance certain characteristics that help to improve speech recognition performance. These preprocessing stages of the audio and video data are known as the audio front-end and visual front-end respectively. Front-end encompasses the preprocessing of the speech signal before the feature extraction phase, as well as the feature extraction itself. The design of the front-end, and particularly the feature extraction phase, plays an important role in maximizing the overall performance of a speech recognition system and is a core area of research in both audio-only and AVASR research. In the audio part of the front-end preprocessing, a number of techniques are available to enhance the speech signal and to reduce the effects of background and channel noise [26]. The design of video front-end is a rather more challenging task, as the video signal will contain substantial information about the speaker and background that are not relevant to the speech itself. This needs to be filtered out and a Region of Interest (ROI) around the mouth of the speaker defined and extracted [27, 28, 29], thereby greatly reducing both the dimensionality of the required feature vector and the computation cost of later processing. In comparison with the audio front-end, the visual front-end will also include the additional steps of speaker face and mouth detection and the extraction of a speech information region from the face of the speaker, collectively known as ROI extraction [27, 28]. The effects of variations in lighting conditions in both the spatial and temporal dimensions may also be addressed as part of the visual front-end, as well as distance and orientation normalization where relevant [29]. Audio and visual front-end processing are performed separately on the two streams and the extracted feature vectors integrated to form a single feature vector or used to train two separate recognizers depending upon the modality fusion approach adopted.

As the original audio and visual speech signals have high dimensionality, then, to use them directly for training and recognition, the classifier will need computational time and resources that are not commonly found even in modern computer systems [15]. Therefore, a more compact set of parameters representing the significant characteristics of speech are extracted from both the audio and video signals. The compact sets of parameters extracted from the two streams are generally referred to as the audio and video features respectively. The performance of a speech recognition system is greatly dependent on the extraction of features which are

robust, stable, and ideally retain all the speech information contained in the original source signal [15]. The main purpose of feature extraction is to capture speech information in a reasonably small number of dimensions [30].

In these cases, multi-modal systems incorporating other measures of the spoken words can significantly improve recognition. Acoustic features such as Mel-frequency Cepstral Coefficients (MFCCs) [30] and Linear Prediction Coefficients (LPCs) [31] represent the most commonly used acoustic features, additional research is ongoing in the field of noise robust acoustic features. After acoustic feature extraction, first and second derivatives of the data are usually concatenated with the original data to form the final feature vector. The original data is also known as the "static coefficients" while the first and second derivatives are also known as "delta" and "delta-delta" or "acceleration" coefficients [30].We do not cover the details of Automatic Audio Speech Recognition (AASR) systems as they have been covered extensively in previous literature [11, 12, 13, 32, 33], but as it is important for the understanding of the complete system of AVSR, a brief description is provided in Section 2.7.

2.7 Audio Feature Extraction

To help prepare the incoming audio signal for feature extraction stage, preprocessing techniques such as signal filtering and audio enhancement must be made in advance. Several results have been reported in the literature regarding the audio feature extraction techniques. Mel-frequency Cepstral Coefficients (MFCCs) [31] and Linear Prediction Coefficients (LPCs) [32, 33] represent the most commonly used audio features in the last few decades. There are still on-going research in the field of robust audio features and such features will not be considered in this work.

MFCCs is very popular and has been shown to outperform other feature extraction techniques as revealed in [31]. MFCCs [31] are derived from a Mel-frequency where this frequency axis is warped according to the Mel-scale, which approximate the human auditory system's response. The dynamic features which are first (delta- MFCCs) and second time-derivatives (delta-delta-MFCCs) of cepstral coefficients is now commonly employed to improve speech recognition performance [34, 35]. Feature extraction is a process that involves converting the speech wave captured either directly through microphones or from the already recorded speech data into a sequence of parameter vectors [36].

The recognition accuracy of speech recognition system heavily depends on feature extraction. This stage, therefore, requires due attention. The main goal of this step is to extract a parsimonious sequence of feature vectors from the input acoustic signal that have the maximum relevant information appropriate for the models used for classification [36]. The 18 few desirable properties that feature extraction should take into account during the development of ASR systems are [36]:

- The discrimination between sub-word classes to be high
- Speaker variability to be minimal
- To be robust enough to consider factors that affect speech recognition accuracy like channel and noise.

The cepstral coefficients and their temporal derivatives are a commonly used set of features for representing the local spectral properties of the signal.

Linear predictive coefficients (LPCs)

LPC is one of the most commonly used parametric modelling techniques in the speech recognition literature. In LPC analysis, it is assumed that the speech signal at any given time can be estimated from a linear combination of the speech samples in the past [32, 37]. If *s* (*n*) is the current speech signal, it can be estimated from its previous values *s* (*n*-1), *s* (*n*-2), *s* (*n*-3), …, *s* (*n*-*p*) as show in Equation 1.

$$s(n) = \sum_{j=1}^{p} a(j)s(n-j) + e(n) \quad (1)$$

Where *e* (*n*) is the error in the estimation of the current signal and the set of coefficients *a* (*j*) are the linear predictive coefficients. The number of predictive coefficients, *p*, is the number of previous samples used in the estimation. The predictive coefficients *a* (*j*) are computed by minimizing the mean-squared error between the predicted and the actual signal. The most frequently-used method to calculate the coefficients is autocorrelation, but covariance and lattice methods are also used [32, 37].

Mel Frequency Cepstral Coefficients (MFCCs)

It has been shown in psychophysical studies that humans do not perceive the variation in speech frequency on linear scale, but rather they are more sensitive to frequency variations

below 500Hz above this, the same degree of variation in pitch is perceived by an unequal increase in frequency. The interval over which a certain level of change in pitch is observed becomes greater as the frequency increases on an ordinary hertz scale. The Mel scale representation is based on this non-linear response of the human ear to pitch perception. A more even distribution of coefficients according to pitch sensitivity is produced by mapping the pitch variations on hertz scale to the Mel scale [33]. The relation between the hertz scale and the Mel scale is given by using Equation 2.

$$\text{Mel}(f_m) = 2595\log_{10}(1 + \frac{f}{700}) \quad (2)$$

Where f and f_m are the frequencies on hertz scale and Mel scale, respectively.

To obtain the MFCC values, the following procedure is normally followed [37]. The discrete Fourier transform (DFT) of the speech signal is taken over the frame duration and the power content of the resultant spectrum is mapped onto the Mel scale using triangular overlapping windows. The Fourier transform taken over a short duration such as the one above, is known as the short-time Fourier transform (STFT) [37]. The MFCC are calculated by taking the discrete cosine transform (DCT) of the logarithm of the power mapped on the Mel frequencies.

2.8 Visual Front-end

The visual front-end stage encodes stimuli coming from the visual cues (mainly the lips) of a speaker and transforms it into a suitable representation that is compatible with that of the recognition module. However, prior to this feature extraction process, a number of preprocessing steps have to be done as shown in Figure 2.1[25]. This involves face detection followed by ROI extraction. Then, the lips of the speaker are tracked in consecutive frames.

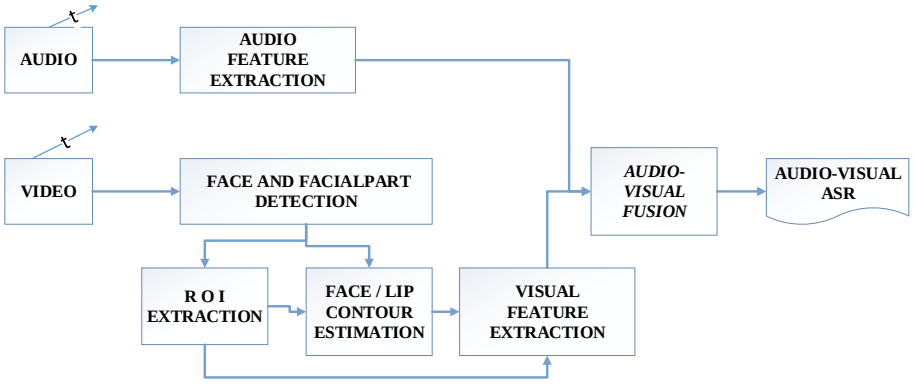


Figure 2.1: *Preprocessing Steps of Visual Front-end*

2.8.1 Visual Front-End Preprocessing

Before extracting any visual features, a number of preprocessing steps are required as described in Figure 2.1 [25]. The first step is usually face detection followed by Region of Interest (ROI) extraction. The ROI consists of the region of the face that contains most of the speech information. There is no unique understanding of where most of the speech information is located and hence there are many interpretations of what we mean by ROI [2, 25]. This issue will be discussed later. However, we can now establish that the ROI depends on the type of visual data being provided to the visual speech recognition system.

2.8.1.1 Face Localization

Face detection is an essential preprocessing step in many face related applications (e.g., face recognition, lip reading, age, gender, and race recognition). The performance of these applications depends on the reliability of the face detection step. Also face detection is an important research problem in its role as a challenging case of a more general problem (i.e., object detection) [38].

The face detection problem encompasses several related automated computational tasks including determining if an image does contain a human face? Where is it? How many faces are in the image? The most common and straightforward example of this problem is the detection of a single face at a known scale and orientation. This example is referred to as face localization and assumes that it is guaranteed to find the location of a face in an image. Yang *et al*. [18] identifies four major approaches to face detection in still images. These are:

Knowledge based methods

These methods use human knowledge about the face, such as: what does it look like? What are the components? What are the relationships between facial features? For example, there are two eyes in the upper part of the face, one mouth in the center and lower part of the face, and one nose located in the center between the eye line and the mouth, etc. [18]. Such information can help to eliminate a large number of spurious hits in the early stages and then rigorous testing need only be applied to a relatively small number of possible locations of a face in an image.

In this approach, face detection methods are developed based on the rules derived from the researcher's knowledge of human faces. It is easy to come up with simple rules to describe the features of a face and their relationships. For example, a face often appears in an image with two eyes that are symmetric to each other, a nose, and a mouth. The relationships between features can be represented by their relative distances and positions. Facial features in an input image are extracted first, and face candidates are identified based on the coded rules. A verification process is usually applied to reduce false detections.

The problem with this approach is that it is difficult to define all the possible rules using only human knowledge (i.e., if the rules are so strict, some faces will not be detected, and if they are too general, something else may be identified as a face).

Feature invariant approaches

These algorithms aim to find structural features that exist even when the pose, viewpoint, or lighting conditions vary, and then use these to locate faces. These methods are designed mainly for face localization [39].

Template matching methods

These methods have been used for both face localization and detection. Such methods entail using one or more patterns (templates) that reflect a typical face, then scanning this pattern over the targeted image to find the best correlation between the pattern and a window in the image. These patterns can be pre-defined templates or deformable templates.

The problem with this approach is that it is difficult to detect faces at different scales, poses and shapes. Deformable templates are used to solve such problems [18]. In cases where the faces are upright and approximately the same size, this method is preferable for its simplicity and speed.

Appearance based methods

In contrast to template matching, the models (or templates) are learned from a set of training images which should capture the representative variability of facial appearance. These learned models are then used for detection. These methods are designed mainly for face detection [18].

These methods use statistical analysis techniques to classify objects or regions into either face or non-face classes based on a probabilistic framework. The images are represented as variable *x* associated with class conditional probabilities, P(*x*| face) and P(*x*| non-face). As the dimensionality of variable *x* is usually high, then, to compute these probabilities directly, they are transformed to a lower dimensional space using suitable dimensionality reduction techniques given by Equation 3.

$$y = Wx \tag{3}$$

Where y is the output lower dimension vector and W is the transformation matrix. The dimensionality of y is substantially lower as compared to x, suitable for calculating the class conditional probabilities. Bayesian classifiers, artificial neural networks, the Fisher linear discriminant or other suitable classifier can then be used to classify the transformed variable as a face or a non-face class. These methods have been widely used in AVASR research to reduce the high dimensional video data to a reasonably small number of dimensions [9].

Bayesian classification or maximum likelihood is normally used to classify a candidate location as face or non-face. This approach includes several methods such as the Hidden Markov Model (HMM), Naïve Bayes classifier and information-theoretical approach [18].

The second approach is to find a discriminant function such as threshold, separating hyper-plane or decision surface, to determine a candidate location as face or non-face. This approach includes several methods such as distribution-based methods, eigenfaces, support vector machines (SVM), and neural networks [40].

2.8.1.2 Lip Localization

The lips and mouth region are the visual parts of the human speech production system; these parts hold the most visual speech information, therefore, it is imperative for any VSR system to detect/localize such regions to capture the related visual information (i.e., we cannot read lips without seeing them first).

Lip detection methods that have been proposed in the literature so far have their own strengths and weaknesses. The most accurate methods often take more processing time, but some methods proposed for use in online applications trade of accuracy for speed.

Many techniques for lips detection/localization in digital images have been reported in the literature, and can be categorized into two main types of solutions [41]. These are:

Model based lip detection methods

This approach depends on building lip model(s), with or without using training face images and subsequently using the defined model to search for the lips in any freshly input image. The best fit to the model, with respect to some prescribed criteria, is declared to be the location of the detected lips. Each model has its own criteria for measuring a fitness score. These methods include the active contours (snakes), active shape models, active appearance models and deformable templates [41].

Image based lips segmentation methods

This approach uses color difference between skin and lips. Since there is a difference between the color of lips and the color of the face region around the lips, detect lips using color information. The main issue is to determine the most appropriate color space to make the difference between skin pixels and lip pixels. The problem of skin or non-skin pixels characterization has been extensively studied in the context of face detection. But, the point here is to be able to distinguish skin pixels from lip pixels [41, 42]. For this purpose, an appropriate color space needs to be selected from those available, including red, green and blue (RGB), cyan, magenta, yellow, and black (CMYK), and hue, saturation, and value (HSV). This work has adopted the HSV color model for segmentation as this model comes closest to mimicking how humans perceive skin color [41, 42].

2.8.2 Visual Feature Extraction

Given an input video of a person speaking, the task of any visual speech recognition system is to extract visual speech features that could be used for recognizing the uttered words. As discussed in Section 2.5, there are several algorithms of visual feature extraction for lip-reading that be found in the literature. They can be classified according to the type of information source they process: shape-based (high-level), appearance-based (low-level) or a combination of both (hybrid) [27, 28, 29].

In the shape-based approaches, features describe contours of figures, in such a way that a geometrical description of them is provided. These techniques are also called contour-based

approaches. The appearance-based techniques use appearance information (texture, grey level information, colors). These approaches are also called pixel-based approaches [27, 28].

One implementation of each approach is performed and a comparison in terms of recognition results is provided in order to find the most convenient feature extraction for the system. For each approach, different solutions can be found in the literature. In the next paragraph, a summary of the main solutions in each group (shape and appearance-based algorithms) is summarized in order to find out the best one to be implemented.

Appearance-Based Features

The appearance-based features are typically extracted from the region of interest **(**ROI**)** using image transforms, such as transformation to different color space components, where pixel values of typical faces/lips are used.

In the appearance-based feature extraction approaches, pixels from the speaker‘s mouth region are used as source of visual speech information for AVASR. Appearance-based approaches do not need sophisticated algorithms for feature extraction but are generally more sensitive to lighting conditions and pose than are shape-based features. The ROI used is typically either a rectangular or circular region that includes the speaker’s mouth. A vector is then obtained either directly using the color or greyscale values of the pixels in the ROI or some suitable transformation of the pixel values is obtained, such as the DCT [43] or the DWT [8].

The dimensionality of this vector is generally too high to be used directly for statistical modelling of speech classes and one of a number of available dimensionality-reduction techniques is normally applied to render the information suitable for recognition purposes while retaining as much of the original speech information as possible. The two most commonly used techniques for dimensionality reduction are principal component analysis (PCA) [44, 45] and linear discriminant analysis (LDA) [46]. PCA transforms data in such a way that the most of the variance in the data is contained to a small number of parameters called principal components. LDA transforms data so as to maximize the discrimination between different classes.

The Viola-Jones algorithm is also considered an appearance-based algorithm where the features used are Haar wavelet in gray scale intensities sequences [1, 2, 8]. These transforms produce high dimensional data, but the transforms also compact the input signal's energy.

Appearance based approaches differ to other detection techniques as knowledge about the shape and texture of the object is learnt in an exemplar and holistic manner. That is to say, all knowledge about an object is gained from a set of example intensity images without any a priori knowledge about the object [41].

Principal Component Analysis (PCA): principal component analysis (PCA) is one of the most popular linear dimensionality reduction techniques and is widely used in pattern recognition applications [47]. In PCA, the data are transformed into a transform space whose dimensions are ordered according to decreasing variance. A certain number of these dimensions, called the principal components, are then identified as containing sufficient information to represent the original data. These dimensions are considered to capture useful information to provide a distinction between the classes contained in the data and so reveal a hidden underlying pattern in the data which would be difficult to extract in the original data space. A detailed discussion on the theory, calculation and various applications of PCA can be found in [45, 46].

For a given set of data of N dimensions, PCA [45, 46] finds a new space of D orthogonal dimensions (D < N) such that the data points mainly lie along these D dimensions. Let M observations of an N dimensional data vector x be represented by a matrix X of order NxM such that each column of X represent one observation of the data vector x. Let the D principal axes be denoted by $T_1, T_2, \ldots, T_D$. These principal axes could be given by the eigenvectors of the covariance matrix S, as shown in Equation 4.

$$ST_i = \lambda_i T_i \qquad i = 1,2,3 \ldots D \tag{4}$$

Where λ_i, is the i^{th} largest eigenvalue of $\boldsymbol{S}$ and $\boldsymbol{S}$ defined as in Equation 5.

$$S = \frac{1}{M}\sum_{j=1}^{M}\left(x_j - \mu\right)^T\left(x_j - \mu\right) \tag{5}$$

Where μ is the mean of the observation vectors and x_j is the j[th] observation vector. As the larger is the value of λ, then the larger is the variance and so the maximum data variance can be found by selecting the first few components in the projected space. A measure for representing the portion of data is the percentage variance. The projected D dimension matrix is given by Equation 6.

$$\boldsymbol{Y} = [Y_1, Y_2, .., Y_D]=[T_1^T \boldsymbol{X},\ T_2^T \boldsymbol{X},\ T_3^T \boldsymbol{X}, \dots, T_D^T \boldsymbol{X}]=T^T \mathbf{X} \quad (6)$$

Where T is the transformation matrix whose columns are made of the principal axisT_i. The DxM dimensional matrix Y thus obtained contains the desired principal components of input matrix X of dimensionality NxM. Although the features extracted using PCA have a minimum correlation along the direction of the principal axis, the approach does not guarantee the separation of classes among data as no class information is used in the PCA calculation. PCA also has a limitation of scale sensitivity implying that the principal components may be affected by the relative scaling of variables in original data.

Linear discriminant analysis: The transformation performed by linear discriminant analysis (LDA) is able to separate the elements of different classes while at the same time minimizing the distance between elements of same class [48]. This approach comes under the domain of supervised dimensionality reduction methods, meaning that prior knowledge of the classes present in the data is used in performing the transformation.

Let the data matrix **X** contain observation vectors from *k* classes, x_1, x_2,..., x_k each having *N* dimensions. If the j^{th} observation of class *i* is represented by x_{ij} such that *i* = 1, 2, …, *k* and *j* =1, 2, …, , M_i, where the M_i is the number of observations in class *i*. The mean of observations in class *i* is then given by Equation 7.

$$\mu_i = \frac{1}{M_i} \sum_{i=1}^{M_i} x_{ij} \quad (7)$$

And the covariance matrix for class i is given by Equation 8.

$$\mathbf{S}_i = \frac{1}{M_i}\sum_{i=1}^{M_i}\left(x_{ij} - \mu_i\right)\left(x_{ij} - \mu_i\right)^T \tag{8}$$

For k classes, the within-class variance $\mathbf{S}_w$ is given by Equation 9.

$$\boldsymbol{S}_w = \sum_{i=1}^{k}\boldsymbol{S}_i \tag{9}$$

And the between-class variance $\mathbf{S}_b$ is defined by Equation 10.

$$\mathbf{S}_b = \sum_{i=1}^{k} M_i\left(\mu_i - \mu\right)\left(\mu_i - \mu\right)^T \tag{10}$$

Where μ is the mean of all the data given by in Equation 11.

$$\mu = \frac{1}{M}\sum_{j=1}^{k}\sum_{i=1}^{M_i} x_{ij} \tag{11}$$

And M is the total number of data vectors such that $M \Sigma M_i$ for i= 1, 2,…, k. The transformation from N-dimensional space to a lower D-dimensional space is performed by Equation 12.

$$\boldsymbol{Y} = \boldsymbol{W^T X} \tag{12}$$

Where **W** is the transformation matrix. The greatest separation between classes can be achieved by maximizing the Fisher Linear Discriminant operator

$$\boldsymbol{J(W)} = \frac{\boldsymbol{W^T S_b W}}{\boldsymbol{W^T S_w W}} \tag{13}$$

The optimum W consists of the D largest eigenvectors, where D is the desired dimensionality of the transformed space.

Discrete Cosine Transform (DCT): The DCT is one of the most popular tools used in image analysis research. It describes an image in terms of its frequency components and is widely used in image reconstruction, filtering and image compression applications. The use of the DCT in pattern recognition research is well established with the majority of AVASR systems using the DCT transformation as a first stage of the visual front-end [43]. The DCT

transformation is lossless and an inverse transform can be performed to reconstruct the original image from the DCT coefficients. The DCT is often used to exploit the inter-pixel and inter-frame redundancies present in images and in video data for compression. A detailed discussion on DCT theory and its applications to image and video analysis can be found in [43].

The number of frequency components generated in the DCT transform domain corresponds to the dimensionality of the input signal. Thus the output of the DCT transform on a sequence of length N will be a sequence of same length N. For a two dimensional image signal of dimensionality MxN, the output of the DCT transform is a matrix of the same order MxN. As the DCT is a separable transform, the two dimensional DCT of an image can be performed by applying a one-dimensional DCT in one dimension followed by a second one-dimensional DCT performed in the second.

A one-dimensional DCT $y[f]$ of a sequence $x[n]$ of length N can be performed by Equation 14.

$$y[f] = r[f] \sum_{n=0}^{N-1} \mathrm{x[n]} \cos\left[\frac{\pi(2n+1)f}{2N}\right] \quad f = 0,1,2,\dots,N-1 \tag{14}$$

Where the coefficient r[f] is defined as shown in Equation 15.

$$r[f] = \begin{cases} \sqrt{\frac{1}{N}} & for\ f = 0 \\ \sqrt{\frac{2}{N}} & for\ f = 1,2,\dots,N-1 \end{cases} \tag{15}$$

The first coefficient $f\,[0]$ in the DCT domain represents the mean value (or energy) of the sequence known as DC (direct coefficient) component.

The two-dimensional DCT $y\,[u, v]$ of a matrix $x\,[m, n]$ of dimension MxN is given by Equation 16.

$$y[u,v] = r[u]r[v] \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \mathrm{x[m,n]} \cos\left[\frac{\pi(2m+1)u}{2M}\right] \cos\left[\frac{\pi(2n+1)v}{2N}\right] \tag{16}$$

u= 0,1,…,M-1

v =0,1,…,N-1

The coefficients $r[u]$ and $r[v]$ are defined by Equations 17 and 18 respectively.

$$r[u] = \begin{cases} \sqrt{\frac{1}{M}} & for\ u = 0 \\ \sqrt{\frac{2}{M}} & for\ u = 1,2, \dots, M-1 \end{cases} \tag{17}$$

$$r[v] = \begin{cases} \sqrt{\frac{1}{N}} & for\ v = 0 \\ \sqrt{\frac{2}{N}} & for\ v = 1,2, \dots, N-1 \end{cases} \tag{18}$$

As the neighboring pixels in an image are generally highly correlated, the DCT transform coefficients of the high frequency components are usually small and, as they contribute little to the perceived image, are often discarded; this being known as *lossy* compression. In this process, although some information contained in the original image is lost, frequencies containing important information are retained, thus resulting in little or no effect on the perceived visual quality of the image. Such inter-pixel redundancy has also been applied in DCT based AVASR systems in an attempt to achieve a compact representation of speech related information from the speaker's mouth region (ROI) [49].

Viola-Jones Algorithm: The basic principle of the Viola-Jones algorithm is to scan a sub-window capable of detecting faces across a given input image. The standard image processing approach would be to rescale the input image to different sizes and then run the fixed size detector through these images. This approach turns out to be rather time consuming due to the calculation of the different size images. Contrary to the standard approach Viola-Jones rescale the detector instead of the input image and run the detector many times through the image each time with a different size. At first one might suspect both approaches to be equally time consuming, but Viola-Jones have devised a scale invariant detector that requires the same number of calculations whatever the size. This detector is constructed using a so-called

integral image and some simple rectangular features reminiscent of Haar wavelets. The next section elaborates on this detector [50].

The first step of the Viola-Jones face detection algorithm is to turn the input image into an integral image. This is done by making each pixel equal to the entire sum of all pixels above and to the left of the concerned pixel. This is demonstrated in Figure 2.2 [50].

Input Image

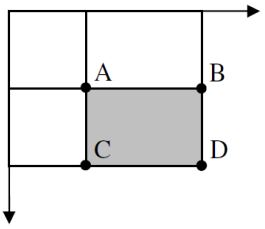
1	1	1
1	1	1
1	1	1

Integral Image

1	2	3
2	4	6
3	6	9

Figure 2.2: *The Integral Image.*

This allows for the calculation of the sum of all pixels inside any given rectangle using only four values. These values are the pixels in the integral image that coincide with the corners of the rectangle in the input image. This is demonstrated in Figure 2.3 [50].



Sum of grey rectangle = D - (B + C) + A

Figure 2.3: *Sum Calculation.*

Since both rectangles B and C include rectangle A, the sum of A has to be added to the calculation. It has now been demonstrated how the sum of pixels within rectangles of arbitrary size can be calculated in constant time. The Viola-Jones face detector analyzes a given sub-window using features consisting of two or more rectangles. The different types of features are shown in Figure 2.4 [50].

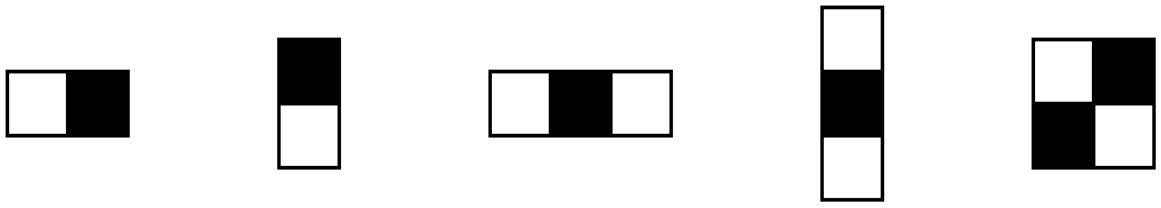


Figure 2.4: *The different Types of Features.*

Each feature results in a single value which is calculated by subtracting the sum of the white rectangle(s) from the sum of the black rectangle(s). Viola-Jones have empirically found that a detector with a base resolution of 24*24 pixels gives satisfactory results. When allowing for all possible sizes and positions of the features in Figure 2.4 [50], a total of approximately 160,000 different features can then be constructed. Thus, the amount of possible features vastly outnumbers the 576 pixels contained in the detector at base resolution. These features may seem overly simple to perform such an advanced task as face detection, but what the features lack in complexity they most certainly have in computational efficiency [50].

Shape-Based Features

In these approaches, the shape of a speaker‘s lips or the face contour itself is used to generate the speech related information for speech recognition. One approach is to obtain geometric features such as the length, width, area and perimeter of the inner or outer parts of the lips. Also, statistical models have been developed to describe the shape of lips or the face [29].

Shape-based visual features can be defined as either geometric or model based. In both cases, an algorithm that extracts the inner and outer lip contours or the entire face shape, is required. Geometric features that are meaningful to humans can be extracted from the inner and outer contours of the lip, such as the height, width, perimeter, and area within the contour. Such features contain significant visual speech information, and have been successfully used in speech reading [51]. Alternatively, model-based visual features can be obtained in conjunction with one of the parametric or statistical lip-tracking algorithms.

Shape model-based feature detections build a number of parametric models for face/lip contours. Some popular methods for this task are Fourier and image moment descriptors of the lip contours [52], active contour or ‘snakes’ [34,53], active shape model (ASM) [54] and active appearance models (AAM) [5,54].

A ‘snake’ is an elastic curve represented by a set of control points and is used to detect important visual features, such as lines, edges, or contours. The snake control point coordinates are iteratively updated, converging towards a minimum of the energy function, defined on basis of curve smoothness constraints and a matching criterion to the desired features of the image [34]. ASMs are statistical models obtained by performing PCA on vectors containing the coordinates of a training set of points that lie on the shapes of interest, such as the eyes, nose, and mouth contours. Such vectors are projected onto a lower dimensional space defined by the eigenvectors that correspond to the largest PCA eigenvalues, representing the axes of genuine shape variation [54]. AAMs are an extension to ASMs with two more PCAs, where the first captures the appearance variation around the ROI and the second PCA is built on concatenated weighted vectors of the shape and appearance representations. AAMs, thus, remove the redundancy that would arise due to shape and appearance correlation and they create a single model that compactly describes shape and the corresponding appearance deformation [54].

Hybrid Feature Approaches

The use of appearance and shape features each has their own strengths and limitations. In an attempt to harness the advantages of both, appearance and shape based features have been combined to make a third class of features known as hybrid features, normally by using a simple concatenation of the two types.

Hybrid approaches utilize both shape and appearance based features to create the final feature vector. The feature vector may be some concatenation of geometric and appearance based features, or, as in the case of AAMs, may be a parametric representation using a joint shape-appearance model [29].

2.9 Recognition

Many different classifiers have been applied to the area of speech recognition, which is a difficult classification task due to the fact that the signals involved are time varying and of different temporal lengths. Whatever the final choice of representation of the visible speech gestures (visemes), the other major issue is how to recognize this information along with the information about the acoustic stream of information so that the best use can be made of the two modalities together. A number of recognition approaches have been proposed in literature

for the task of audio-visual recognition. These approaches include: a simple weighted distance in visual feature space [55], artificial neural networks [56, 57], support vector machines (SVMs) [58] and dynamic bayesian networks (DBNs) [59], which include hidden markov models (HMMs). We will discuss in this section some of the most widely used approaches.

2.9.1 Dynamic Bayesian Networks (DBNs)

The HMM as well as other audio-visual models used in existing AVSR systems, are special cases of dynamic Bayesian networks. DBNs are directed graphical models of stochastic processes in which the hidden states are represented in terms of individual variables or factors. A DBN is specified by a directed acyclic graph, which represents the conditional independence assumptions and the conditional probability distributions of each node [59].

Hidden Markov Models (HMM)

The HMM provides a stochastic framework that is commonly used for speech recognition. It is the most commonly used classifier in both audio-only and audio-visual speech recognition. HMMs statically model transitions between the speech classes and assume a class-dependent generative model for the observed features [29, 60].

Hidden Markov models represent a doubly stochastic process in which one process, the "hidden", or unobservable process, progresses through a discrete state space while a second observable process takes on distinct stochastic properties dependent upon the hidden state. In this context, unobservable implies that the process itself does not emit any information that one may directly gather; therefore it is hidden from the observer. One, however, may infer information about this hidden process by gathering information produced by the directly observable process due to its dependence on the hidden process. This inference lies at the heart of HMMs.

Table 2.3 [29] summarizes the notation used when working with hidden Markov models and Figure 2.5 [29] shows left-to-right single-stream HMM. The three primary issues facing HMMs are described in [29, 60].

i. **Evaluation:** How does one evaluate the probability of an observed sequence given the model parameters?

Given a model HMM λ = (A, B, π) and the observation sequence O=o_1, o_2 ..., o_K , calculate the probability that model λ has generated the observed sequence O. How do we efficiently compute the probability of the observation sequence given the model P (O/λ)? This problem is used to evaluate how well a given model matches a given observation sequence or how likely is an observed sequence to have been generated by a given HMM? The forward algorithm is quite efficient solution for this problem.

ii. **Hidden state recovery:** How can the hidden state sequence be determined from an observation sequence given the model parameters?

Given the model HMM λ = (A, B, π) and the observation sequence O=o_1,o_2,..., o_K, calculate the most likely sequence of hidden states Q=q_1, q_2,..., q_T, that produced this observation sequence O. In order to solve this decoding problem which is finding the optimal state sequence, using Viterbi algorithm is the best solution.

iii. **Model updating:** How can one determine the parameters of an HMM from multiple observations?

Given some training observation sequences O=o_1, o_2,..., o_K, and general structure of HMM (numbers of hidden and visible states), adjust the model parameters λ = (A, B, π) to maximize the probability. This problem can be solved by the Baum-Welch re-estimation algorithm. Baum-Welch re-estimation algorithm is the forward and backward procedures combination.

Table 2.3: *Notation Reference for Hidden Markov Models.*

Notation	Representation
o_t	Vector of total observations at time t
O_t^a , O_t^v	Vectors of audio and video observations, respectively at time t
Q	Time sequence of hidden states
π_{q_1}	Probability of starting in hidden state
$a_{q_1 q_2}$	Probability of transitioning from hidden state q_1 to q_2
$b_j(o_t)$	Probability of state j emitting observation o_t
λ_i	Hidden Markov model parameters for model i, including state transition probabilities , state emission probabilities , and state probabilities

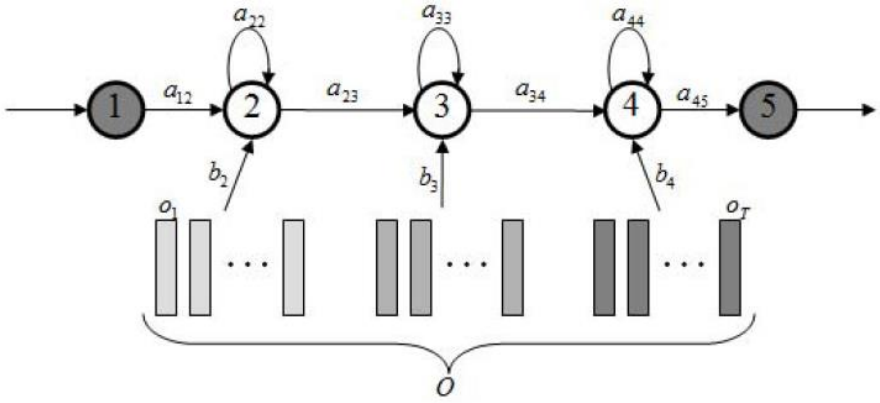


Figure 2.5: *Left-to-right Single-stream HMM*

If single stream HMMs are to be employed for speech recognition, audio and visual features must be combined into a single observation vector,o_t consisting of the audio observation vector O_t^a concatenated with the visual observation vector O_t^v as shown in Equation 19.

$$o_t = \begin{bmatrix} O_t^a \\ O_t^v \end{bmatrix} \tag{19}$$

Most commonly, Gaussian mixture models (GMMs) are used to model the state emission probability distributions, which can be expressed as in Equation 20.

$$b_j(o_t) = \sum_{m-1}^{M} C_{jm} N(o_t; u_{jm}, \sum_{jm}) \tag{20}$$

Where, b_j refers to the emission distribution for state *j* in a context-dependent HMM as in Figure 2.6 [28]. The Gaussian mixtures weights are denoted by c_{jm}, for all *M* Gaussian mixtures, and *N* stands for a multivariate Gaussian with mean, u_{jm}, and covariance matrix, $\sum_{jm}$. The sum of all mixtures weights,C_{jm}, should be 1. Now, recognition occurs by summing the joint probability of a set of observations and state sequences over all possible state sequences for each model, which is expressed in Equation 21.

$$\arg\max_{\lambda_i} P(O, Q|\lambda_i) = \arg\max_{\lambda_i} \sum_{\forall Q} P(O, Q|\lambda_i)\ P(Q|\lambda_i) \tag{21}$$

In Equation 21 λ_i stands for the i^{th} word model, Q represents all combinations of state sequences, and we are summing over all possible state sequences for the given model. More specifically, given a series of observations O = $[o_1, o_2, \dots o_T]$, state-transition likelihoods, $a_{q_{t-1}q_t}$, state-emission probabilities, $b_j(o_t)$ and the probability of starting in a state π_q for each model, the word with the highest probability of having generated O can be determined by summing over all possible state sequences, Q = $[q_1, q_2, \dots q_T]$, as shown in Equation 22.

$$\sum_{\forall Q} P(O, Q|\lambda_i)\ P(Q|\lambda_i) = \sum_{q_1, q_2, \dots q_T} \pi_{q_1} b_{q_1}(o_1) o_{q_1 q_2}(o_2) \dots o_{q_{T-1} q_T}(o_T) \tag{22}$$

So the recognized word will have a certain state sequence with a higher probability of generating the observation vector, O, than any other word model. It is worth noting that the modeled objects could also be phonemes or visemes. It is clear that the brute force method for computing probabilities for all models and combinations of state sequences becomes infeasible even for relatively small vocabularies.

When multiple modalities are present, multi-stream HMMs, stream weights are commonly used to integrate stream information as part of the evaluation process. The state-topology of a typical multi stream HMM is shown in Figure 4. In the case of audio-visual speech or speaker recognition, audio and visual stream weights are applied as exponential factors to each modality in calculating the state emission probability, which is expressed in Equation 23.

$$b_j(o_t) = \prod_{s \in \{a,v\}} \left[\sum_{m=1}^{M} o_{jsm} N(O_t^a; u_{jm,} \sum_{jsm}) \right]^{\gamma_s} \tag{23}$$

The index, *s*, indicates either the audio or visual modality, and the exponential weight, γ_s reflects the importance of the audio or stream weight in the recognition process. It is often assumed that the stream weights sum to one, $\gamma_a + \gamma_v = 1$.

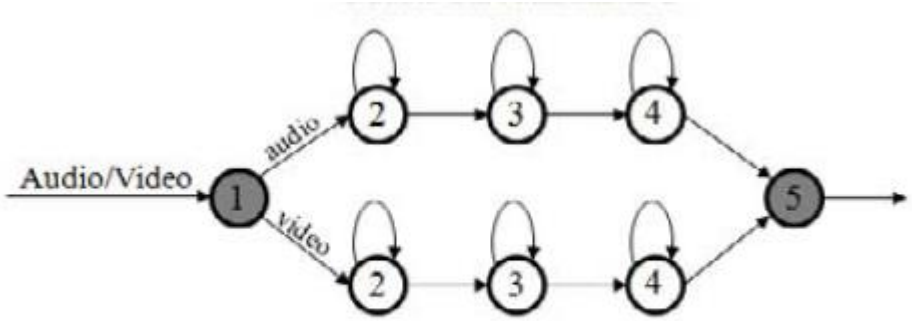


Figure 2.6: *HMM State Topology for a 3-state, Multi- stream HMM*

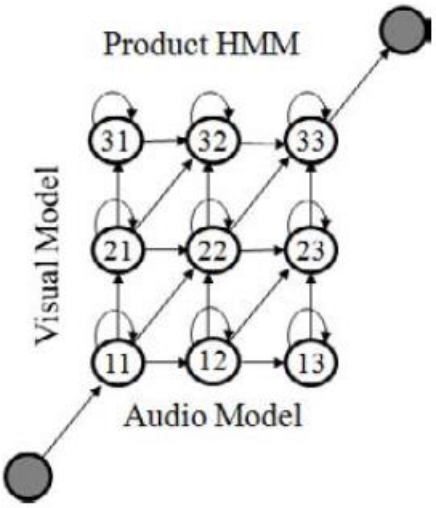


Figure 2.7: *A Product HMM with 9 States*

As discussed in Section 2.10, there are two principal models for audio-visual fusion: feature fusion and decision fusion. Feature fusion models combine acoustic and visual features into a single feature vector and transmit them directly to a single, bimodal classifier. For these models, a regular left-to-right HMM [36] is used. On the other hand, in decision fusion systems, two parallel, unimodal classification systems are employed and the results from each are fed forward for fusion and final decision making, for example, on a probabilistic basis. For these kinds of systems, conventional HMM recognizers are useless because they assume asynchrony of the visual and acoustic data (which is not always the case). Therefore, other models have been used. Some of the most successful decision fusion models include the Multi-Stream HMM (MSHMM) [61], the Product HMM (PHMM) [29, 36], the Independent

HMM (IHMM) [62], the Factorial HMM (FHMM) [63] and the Coupled HMM (CHMM) [64, 65].

2.9.2 Neural Networks (NN)

In contrast to HMMs, Neural Networks (NN) make only few assumptions about the underlying data and thus they can be generalized to large classes using sufficiently large training data. However, training is slow and asynchrony modeling is difficult to achieve. One such approach is Multiple State-Time Delayed Neural Network (MS-TDNN) for recognition of the audio-visual speech task. Combining visual and acoustic data is done on the phonetic layer or on lower levels [66].

Neural networks can also be used to convert acoustic parameters into visual parameters. In the training phase, input patterns and output patterns are presented to the network, and an algorithm called back propagation can be used to train the network weights. The design choice lies in selecting a suitable topology for the network. The number of hidden layers and the number of nodes per layer may be experimentally determined. Furthermore, a single network can be trained to reproduce all the visual parameters, or many networks can be trained with each network estimating a single visual parameter.

2.10 Multimodal Fusion

Multimodal fusion is a very important research area that relies on measuring a set of complementary features from multiple sensors or modalities and combining these features in an “intelligent” way that maximizes information gather and minimizes the impact of noise coming from the individual sensors. In AVSR, the issue of multimodal fusion has received a lot of attention, as it aims to combine the multiple speech informative streams into a multimodal classifier that can achieve better classification results than the audio and visual-only classifiers. The first issue to be addressed in fusion of audio-visual speech is where the fusion of the data takes place. Cognitive studies have suggested three architectures for the combination of audio and visual modalities. These are early integration, late integration and hybrid fusion [8, 67, 68].

2.10.1 Feature Level Fusion

Feature fusion assumes class-conditional dependence between streams and frame synchronous information integration. This is referred to in literature as feature fusion, direct identification (DI), or early integration. In this case, the audio and visual features are used simultaneously and equally for classification using a bimodal classifier. Feature-level fusion algorithms train this classifier on the concatenated vector of audio and visual features or any appropriate transformation of it. Examples of feature fusion methods include plain feature concatenation and hierarchical discriminant feature extraction [36]. Concatenate feature fusion is the simplest fusion technique. Given time-synchronous audio and visual feature vectors $o_t^{(A)}$ and $o_t^{(V)}$, with dimensionalities D_A and D_V respectively, this method generates a concatenated audio-visual feature vector at every time instance t expressed by Equation 24 [69].

$$o_t^{(AV)} = [o_t^{(A)T},\ o_t^{(V)T}]^T \quad \in R^D \tag{24}$$

Where D = D_A + D_V is the dimensionality of the combined feature vector. This can result in a very large feature vector which may not be suitable for representing the underlying data.

EI techniques for audio-visual speech are of benefit as they model the dependencies between acoustic and visual speech modalities directly. However, EI approaches suffer in two respects. First, if the acoustic or visual speech modalities are corrupted then the entire speech modality is corrupted due to classification occurring at such a low level. Second, there is an assumption that the acoustic and visual speech modalities are synchronized at the state level when HMM classifiers are being employed [30].

2.10.2 Decision Level Fusion

Decision fusion incorporates separate recognizers for audio and video channels and then combines the outputs of the two recognizers to get the final result. The final step of combining the two outputs is the most important step in this approach, as it has to deal with the issues of orthogonally between the two channels and the reliability of the channels. The level in which decisions can be fused is a variable parameter that offers potential solutions. Many different levels have been proposed such as state level, phone level or word level. For automated

isolated word applications late integration (LI) strategies have reported superior results to EI for speech recognition tasks [30].

The problem with feature fusion techniques is that they provide no way of capturing the reliability of the individual streams of information. Reliability in audio-visual integration is an important issue because factors such as noise, face occlusions and volume of the speaker's sound can lead to a certain modality being more trustworthy than the other. Decision-level fusion (also called separate identification (SI) or late integration), on the other hand, uses the two outputs of the audio and visual classifiers to combine the two modalities [69].

In the case where single-stream HMMs, with the same set of speech classes (states), are used for both audio-only and visual-only classification, we consider this likelihood combination to be at a frame (HMM state) level, and it is modeled by means of a multi-stream HMM. The state-dependent emission of the audio-visual observation vector $o_t^{(AV)}$ is thus governed by using Equation 25 [69].

$$P(o_t^{AV} \mid c) = P\left(o_t^{(A)} \mid c\right)^{\lambda A} P\left(o_t^{(V)} \mid c\right)^{\lambda V} \tag{25}$$

For all HMM states c ∈ C.

Decision fusion at a state or frame level is not good enough because the states are probably not in synchrony. For this reason, decision should be done at a later stage. One approach could be to wait until the end of an utterance and then fuse the decisions about the different streams based on their log-likelihoods [69]. In this integration strategy decisions of each classifier as shown in Equation 26.

$$P_{Late}\,(W_j|\ O^{AV}) = P\,(W_j|\ O^A).\,\alpha_A + P\,(W_j|\ O^V).\,\alpha_v \tag{26}$$

This is the reason why it is also called decision fusion. The final solution will be the word in Equation (26) that maximizes the combined probability of Equation (27).

$$\underset{W_j}{arg\,max}\{\ P_{Late}\,(W_j|\ O^{AV})\ \} \tag{27}$$

In decision fusion each of these independent models will be defined by a set of parameters that will be obtained in two independent training processes one for the audio and another one for video as shown in Equations 28 and 29 respectively.

$$P\,(W_j|\ \lambda_{w_j}) = f\{a_{w_j}^{A}\ ,\ b_{w_j}^{A}\ ,\ \pi_{w_j}^{A}\} \tag{28}$$

$$P\,(W_j|\ \lambda_{w_j}) = f\{a_{w_j}^{v}\ ,\ b_{w_j}^{v}\ ,\ \pi_{w_j}^{v}\} \tag{29}$$

Weighted Bayesian Fusion

In pattern classification problems, we measure a property (feature) of a pattern instance and try to decide to which of M classes c_i, i = 1,…, M it should be assigned. Multimodal fusion or integration combines S complementary features, originating from a single or multiple modalities, in order to maximize information gather and to overcome the impact of noise in each individual stream.

Let S_k; k = 1, …, S, denote the information streams that we want to integrate.

Let X_s; s = 1, …, S, denote the feature vectors of every stream.

The simplest way to combine audio and video data is to use Bayes' rule and multiply the audio and video a posteriori probabilities. From a probabilistic perspective, this approach is valid if the audio and video data are independent. Perceptive studies have shown that in human speech perception, audio and video data are treated as class conditional independent [67]. In this case, the conditional probability of the observation vector $X_{1:s} = (X_1, \dots,\ X_s)$ given the class label c_i is governed by the product:

$$P\,(X_{1:s}|c_i) = P\,(X_1, \dots, X_s|c_i) = \prod_{s=1}^{s} P\,(X_s|c_i) \tag{30}$$

Using Bayes' rule, we get the desired a posteriori probability of the class given the features:

$$P\,(c_i|X_{1:S}) = \frac{\prod_{S=1}^{S} P\,(c_i|X_S)}{P\,(c_i)} \frac{\prod_{S=1}^{S} P\,(X_S)}{P\,(X_{1:S})} \tag{31}$$

By replacing the probabilities P by estimates $\hat{P}$ we get a representation of the Bayesian Fusion (BF):

$$\hat{P}_{BF}(c_i|X_{1:S}) = \frac{\prod_{S=1}^{S} P\,(c_i|X_S)}{P\,(c_i)} . \eta \tag{32}$$

Where, the terms independent of the actual class are replaced by the normalization factor η

$$\eta = \frac{1}{\sum_{j=1}^{M} \frac{\prod_{S=1}^{S} P\,(c_i|X_S)}{P\,(c_i)}} \tag{33}$$

Where *M* is the number of classes. This probability can then be used in classification by making use of the *Maximum A Posteriori (MAP)* rule:

$$\hat{c}=\text{argmax } \hat{P}_{BF}(c_i|X_{1:S}) \qquad \text{where } c_i \in \text{C} \tag{34}$$

The standard Bayesian Fusion approach does not deal with varying reliability levels of the input streams. In order to improve classification performance, several authors have introduced stream weights $\{\lambda_1, \lambda_2,\lambda_3 ,\ldots, \lambda_s\}$ as exponents in Equation 32, resulting in the modified score:

$$\hat{P}_{WBF}(c_i|X_{1:S})=\frac{\prod_{S=1}^{S} P(c_i|X_s)^{\lambda_s}}{\sum_{j=1}^{M}\prod_{S=1}^{S} P(c_i|X_s)^{\lambda_s}} \tag{35}$$

Notice that Equation 35 corresponds to a linear combination in the log likelihood do main; however, it does not represent a probability distribution in general, and will consequently be referred to as a score. Such schemes have been motivated by potential differences in reliability among different information streams, and larger weights are assigned to information streams with better classification performance. Using such weighting mechanisms has experimentally been proven beneficial for feature integration in both intra-modal and inter-modal scenarios.

In order to determine the weights $\{\lambda_1, \lambda_2,\lambda_3 ,\ldots, \lambda_s\}$ we first need to define reliability measures for the individual streams. These reliability measures should reflect the quality of the observation conditions by considering statistical information conveyed in both prior and current classification results. The second step is to find an optimal mapping between these reliability indicators and the stream weights $\{\lambda_1, \lambda_2,\lambda_3 ,\ldots, \lambda_s\}$ The only constraints that this problem has are that the weights should be positive and should add up to 1.

$$\sum_{S=1}^{S} \lambda_s = 1, \lambda_s \geq 0 \tag{36}$$

2.10.3 Hybrid Fusion

In hybrid fusion, the audio and visual modalities are integrated at a stage intermediate between the two extremes of feature and decision fusion. Although there is a range of possible levels at which integration could take place, most commonly hybrid fusion occurs at state level due to its simplicity of implementation in a multi-stream HMM framework [48]. Hybrid fusion

thus attempts to exploit the individual advantages of both feature and decision fusion, in particular capturing the mutual dependencies of the audio and visual modalities while at the same time giving a better control of modality reliability compared to feature fusion [2].

2.11 Types of Speech Recognition System

When developing a speech recognition system, machines have to do their jobs by considering so many factors or variabilities that have to be taken into account. Ideally, a speech recognition system should be free from any constraint, this means a speech recognizer can be speaker independent, continuous speech, very large vocabulary and spontaneous speech. Based on those factors, types of ASR systems as follows [15, 47, 48]:

- Enrollment (Speaker dependent or independent): Speaker-independent systems are developed to recognize speech from any new speaker. Speaker dependent systems, on the contrary, recognize only the speech of people who were involved during the development of the recognizer.
- Speaking Mode (Isolated /discrete/ or continuous speech): Isolated or discrete word recognizer is unnatural way of reading words such that there is a brief pause between them which could make easier to find the beginning and end points of words which in effect makes the pronunciation of each word not to have an impact on the others. Whereas in continuous speech recognizers, a user speaks naturally where words do not get separated from one another by pauses.
- Speech Types (Read or Spontaneous): depending on the types of speech either scripted, planned or spontaneous [48]. ASR system can be classified as read or spontaneous speech recognizer. Building the Spontaneous speech recognition is more difficult both acoustically and grammatically than the read speech recognizer just because spontaneous speech has characteristics such as false starts, incomplete sentences, long pauses and mispronunciations, unrestricted vocabulary which in ultimately reduces pronunciation quality.
- Vocabulary Size (Small, Medium or Large Vocabulary system): the vocabulary size of small vocabulary recognition systems is limited between 1 and 99 whereas medium

and large vocabularies have 100-999 words and 1000 words or more respectively as described in [48].

2.12 Summary

In this chapter, the architecture of AVASR systems is discussed. Current approaches are reported in the AVASR literature and their relative advantages and disadvantages are identified. Some of component parts of AVASR systems, such as classifier methods and modality fusion are multidisciplinary research areas while the audio front-end design and ROI detection and extraction are performed by approaches borrowed from other research areas such as audio-only ASR and image analysis research. The main focus of AVASR is the extraction of speech informative visual features to complement and supplement the audio steam, particularly when the audio channel is noisy. As the quality of extracted visual feature values are greatly dependent on the accurate extraction of the mouth ROI, it is potentially beneficial to view the ROI extraction task from an AVASR perspective, in particular to exploit the information available in video sequences, in contrast to image analysis approaches where the image segmentation is based only on information available in individual images.

Chapter Three: Related Work

3.1 Introduction

In this chapter, we review some of related works on audio speech recognition and audio-visual speech recognition for Amharic and other languages. This chapter also presented the research gap on audio only speech recognition.

3.2 Speech Recognition for Amharic Language

Automatic speech recognition for Amharic was conducted in 2001 by Solomon Berhanu [10].The author developed isolated Consonant-Vowel syllable Amharic recognition system which recognizes a subset of isolated consonant-vowel (CV) syllable using HTK (Hidden-Markov Modeling Toolkit). The author selected 41 CV syllables of Amharic language out of 234 and the speech data of those selected CV syllables were recorded from 4 males and 4 females with the age range of 20 to 33 years. The average recognition accuracies were 87.68% and 72.75% for speaker dependent and independent systems, respectively.

Kinfe Tadesse [11] developed a sub-word based isolated Amharic word recognition systems using HTK (Hidden Markov Model Toolkit). In this experiment, phones, triphones, and CV-syllables were used as the sub-word units and selected 20 phones out of 37 and 104 CV syllables for developing the system. The speech data of those selected recorded from 15 speakers for training and 5 speakers for testing. Average recognition accuracies of 83.07% and 78% were obtained for speaker dependent phone-based and triphone-based systems respectively. With respect to speaker independent systems, average recognition accuracies of 72% and 68.4% were obtained for phone and triphone-based speaker independent systems respectively.

Asratu Aemiro [70] developed two types of Amharic speech recognition (ASR) systems, namely canonical and enhanced speech recognizers. The canonical ASR system is developed based on the canonical pronunciation model which consists of canonical pronunciation dictionary and decision tree. The canonical pronunciation dictionary is prepared by incorporating only a single pronunciation for each distinct word in the vocabularies. The canonical decision tree is constructed by only considering the place of articulations of phonemes as it was commonly used by the previous Amharic ASR researchers. On the other

hand, the development of enhanced speech recognition system takes enhanced pronunciation model which consists of enhanced pronunciation dictionary and enhanced decision tree where both are designed by considering the patterns we identified based on the co-articulation effects of phonemes. The construction of the enhanced pronunciation model incorporates alternative pronunciations for each distinct word according to the identified patterns. Finally, the author evaluated the recognition accuracy of the two ASR systems by introducing the enhanced pronunciation model into Amharic ASR systems, and obtained an improvement of 14.04% and 13.93% at the word and sentence level, respectively using enhanced ASR system. Furthermore, the author tested the recognition accuracy of the two ASR systems with different parameters and the test results are reported. Accordingly the author recommend that incorporating the enhanced pronunciation model in to Amharic ASR systems would be important in order to improve the recognition accuracy of the recognizers.

3.3 Audio-Visual Speech Recognition for other Languages

Stéphane and Juergen [25] proposed speech recognition system that uses both acoustic and visual speech information to improve the recognition performance in noisy environments for French language. Their system consists of three components: 1) a visual module; 2) an acoustic module; and 3) a sensor fusion module. The visual module locates and tracks the lip movements of a given speaker and extracts relevant speech features. This task is performed with an appearance-based lip model that is learned from example images. Visual speech features are represented by contour information of the lips and grey-level information of the mouth area. The acoustic module extracts noise-robust features from the audio signal. Finally, the sensor fusion module is responsible for the joint temporal modeling of the acoustic and visual feature streams and is realized using multi-stream hidden Markov models (HMMs). The multi-stream method allows the definition of different temporal topologies and levels of stream integration and hence enables the modeling of temporal dependencies more accurately than traditional approaches. They present two different methods to learn the asynchrony between the two modalities and how to incorporate them in the multi-stream models. The superior performance for the proposed system is demonstrated on a large multi-speaker database of continuously spoken digits. On a recognition task at 15 dB acoustic signal-to-noise ratio (SNR), acoustic perceptual linear prediction (PLP) features lead to 56% error rate,

noise robust RASTA-PLP (Relative Spectra) acoustic features to 7.2% error rate and combined noise robust acoustic features and visual features to 2.5% error rate.

Paul Duchnowski *et al* [26] presented the development of a modular system for flexible human–computer interaction via speech. The speech recognition component integrates acoustic and visual information (automatic lip-reading) improving overall recognition, especially in noisy environments. The image of the lips, constituting the visual input, is automatically extracted from the camera picture of the speaker's face by the lip locator module. Finally, the speaker's face is automatically acquired and followed by the face tracker sub-system. Integration of the three functions results in the first bi-modal speech recognizer allowing the speaker reasonable freedom of movement within a possibly noisy room while continuing to communicate with the computer via voice. Compared to audio-alone recognition, the combined system achieves a 20 to 50 percent error rate reduction for various signal/noise conditions. They also presented the components of a lip-reading/speech recognition system that *non-invasively* and automatically captures the required visual information. The system which comprises them performs automatic lip-reading in realistic situations where lip motion information enhances speech recognition under both favorable and acoustically noisy conditions. Simultaneously, the speaker is allowed a reasonable freedom of movement within a room, with no need to position himself in any particular location.

Ayaz *et al* [4] described a lip-reading technique based on motion capturing using optical flow analysis and support vector machine (SVM) for English language. The experimental results demonstrate that the inter and intra subject speed of speech variations can be overcome through normalization using linear interpolation and mean square Error (MSE), and the vertical component of optical flow can be used for speech recognition. A robust feature selection technique based on non-overlapping fixed size column is used. The features are classified using SVM. The results indicate that the reported technique can produce very high success rates. Overall accuracy of 95.9%, Specificity of 98.1% and Sensitivity of 66.4% has been obtained. Such a system may be applied to drive computerized machinery in noisy environments, and can be used for the rehabilitation of speech impaired people.

Rodomagoulakis [71] investigated some problems in the field of audio-visual automatic speech recognition (AV-ASR) concerning visual feature extraction and audio-visual integration. Color based detection and template matching strategies are used to detect and track the mouth region, which is considered as the region of interest (ROI) through sequential time frames. Subsequently, discrete cosine transform (DCT) is used to transform pixel values in "compact", descriptive features. The author prove that some factors related with ROI detection, like the inclination of the speaker's head as well as its size, affect the performance of both visual and audio-visual recognizers. In order to counter these effects, the author investigated some methods for rotation correction and scaling normalization. The improved visual front-end schema yielded a word recognition error (WER) reduction of 95% over a baseline implementation. On the other hand, the author investigated a new approach for the unsupervised stream weight estimation which is based on K-means clustering. Stream weight behavior and adaptability is tested under a variety of noises, in a word or sentence level, for classification and recognition tasks. Finally, we compare the results for static and adaptive weighting to evaluate their weight estimation approach and moreover, they measured the improvements achieved by using an audio-visual recognizer instead of the traditional audio-only recognizer. All experiments were based in CUAVE AV English language database and furthermore, recognizers were built using HTK.

Kuniaki *et al* [72] proposed an AVSR system based on deep learning architectures for audio and visual feature extraction and a multi-stream HMM (MSHMM) for multimodal feature integration and isolated word recognition for Japanese language. Their experimental results demonstrated that, compared with the original Mel-frequency Cepstral Coefficients (MFCCs), the deep de-noising auto encoder can effectively filter out the effect of noise superimposed on original clean audio inputs and that acquired de-noised audio features attain significant noise robustness in an isolated word recognition task. Furthermore, our visual feature extraction mechanism based on the convolutional neural network(CNN) effectively predicted the phoneme label sequence from the mouth area image sequence, and the acquired visual features attained significant performance improvement in the isolated word recognition task relative to conventional image-based visual features, such as PCA. Finally, an MSHMM was utilized for an AVSR task by integrating the acquired audio and visual features. Their experimental results demonstrated that even with the simple but intuitive multimodal integration

mechanism, it is possible to attain reliable AVSR performance by adaptively switching the information source from audio feature inputs to visual feature inputs depending on the changes in the reliability of the different signal inputs. Although automatic selection of stream weight was not attained, their experimental results demonstrated the advantage of utilizing an MSHMM as an AVSR mechanism. The next major target of their work is to examine the possibility of applying our current approach to develop practical, real-world applications. Specifically, future work will include a study to evaluate how the VSR approach utilizing translation, rotation, or scaling invariant visual features acquired by the CNN contributes to robust speech recognition performance in a real-world environment, where dynamic changes such as reverberation, illumination, and facial orientation, occur.

3.4 Summary

The previous works for Amharic language are audio only speech recognition. These traditional acoustic based speech processing systems have attained a high level of performance in recent years, but the performance of these systems is heavily dependent on a match between training and test conditions. In the presence of mismatched conditions (i.e., acoustic noise) the performance of acoustic speech processing applications can degrade markedly. The visual speech modality is independent to most possible degradations in the acoustic modality. This independence, along with the bimodal nature of speech, naturally allows the visual speech modality to act in a complementary capacity to the acoustic speech modality. It is hoped that the integration of these two speech modalities will aid in the creation of more robust and effective speech processing applications in the future. In this research, we model Amharic speech recognizer by considering both audio and visual.

Chapter Four: Design of Audio-Visual Amharic Speech Recognition

4.1 Introduction

This chapter presents detailed techniques and ways employed to design and model the audio-visual speech recognition for isolated word and phone (vowels) of Amharic language. The system architecture describes the overall design of the system and all the components of the system architecture are discussed in detail.

4.2 System Architecture

Figure 4.1 depicts the overall system architecture developed in this work. The system architecture shows how these components interact to accomplish the recognition process. The system starts by acquiring the audio speech signal of a speaker through a microphone as well as the video frames of the speaker's face by means of a camera. The audio and visual streams are then ready for analysis at a signal level.

In contrast to audio-only speech recognition systems, where only the audio stream of information is available, here there are two streams of speech information, these are audio stream and the video stream. In this architecture, audio and video signals are separated and independently processed to extract relevant features. Once the relevant information is picked from each signal; they are fused together and then used for an improved speech recognition system. The system implementation consists of three main stages these are design of the front-end processing system for both audio and visual, integration of audio and visual vectors and the training of the recognizer.

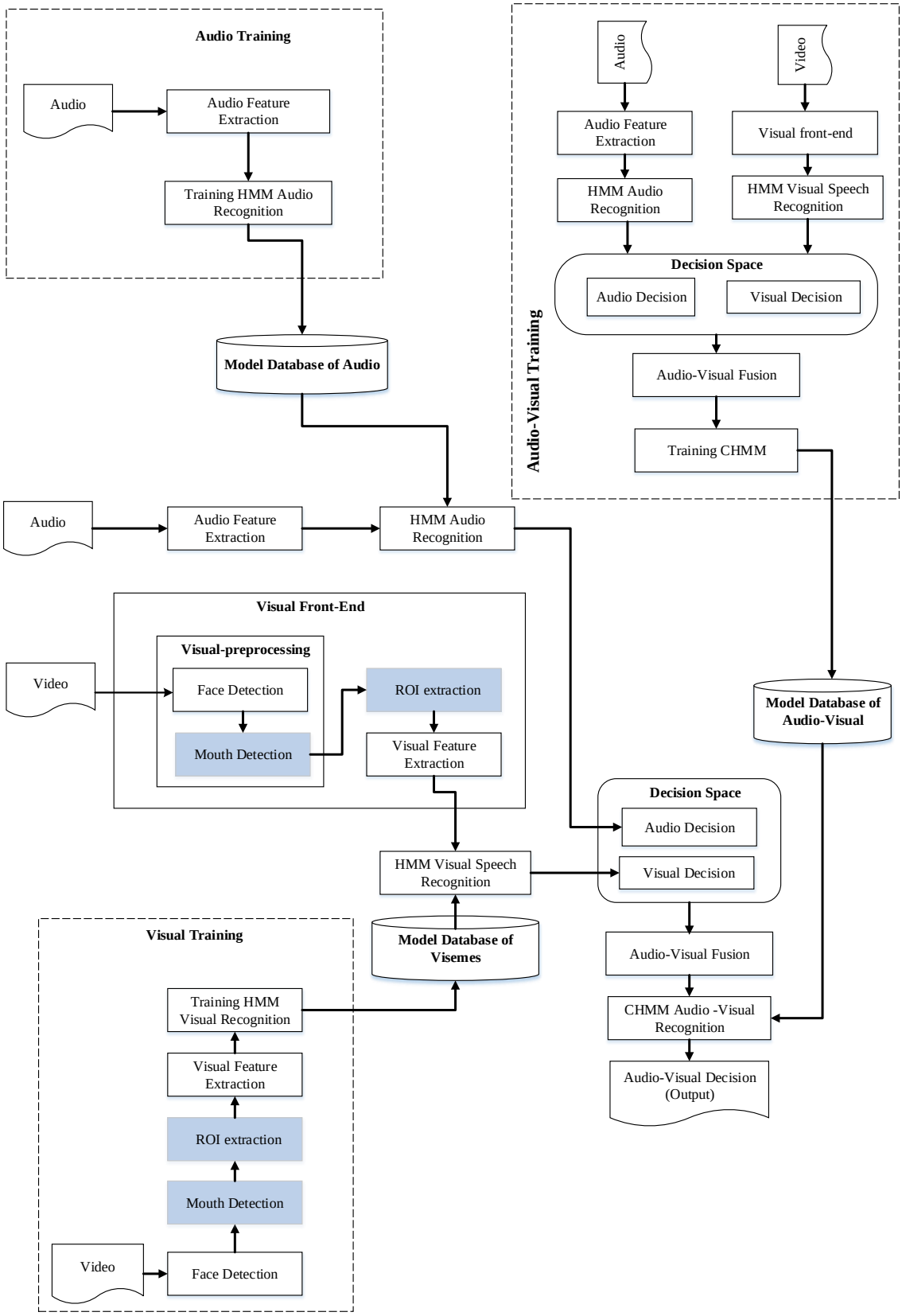


Figure 4.1: *System Architecture*

4.2.1 Visual Front-End Component

As the videos of speakers contain information not related to the speech itself, such as the identity and background, the visual front-end needs to remove this superfluous information leaving only that related to speech. The mouth region of the speakers is identified and a region of interest (ROI) is isolated prior to the extraction of visual speech features. Front-end processing transforms speech into a parameter vector suitable for subsequent processing and consists of the preprocessing of video sources, followed by feature extraction.

The visual front-end identifies the portion of the speaker's face following the mouth that contains the most speech information and extracts that information in a parametric form suitable for processing by the recognizer. The front-end component can be divided into three sub-tasks: visual preprocessing, region of interest (ROI) extraction and feature extraction. Though often considered separately, the three tasks are largely interdependent.

4.2.1.1 Visual Preprocessing

Before being applied to the recognizer for training or recognition purposes, visual streams need to be preprocessed to remove data irrelevant to speech and to enhance certain characteristics that help to improve speech recognition performance.

The first step of visual preprocessing is face detection followed by mouth detection and ROI extraction. The image which acquired is by the camera is RGB image. Before applying visual preprocessing on the input frame image the image should be change to the grey-level image.

Gray-level images are referred to as monochrome, or one-color image. They contain brightness information only. The typical image contains 8 bit/ pixel (data, which allows us to have (0-255) different brightness (gray) levels. The 8 bit representation is typically due to the fact that the byte, which corresponds to 8-bit of data, is the standard small unit in the world of digital computer.

Face and Mouth Detection

Following the conversion the RGB image to gray-levels image the face and mouth detection are performed. The method adopted for the purpose of face and mouth detection is the Viola-Jones object recognizer that uses rectangular haar features [73] and is applied to each frame image. To select specific Haar features, the AdaBoost [74] technique is used to train a weak

classifier. Single strong object classifiers can then be formed by cascading such weak classifiers as shown in Figure 4.2 to detect the face. The advantage of having weak classifiers operating in cascade is that early processing can isolate regions likely to contain objects, thereby allowing greater concentration of effort to be brought to bear on these regions in subsequent operations. Also note that an accelerated computation can be achieved by adopting integral images in order to reduce multiplicative operations to those involving only addition and subtraction. This technique was chosen because of its high detection accuracy and ability to minimize computation time [75]. The approach is applied in two stages, first to obtain the face region and secondly the mouth region was found from the lower half of the face in which it is assumed the mouth is located as shown in Figure 4.3. For more information see AdaBoost algorithm at Appendix B.

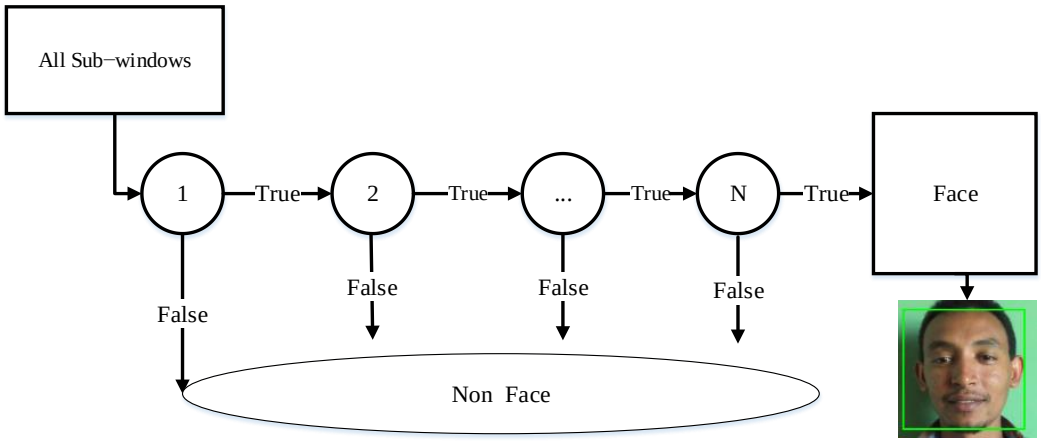


Figure 4.2: *Results of Face Detection Using Viola-Jones Object Recognizer*

After detecting the face then we divide the face into upper-face and lower-face to simplify the next process. When applying mouth detection on the full face, the probability of detection of false mouth become high. Thus, to reduce the detection of the false mouth we divide the detected face into two parts as illustrated the Figure 4.3.

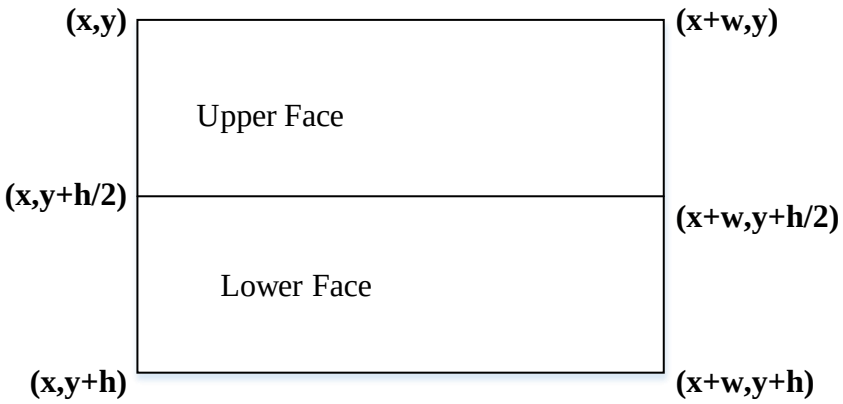


Figure 4.3: *Coordinates of Detected Face*

From the detected face, we got the bounding box coordinate of the face. As shown in Figure 4.4, h= height of bounding box of the face, w = width of bounding box of the face, (x, y) = left-top the coordinate of bounding box of the face, (x+w, y) = right-top coordinate of the face, (x, y+h) = left-bottom coordinate of bounding box of the face and (x+w, y+h) = right-bottom coordinate of bounding box of the face. Figure 4.4 shows detection of mouth process and Algorithm 4.1 shows the processes of face and mouth detection.

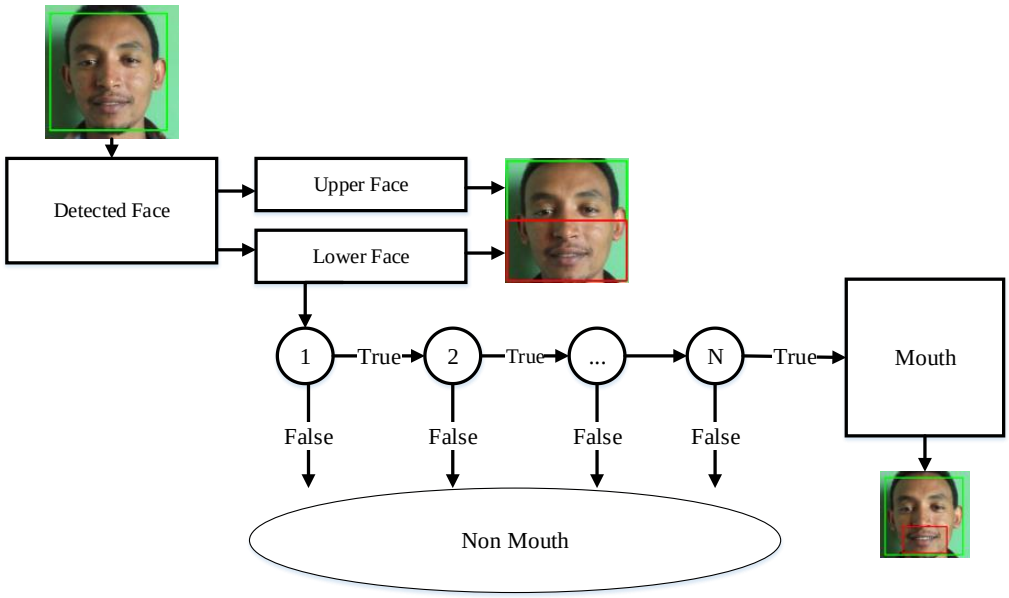


Figure 4.4: *Results of Mouth Detection Using Viola-Jones Object Recognizer*

```
Get frame I
Normalized input frame size
Convert RGB image to GRAY image
Apply cascade Classifier to detect the Face
    For each (x, y, w , h) in faces:
    Find the coordinate of detected face
        Divide the face into two part
        Apply cascade Classifier to on lower face part detect
        the mouth
            For each (mx, my, mw , mh) in mouth:
            Find the coordinate of detected mouth
    show the detected face
    Show the detected mouth
Release the camera
Destroy all windows
```

Algorithm 4.1: *Detecting Face, Upper Face, Lower Face, and Mouth*

4.2.1.2 Region of Interest (ROI) Extraction

The ROI provides the raw input data for visual feature extraction and thus the overall performance of an audio-visual automatic speech recognition (AVASR) system is greatly influenced by the accurate extraction of ROI. The identification of the ROI is made more difficult due to the high deformation of lip shape, as well as the variation in the content of the mouth region due to the presence or absence of tongue, teeth, and opening and closing of mouth during speech. ROI detection approaches are also often influenced by variations in lighting conditions and changes in the pose and orientation of the speakers. The presence or absence of a beard or moustache also affect ROI extraction.

```
If(mouth Detected):
     Get mouth bounding box coordinate
     Get the coordinate of center of image
     Resize the image from  center of image
     Extract the Reign of Interest
     Save the image of ROI
Else:
     Print mouth is not detected in this image
```

Algorithm 4.2: *Pseudo-code for Extraction of ROI*

After identification of speakers mouth region the next stage is the extraction of the ROI as Shown in Figure 4.5. For appearance based feature approaches a bounding box around the lower half of the face containing the mouth region is extracted as desired ROI. In our case, we use bounding box and with the size based on the detected mouth size and resize this bonding box to ROI size based on the center of the detected mouth as shown in the Figure 4.6 and 4.7 the coordinates of bounding box are selected in such a way that it contains the desired ROI in all the frames of utterance. This is used to create uniform image size for every utterances.

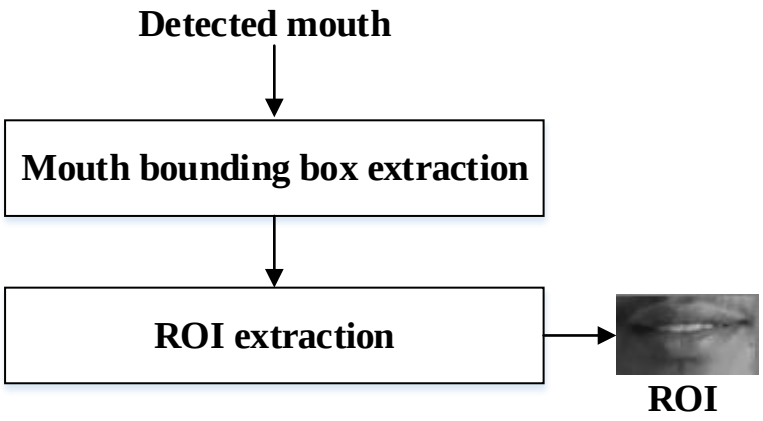


Figure 4.5: *ROI Extraction form Single Frame*

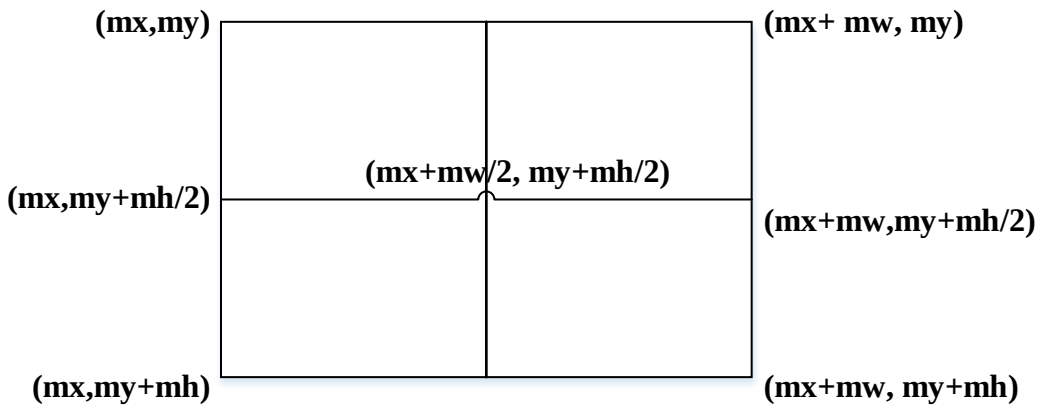


Figure 4.6: *Coordinate of Detected Mouth and Center of Bounding Box*

As shown in Figure 4.6, mh is the height of bounding box of the detected mouth, mw is the width of bounding box of the detected mouth, (mx, my) is the left-top coordinate of bounding box of the detected mouth, (mx + mw, my) is the right-top coordinate of bounding box of the detected mouth, (mx, my+ mh) is the left-bottom coordinate of the detected mouth and (mx + mw, my + mh) is the right-bottom coordinate of the bounding box of the detected mouth. Therefore, the ROI coordinate is as shown in Figure 4.7 relative to the center of bounding box of the detected mouth (mx+mw/2, my+mh/2).

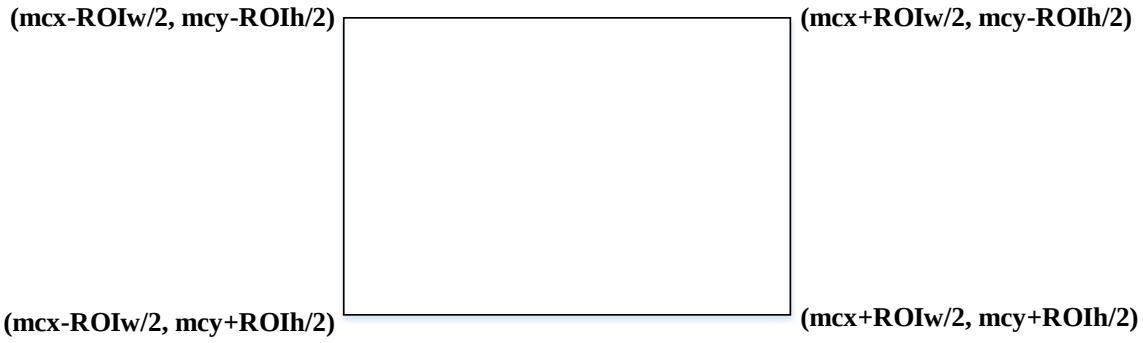


Figure 4.7: *Coordinate of Region of Interest (ROI)*

As discussed in Figure 4.8 the center of bounding box of the detected mouth is (mx+mw/2, my+mh/2). This is considered as the center coordinate of ROI to resize the bounding box of the detected mouth into a uniform size of ROI whose value is assigned to the (mcx, mcy) coordinate. Where mcx is the coordinate x value of the center of ROI, mcy is the coordinate of y value of ROI, ROIh is the height of the region of interest, and ROIw is the width of region of interest. Algorithm 4.2 shows allover ROI extraction process.

4.2.1.3 Visual Feature Extraction

The purpose of feature extraction is to retain as much speech related information as possible from the original images of the speaker in a reasonably small number of parameters. In visual feature extraction, a range of transformation techniques, such as discrete cosine transform (DCT), discrete wavelet transform (DWT), principal components analysis (PCA) and linear discreminent analysis (LDA) are used.

Once we extracted the mouth region (the region of interest), there is a wide choice of algorithms that can be applied to extract visual features form the ROI. The most commonly used transforms in appearance-based feature extraction approaches for AVASR research are the DCT and the DWT. According to Lee *et al* [76], the DWT has many advantages over the DCT. First, DCT difference coding is computationally expensive. Second, wavelets do not cause blocking artifacts. Thus, for this work we used appearance-based feature extraction method called DWT. In this step the extracted ROI image is used as an input for feature extraction. For more information see the implementation in Appendix F.

The DWT transform decomposes the input image into a low-frequency sub band (known as the approximate image) and high-frequency sub-bands (known as detailed images), as shown in Figure 4.8. The LL region of the DWT transform in Figure 4.8 contains the low frequency contents of the image, the HL region contains the high-frequency horizontal details, LH the high-frequency vertical details and HH the high-frequency details for both the horizontal and vertical direction. The application of the DWT to an image results in high-pass and low-pass filtering of the image. Further refined details of an image can be extracted by applying higher levels of decomposition. This is achieved by the application of DWT to the sub-images obtained in the lower level, starting from the original input image. First-level decomposition means the DWT of the original image; second-level decomposition means the DWT of sub-images obtained in the first level and so on, whereas the low frequency components are known as approximate coefficients while the high frequency components are known as detailed coefficients.

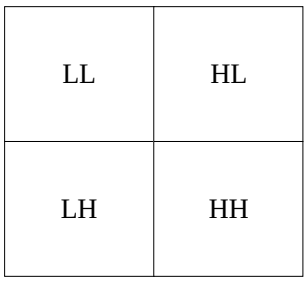


Figure 4.8: *Single Level DWT Decomposition of an Image.*

Generally, two dimensional wavelet transformation is applied on the image of ROI which results in four sub images, as shown in Figure 4.8, as average image(LL) and three detail images (HL, LH and HH). For the purpose of image classification, the three detail images are discarded and the average sub image is converted into a vector by concatenating the columns. This vector is used as image representation for the purpose of image classification. The average image which is in the form of vectors as ROI image feature, is reduced to 30 dimensions by applying LDA. Figure 4.9 shows the process of DWT on ROI image.

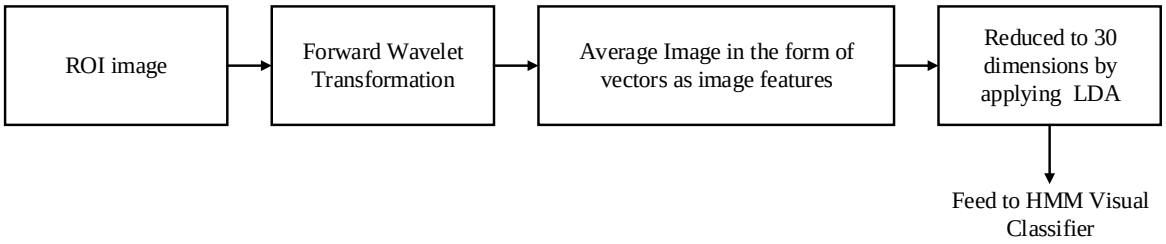


Figure 4.9: *DWT and HHM Visual Classifier*

Figure 4.9 shows the reconstruction of the mouth region (isolated ROI of single frame) of speakers from low and high frequency coefficients of the DWT. In Figures 4.11, images (a) and (b) are reconstruction from low frequency coefficients, while (c) and (d) are reconstructions from high frequency coefficients. In Figure 4.11, images (a) and (b) are reconstructions from the 2^{nd} level and 3^{rd} level approximate coefficients of DWT decomposition, while (c) and (d) are reconstructions from the remaining detailed coefficients. These image reconstructions suggest that while the overall subjective appearance of the image is well retained in low frequency coefficients, the edges of the mouth are better preserved in detailed coefficients, and hence the use of these coefficients could potentially be useful for AVASR purposes.

.

(a) Approximate coefficients at level 2

(b) Approximate coefficients at level 3

(c) Detailed coefficients at level 2

(d) Detailed coefficients at level 3

Figure 4.10: *Image Reconstructions from DWT Coefficients*

Visual Speech Modeling

Prior to the visual feature extraction stage, visual speech modeling is required. This issue is very important to the design of audio-visual fusion. The basic unit of speech in the visual space is viseme. The concept of viseme is usually defined in accordance with the mouth shape and mouth movements. To represent a viseme, one should develop a method for representing the video sequence, further complicating the video processing stage. Fortunately, most of the visemes can be represented by stationary mouth images. The benefit of using such a representation is that it can be mapped directly to the acoustic speech units, which makes the integration process pretty easy. Therefore, for this thesis, we use visemes as visual speech units. For each phone and word we used 30-40 sequence of images.

To be able to design the visual front-end, it is desirable to define for each phoneme its corresponding viseme. This enables us to integrate the visual speech recognition system into existing acoustic-only systems. Unfortunately, speech production involves invisible articulatory organs, which renders the mapping of phonemes to visemes into many-to-one. Consequently, there are phonemes that cannot be distinguished in the visual domain. For

example, the phonemes ፕ [pe], ብ [be], ጵ [pe’], and ም [me] are all produced with a closed mouth and cannot be distinguished visually one phoneme from the other phonemes, so they will be represented by the same viseme. It is important also to consider the effect of the dual of the allophone, where the same viseme can be realized differently in the visual domain due to the speaker variability and the context. To our best knowledge, unlike the phonemes, there is no viseme set that is commonly used by all researchers for Amharic language.

For notational convenience, we shall identify the visemes by the names of the phonemes they represent. As our focus is on oral movement (place of articulation), we shall refer to the movement of mouth when voicing a particular phoneme as a viseme.

The clustering of the different mouth images into viseme classes is done based on the place of articulation and the manner of articulation manually on the base of visual similarity of these images. Accordingly, we obtain the viseme classes and the phoneme-to-viseme mapping for Amharic consonant are in Table 4.1 and phoneme-to-viseme mapping for Amharic vowels in Table 4.2. Table 4.3 shows sample sequences of image for Amharic vowels.

Table 4.1: *Viseme Classes and the Phoneme-to-Viseme Mapping for Amharic Consonant*

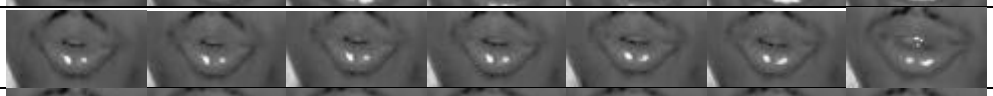
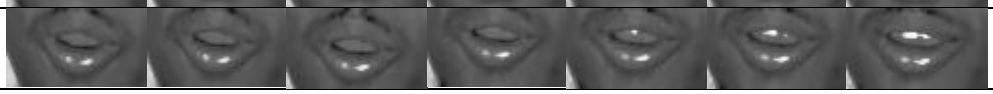
Viseme Number	Group of phoneme	Place of Articulation	Manner of Articulation	Mouth description / Viseme description
0	None	-	-	(silence and relax)
1	ብ[b],ፕ[p],ም[m], ጵ[p’]	Bilabial	Stops	• These are sounds produced when the lips are brought together. • The nature of mouth: lips together.
2	ዉ[w]	Bilabial	Glides	• Small-rounded open mouth state.
3	ቭ[v],ፍ[f]	Labiodental	Fricatives	• Almost closed mouth state; upper teeth visible; lower lip moved inside. • Lower lip is raised towards the upper front teeth.
4	ድ[d],ጥ[t’],ት[t]	Alveolar	Stops	• Medium open, not rounded, mouth state; teeth fully visible, tongue partially visible.
5	ዝ[z], ስ[s] ,ጽ[s’]	Alveolar	Fricatives	• Medium open, not rounded mouth state, teeth fully visible but the tongue is not visible.
6	ን[n]	Alveolar	Nasals	• Medium open, not rounded, mouth state; teeth visible.
7	ል[l]	Alveolar	Liquids	• Tip of tongue behind open teeth, gaps on sides.
8	ዥ[ž],ሽ[š],	Palatal	Fricatives	• The upper and the lower teeth closed together • longitudinal open mouth state
9	ጅ[j],ች[c], ጭ [c’]	Palatal	Affricates	• The upper and the lower teeth closed together • open mouth state
10	ኝ[N]	Palatal	Nasals	• The upper and the lower teeth closed together with little gap between. • open mouth state
11	ር[r]	Palatal	Liquids	• Open mouth state, the tip of tongue close to the upper teeth.
12	ይ[y]	Palatal	Glides	• Open mouth state, the upper and the lower teeth closed together with little opening.

14	ግ[g],ክ[k],ቅ[k’]	Velar	Stops	• Slightly open mouth with mostly closed teeth
15	ጓ[$g^{wl,}$ኳ[k^{w}]·ቋ[k'^{w}]	Labiovelar	Stops	• Started with round lip state, with small open and lip pic, and end with wiled mouth open.
16	ዕ [?] ,ሀ[h] ,ኋ[h^{w}]	Glottal	Fricatives	• The lip is static but slightly open to pass the internal air to outside.

Table 4.2: *Viseme Classes and the Phoneme-to-Viseme Mapping for Amharic Vowels*

Visemes Number	**Group of phoneme**	**Mouth description / Viseme description**
1	ኢ [i], ኤ [e]	• For ኢ [i], ᴏpen mouth state, the middle of tongue at the lower teeth. • For ኤ [e], Wild open mouth state, visible tongue and the tip of the tongue at lower teeth.
2	ኦ[o], ኡ[u],	• Round lip, with small open, and the lip pic
3	እ[I], ኧ[A], ኣ [a]	• Longitudinal open mouth state; tongue visible.

Table 4.3: *Sample Viseme Image for Amharic Vowels*

Phoneme	Image sequence example (from the new database)
አ[a]	
ኡ[u]	
ኢ [i]	
ኧ[A]	
ኤ [e]	
እ[I]	
ኦ[o]	

4.2.2 HMM Visual Speech Recognition

The last processing stage of visual speech (lip-reading) is feature classification. For the classification, process the HMM is used due to its popularity that has followed from many successful applications in the statistical modeling of audible speech and packages availability in python programing for implementation HMM (e.g., hmmlearn, scikit-learn).

Building HMM Visual Classifier

At this stage, it is crucial to define the basic structure of the HMM developed for viseme-based visual speech recognition, so that we can understand the general framework in which our proposed visual features will be integrated.

Training the HMM is one of the basic tasks of the recognition process. Now, each visual word or visual phone is represented with a sequence of symbols. For a single training video, we have a symbol vector. In training HMM, the basic inputs are the output sequence (which is the symbol matrix in our case), the initial transition and emission matrices. Before the preparation of the initial transition and emission matrices deciding the type of the model and number of states is a very critical task.

Depending on the type of the problem, various model types and number of states can be selected. In this work, two different types of HMM architectures were used based on word and phone models. As shown in Figure 4.11, for each *viseme (phone)* in the database, a 3 states HMM was designed, and the output (most likely) visemes sequences is recognized as a phone by means of HMM. As shown in Figure 4.12, for each *word* in the database, an HMM with a different number of states is designed, the number of the states being the number of the visemes (phone) appearing in a specific word.

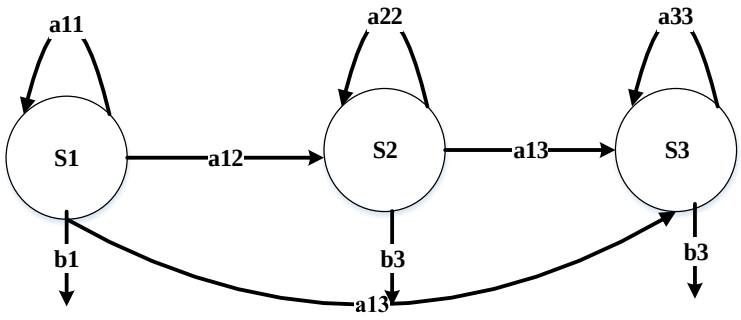


Figure 4.11: *A Three State HMM Topology for Phone*

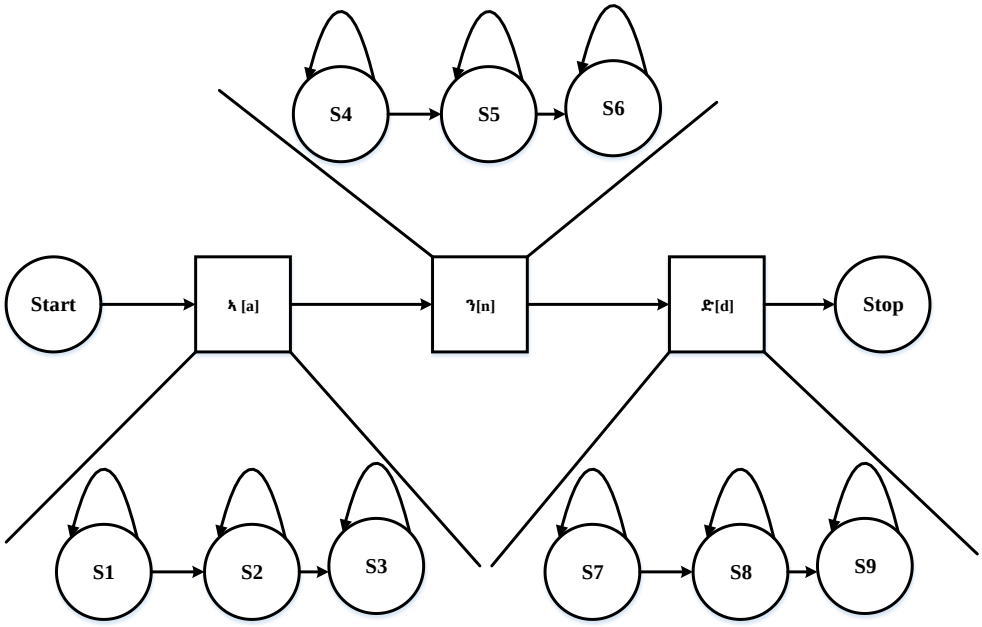


Figure 4.12: *HMM Topology for the Word One [ANID]/ ኣ ን ድ*

To train the HMM, in addition to the observed symbol sequence and the decision of the number of states, we need initial state transition and emission matrices. We will prepare these matrices with random numbers. The state transition matrix is an NXN matrix where N is the number of states. The emission matrix is an NXM matrix where N is the number of states and M is the size of the observable symbols which is found empirically during the vector quantization process.

Now we can start training our model and the final HMM parameters will be produced by iterative process. We used the Baum Welch algorithm [77] to train the HMM. The algorithm updates the parameters of the HMM iteratively until convergence following the procedure below.

Finally, we will have our HMM for a given visual word or visual phone. The HMM model will be represented by the state transition and emission matrices produced after training. These matrices will be retained in the database with their respective visual word or visual phone id to use them later for recognition process.

Visual Speech Recognition

Recognition is the process of finding the most probable HMM from a set of HMMs (which were produced during the training phase) that can produce a given observed sequence. For this process we used the forward and backward algorithms to compute the likelihood that a model produced for a given observation sequence.

The likelihood can be effectively calculated using dynamic programming by forward algorithm which reproduce the observation through HMM and backward algorithm which back trace the observation through HMM.

The same steps will be followed as the training part to have a set of feature vectors that represent the frames and vector quantization process which enabled us to substitute a sequence of feature vector values to a set of distinct symbols. But, for testing the recognition system we used different data from the ones that we used for training purpose.

A given sequence of images for a visemes of word or phone (Amharic vowels phone) will be checked against each trained HMM and the model with a largest probability will be selected.

4.2.3 Audio feature extraction

Audio-only ASR solutions, perhaps due to their relative maturity, have generally settled on the use of a single set of feature types, namely the Mel-frequency Cepstral Coefficients (MFCC). In this study, we use MFCC for audio feature extraction. Before applying feature extraction the audio file type change to wav file. To implement audio feature extraction we use python package called librosa. The sample code used during the implementation of this component is given in Appendix I.

4.2.4 Audio-Visual Fusion

The main difference between audio-only and audio-visual ASR lies in the design of the front-end, as two input streams (the audio stream and the video stream) are now available. Additionally, at some stage in the recognition process, the streams of information from the audio and visual modalities need to be fused.

During fusion the issue where the fusion of the data takes place should be addressed. As discussed in Section 2.10 of Chapter Two, several architectures have been developed in literature to tackle this issue. Feature fusion integrates data on the feature level, where audio and visual features are used simultaneously and equally to identify the corresponding speech unit, thus, feature-level fusion algorithms train a single classifier on the concatenated vector of audio and visual features. Decision fusion, on the other hand, takes place after the independent identification of each stream and is thus an integration of identification results.

In this study, for audio-visual recognition we used a decision fusion architecture because different comparisons showed superior performance of the decision fusion compared to the other fusion architectures. By using this techniques, we combined the likelihoods of single-modality (audio- only and visual-only) HMM classifier decisions outputs to recognize audio-visual speech. Thus, this isolated word or phone speech recognition, we implemented by calculating the combined likelihood for the acoustic and the visual observation for a given word or phone model.

As shown in Figure 4.13 based on decision fusion architecture in this study, there are two recognizers working independently for the audio and the video/visual channel respectively. The combination is performed at the output of each recognition process. Therefore, for each

word in the model two different probabilities will be provided one for each modality $P(W_j|\ O^A)$ and $P(W_j|\ O^V)$. This is the reason why it is also called decision fusion. The final solution will be the word that maximizes the combined probability $\overset{arg\ max}{W_j}\{\ P(W_j|\ O^{AV})\ \}$.

In order to obtain the two probabilities $P(W_j|\ O^A)$ and $P(W_j|\ O^V)$, *an* acoustic model and a visual model must be found. Each of these independent models will be defined by a set of parameters that will be obtained in two independent training processes one for the audio and for the video.

The identification results in our case are the a posteriori probabilities of the observation vectors. Finally, the audio and visual features are combined and the resulting is used for training and testing. Figure 4.13 shows an overall diagram of our fusion system.

As shown in the Figure 4.13, we use weighted Bayesian fusion to combine the two complementary features (audio and visual), originating from audio and visual modalities, in order to maximize information gather and to overcome the impact of noise in each individual stream.

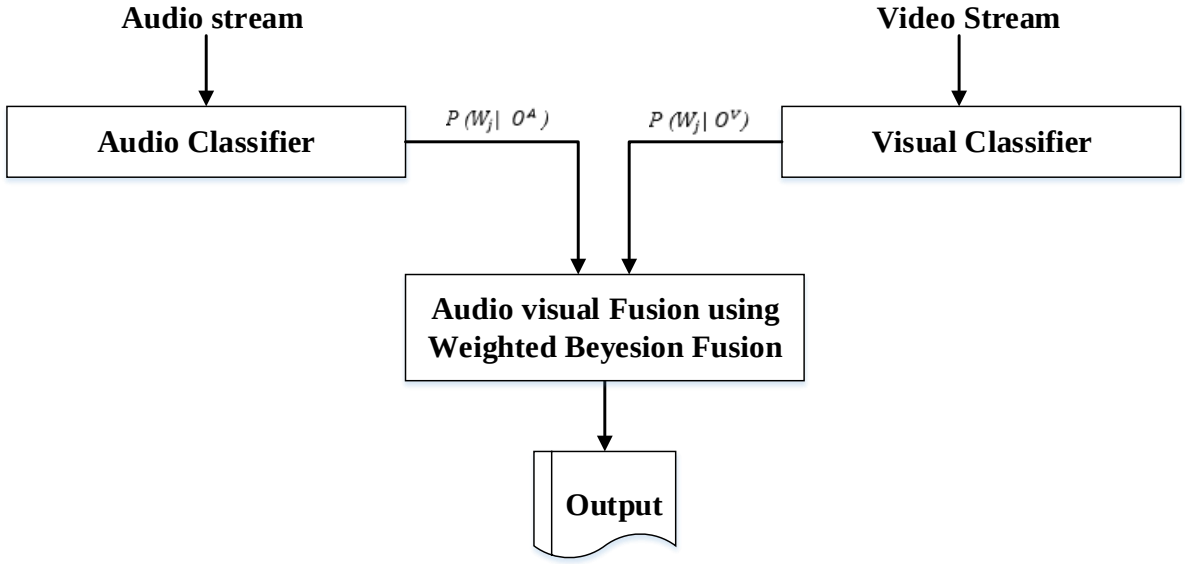


Figure 4.13: *Block Diagram of the Multimodal (Audio-Visual) Fusion.*

4.2.5 Audio-Visual Recognition

After deciding the methods of integration audio and visual, the last step of AVSR system is tanning of the integration features and audio-visual recognition. As discussed in Section 2.9, there are many audio-visual recognition models, such as product HMM, factorial HMM and coupled HMM. In this work, the coupled HMM is used which is the work of Ara *et al* [78]. The CHMM is a generalization of the HMM suitable for a large variety of multimedia applications that integrate two or more streams of data.

In this study, a two-stream CHMM used for our audio-visual speech recognition system. In our experiments, the bimodal speech recognition system employs a two-chain CHMM, with one chain being associated with the acoustic observations, the other with the visual features.

CHMM Training

In this work, the training of the CHMM parameters is performed in two stages. In the first stage, the CHMM parameters are estimated for isolated phoneme-viseme pairs. These parameters are determined first using the Viterbi-based initialization [78], followed by the expectation-maximization (EM) algorithm [42]. In the second stage, the parameters of the CHMMs, estimated individually in the first stage, are refined through the embedded training of all CHMMs. In a way similar to the embedded training for HMMs, each of the models obtained in the first stage are extended with one entry and one exit non- emitting states.

Recognition

The word and phone (vowel) recognition is carried out via the computation of the Viterbi algorithm [79] for the parameters of all the word and phone (vowel) models in the database. The parameters of the CHMM corresponding to each word and phone (vowel) in the database are obtained in the training stage. In the recognition stage, the influence of the audio and visual streams is weighted based on the relative reliability of the audio and visual features for different levels of the acoustic noise.

4.3 Conclusion

In this Chapter, we discussed about the approaches we have selected and why they are selected, we have also introduced the overall design of the proposed system of AVSR for Amharic language, which includes face detection, mouth detection, region of interest (ROI) extraction, visual feature extraction, visual speech classifier, audio-visual integration and audio-visual recognition. Finally, we briefly discussed the functions of each component of AVSR and some algorithms to show the components work.

Chapter Five: Experiment

5.1 Introduction

In this chapter, we discuss the implementation of phone based and isolated word based audio-visual speech recognition system using lip movement for the Amharic language. The tools used, the database, the training and testing processes to implement the prototype will be discussed.

Data corpus can greatly influence the research results obtained. To our best of knowledge there is no visual corpus for Amharic speech reading .we there developed new audio-visual corpus for Amharic speech reading which is called the Amharic audio-visual data corpus (AAVC). To keep data collection size requirements manageable do to time limitation, we focus our study on a small-vocabulary recognition task. It contains Amharic digits from 'zero' to 'nine', days of the week, months name and Amharic vowels. A brief introduction to the new data corpus is given in Section 5.4 and the contents of the data corpus are detailed in Table 5.1 and Table 4.3. The chapter also discusses the evaluation of the system developed prototype.

5.2 Data Collection

5.2.1 Subject Population

The data corpus contains the utterances of 17 speakers, 3 females and 14 males. The participants were undergraduate students at Addis Ababa Science and Technology University and two others. The participants were asked to read the word in their own natural style and with no instruction regarding pronunciation. To protect the identity of the participants, each of them have been given a special code to represent them in the data corpus, for example, first participant identity is 'Vs1m', second is 'Vs2m' and so on. The word 'v' represents AAVC while the word 'm' represents gender, 'm' for male or 'f' for female and 's1' represents participant 'speaker 1'.

5.2.2 Collected Data

Data was collected in two sessions within two different environments. For both sessions, the environments had normal background. The input video was captured with a laptop webcam

camera which has frame rate of 30 frames per second. The audio is recorded at a sampling rate of 32 kHz and 16 bits quantization. The position of the camera is stationed in front of the speaker and made static. This position of the camera enabled us to get face part of the speakers properly which is necessary to have meaningful lip-reading. This position has equal distance of the speakers from the camera which enabled us to get equivalent size of the speakers in the video. The speaker did not wear eyeglasses that helped us to facilitate face detection process.

Since we are working on isolated word and phone based audio-visual speech recognition using lip-reading, we made the starting and ending position of the lip uniform. The starting and ending position of the lip is silent mode or closed mouth state. We captured a separate video for each phone and word and on average the videos have 1 to 2.5 seconds length.

After capturing the video, the first task was removing video parts which are not necessary for the training as well as the recognition process. These parts of the video are usually found at the beginning and end of the videos. We cropped these parts manually since they contain frames which may affect the results of the recognition. We also convert the video format to avi video format. To do this, we used HD video convertor Factory video tool. A HD video convertor was used to extract audio information from the recorded video file, generating a standard quality stereo audio file at a sampling rate of 32 kHz and at 16-bit resolution. This quality is sufficient for speech recognition as human speech production bandwidth is generally between 100Hz and 8 kHz.

As discussed above in the data collection 17 speakers were involved. The data consists of two separate parts. The first part is a list of isolated words which contains the digits ‘ዜሮ (zero)’ to ‘ዘጠኝ (nine)’, name of the days and name of months. In the second part is list of phones which contains Amharic vowel. A list of the data recorded in the AAVC data corpus is given in Table 4.3 in Chapter four which contains Amharic vowel and the other in Table 5.1.

Each speaker is expected to read words or phones 10 times. For both audios and videos, we used six of them for training purpose and the rest four for testing purpose. Therefore, for a specific word or phone we have 102 training data (audio and videos) and 68 testing data (audio and videos). In general, in the AAVC data corpus we have a total of 5100 videos and 5100 audios for all 30 words and 1190 total videos and 1190 audios for all 7 vowels. In total we AAVC data corpus has 6290 videos and 6290 audios.

Before the audio and video can be used for recognition purposes, it must be correctly labelled according to the utterances made. Ids are given for each word/ phone and speakers. In addition to this, we give sequence numbers for videos of a word /phone. Therefore, we organized the captured videos with a speaker id, word id and a sequence number of the video id. Table 5.1 shows the words that we collect and Table 4.3 shows the vowels.

Table 5.1: *Sample Collected Words*

ID	Word	Meaning
1	ሠኞ[SEGNO]	Monday
2	ማክሰኞ[MAKSEGNO]	Tuesday
3	እሮብ[EROB]	Wednesday
4	ሀሙስ[HEMUS]	Thursday
5	አርብ[ARB]	Friday
6	ቅዳሜ[KDAME]	Saturday
7	እሁድ [EHUD]	Sunday
8	ዜሮ[ZERO]	Zero
9	አንድ[AND]	One
10	ሁለት[HULET]	Two
11	ሶስት[SOST]	Three
12	አራት[ARAT]	Four
13	አምስት[AMST]	Five
14	ስድስት[SIDST]	Six
15	ሰባት[SEBAT]	Seven
16	ስምንት[SMINT]	Eight
17	ዘጠኝ[ZETEGN]	Nine
18	መስከረም[MESKEREM]	September
19	ጥቅምት[TIKMT]	October
20	ህዳር [HIDAR]	November
21	ታህሳስ[TAHSSAS]	December
22	ጥር [TIR]	January
23	የካቲት[YEKATIT]	February
24	መጋቢት[MEGABIT]	March
25	ሚያዝያ [MIYAZIYA]	April
26	ግንቦት [GNBOT]	May
27	ሰኔ [SENE]	June
28	ሐምሌ[HEMLE]	July
29	ነሐሴ [NEHESE]	August

30	ጳጉሜ[PAGUME]	Pagume

5.3 Implementation

5.3.1 Tools and Programing language

The following tools are used to develop the prototype of audio-visual Amharic speech recognition using lip movement.

Conda: While it's possible to install Python directly from python.org, we recommend using conda instead. Conda is a package manager similar to pip. Conda allow you to install python and non-python library dependencies but with pip you can install only python dependencies. In general conda is a cross-platform and Python-agnostic package manager and environment manager program that quickly installs, runs and updates packages and their dependencies and easily creates, saves, loads and switches between environments on your local computer. Conda is included in all versions of Anaconda, Miniconda and Anaconda Repository [79].

- **Miniconda:** Miniconda includes conda, Python and a small number of other useful packages including pip, zlib and a few others [79].
- **Anaconda:** Anaconda includes everything in Miniconda and a stable collection of over 150 standard open source packages for data analysis and scientific computing that have all been tested to work well together, including scipy, numpy and many others. These packages can all be installed automatically with one quick and convenient installation routine [79].

Based on the above discussion we chose Anaconda version 4.3.21 for our implementation. For more information how to managing packages using conda command see appendix A.

Database: The database is prepared to store the label name for each extracted visemes (visual speech). SQLiteStudio [80] database is chosen to implement the database since it has very important features suitable for the development of various applications. The database contain three tables one for visual speech, one for audio speech and one for audio-visual speech as shown in Appendix F. Majority of machine learning algorithms work with numbers, so you can transform the categorical values and string into numbers.

OpenCV (Open Source Computer Vision): is a library of programming functions mainly aimed at real-time computer vision [81]. For implementation we use OpenCV version 3.1.0.

PyAudioAnalysis: an open-source Python library that provides a wide range of audio analysis procedures including: feature extraction, classification of audio signals, supervised and unsupervised segmentation and content visualization [82].

PyWavelets: PyWavelets is free and Open Source wavelet transform software for the Python programming language. It combines a simple high level interface with low level C and Cython performance. We use this because PyWavelets has 1D, 2D and n-D forward and Inverse Discrete Wavelet Transform (DWT and IDWT) features. We use this package for Visual feature extraction [83].

Spyder's Text Editor: To write a prototype code we used Spyder's text editor. This editor is a multi-language editor with features such as syntax coloring, code analysis (real-time code analysis powered by *pyflakes* and advanced code analysis using *pylint*), introspection capabilities such as code completion, call-tips and go-to-definition features (powered by *rope*), function/class browser, horizontal/vertical splitting features, etc. [84].

Programing language: The prototype is written with python programing language. Python provides a large standard library for image processing and audio processing like scipy and numpy. And many high use programming tasks have already been scripted into the standard library which reduces length of code to be written significantly. Python language is developed under open source license, which makes it free to use and distribute, including for commercial purposes [85].

5.3.2 Preparing Development Environment

The prototype is developed and tested in computer with capacity Core I7, CPU 2.20GHZ and RAM 4GB with hard disc 500GB and Windows 10 64bit operating system.

When we install anaconda, most python packages are installed by default. But, for this work the development environment should have the following dependencies: python packages numpy, scipy, matplotlib, hmmlearn, librosa, scikit-learn, librosa, PyWavelets, and PyAudioAnalysis. To install dependencies see Appendix B.

NumPy is the fundamental package for scientific computing with Python. This is required by some other libraries used, and it is also used for implementation of algorithms like DCT (Discrete Cosine Transform) [86].

SciPy is a Python-based ecosystem of open-source software for mathematics, science, and engineering. We use it for calculation of commonly used mathematical operations like calculating mean and standard deviation of arrays [87].

Matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. We use this library for plotting graphs [88].

Hmmlearn is a set of algorithm for learning and inference of Hidden Markov Models. This library provides the implementations of algorithms like Forward-Backward Algorithm and Baum Welch Algorithm [89].

Librosa is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems. For this work we use this package to extract audio features [90].

Bayesan is a small Python utility to reason about probabilities. It uses a Bayesian system to extract features, crunch belief updates and spew likelihoods back. You can use either the high-level functions to classify instances with supervised learning, or update beliefs manually with the Bayes class [91].

5.3.3 Audio-Visual Speech Recognition Components

As we discussed in Chapter four AVSR the system has nine major components of the system. These components are responsible to conduct major tasks in the recognition process. We discuss them as follows.

Face and Mouth Detection: This component is responsible to input videos from the folder where we put as a training and testing sample videos. The sample code used during the implementation of this component is given in Appendix C. More detail information face and mouth detection sample result from AAVC listed in Appendix E.

Region of Interest Extraction: After identification of speakers mouth/lips region the next stage is the extraction of the ROI. The detected mouth used as the input for ROI process. For this work the size of ROI is set as 40X70. The sample codes used during the implementation of this component is given in Appendix C. Sample ROIs are shown in Figure 5.1.

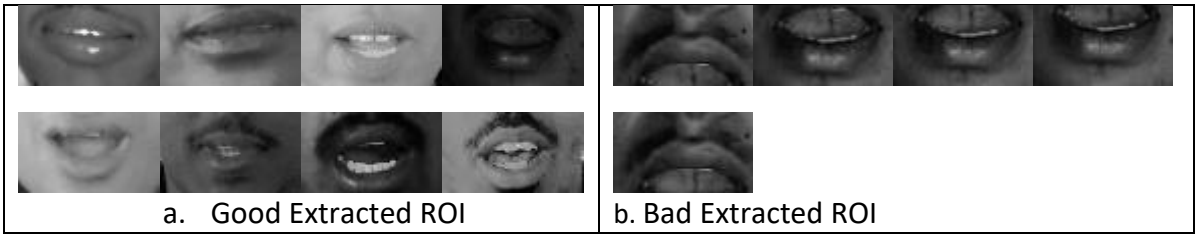


Figure 5.1: *Sample Result of Extracted ROI Different Speakers form AAVC*

Feature Extraction: This component is dedicated to accept the binary images (extracted ROI image) as an input, extract the features and store it in the database. The component stores the features in an organized way to identify the word/phone, the speakers and the video identification number. Extracted feature form ROI image is store as dimensional feature vector as shown in Figure 5.2.

```
grid_x: 8
grid_y: 8
histograms:
   - !!opencv-matrix
      rows: 1
      cols: 16384
      dt: f
      data: [ 3.12500000e-002, 0., 0., 0., 0., 0., 0., 3.12500000e-002,
          0., 0., 0., 0., 0., 0., 3.12500000e-002, 0., 0., 0., 0., 0.,
          0., 0., 0., 0., 0., 0., 0., 0., 1.25000000e-001, 0.,
          3.12500000e-001, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
          0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
          0., 0., 0., 0., 1.87500000e-001, 9.37500000e-002,
          3.12500000e-002, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
```

Figure 5.2: *Sample Result of Extracted feature vectors*

5.4 Experimentation Criteria

Our audio-visual speech recognition system is evaluated using the visual information, audio information and in combination with the audio information. The system evaluation is

conducted to test the performance of the system. This can be seen from the accuracy of the result the system produces. Due to time limitation for evaluation, we chosen only 8 speakers randomly from the 17 speakers. We divided the evaluation process into two parts.

The first part is evaluation based on speakers (which is speaker dependent). This was conducted on each subject alone, all the test examples and the training examples pertaining to the same subject (person). The main goal of this experiment is to test the way of speaking to each speaker and each one's ability to produce a visual signal that was easily read. For word evaluation a total of 120 testing data (4 testing record data and 30 sample word each) for each audio and video independently. For phone/vowels evaluation a total of 28 testing data (4 testing record data and 7 sample phone/ vowels each) for each audio and video independently.

The result of the recognition for a speakers (speaker dependent) is calculated using Equation (37), Equation (38) and Equation (39). This evaluation used for both individual modal audio stream and video stream and a combination of the two modal.

$$\text{VR ratio of speaker dependent} = \frac{Number\ of\ recognized\ of\ speakers}{Number\ of\ test\ word\ of\ speakers} \quad (37)$$

$$\text{AR ratio of speaker dependent} = \frac{Number\ of\ recognized\ of\ speakers}{Number\ of\ test\ word\ of\ speakers} \quad (38)$$

$$\text{AVR ratio of speaker dependent} = \frac{\#\ recognized\ of\ speakers}{\#\ test\ word\ of\ speakers} \quad (39)$$

The second part of evaluation process is speaker-independent experiment. In this type of experiment, the computer evaluates a group of persons out of the training set, and each time one person gets out of the training set and is tested against the remaining persons in the group. The training set does not contain any examples belonging to the tested subject (i.e., the training set contains all the subjects from the AAVC data corpus, except the tested subject). Each time, after testing each subject, the current test subject joins the training set, and another subject is removed from the training set and assigned as a new test subject (test set), and so on, until no more subjects need to be tested. The average is then taken to verify the accuracy of the experiment.

In this case, we used all recorded data of the speaker for testing that means for a single speaker for word evaluation we have a total of 300 testing data (10 testing record data for each speaker

and 30 sample word each) for each audio and video independently. For phones/vowels evaluation a total of 70 testing data (10 testing record for each speaker data and 7 sample word each) for each audio and video independently were used. Due to time limitation for this experimentation we chosen only 8 speakers randomly out of 17 speakers form AAVC corpus.

The result of the recognition for a speakers (speaker independent) is calculated using Equation (40), Equation (41) and Equation (42). This evaluation is used for both individual modal audio stream and video stream and a combination of the two modal.

$$\text{VR ratio of speaker independent} = \frac{Number\ of\ recognized\ of\ speakers}{Number\ of\ test\ word\ of\ speakers} \quad (40)$$

$$\text{AR ratio of speaker independent} = \frac{Number\ of\ recognized\ of\ speakers}{Number\ of\ test\ word\ of\ speakers} \quad (41)$$

$$\text{AVR ratio of speaker independent} = \frac{\#\ recognized\ of\ speakers}{\#\ test\ word\ of\ speakers} \quad (42)$$

5.5 Test Results

As discussed in Section 5.4 we have two cases to evaluate the recognition of the system. To calculate the evaluation, we wrote a unit test code, see Appendix G. For the first case, that is, speaker dependent recognition the results are shown in Table 5.2, Table 5.3 and Table 5.4 respectively. For the second case that is speakers independent Table 5.5 Table 5.6 and Table 5.7 show the results.

Table 5.2: *Speakers Dependent Visual Only Speech Recognition Result*

Sample Speakers	**Total Number of Test Visual speech**		**Number of Recognized**		**Recognition Percentage**	
	Words	**Vowels**	**Words**	**Vowels**	**Words**	**Vowels**
Vs1m	120	28	79	19	65.83%	67.85%
Vs2m	120	28	76	20	63.33%	71.42%
Vs3f	120	28	77	23	64.17%	82.14%
Vs4f	120	28	75	22	62.50%	78.57%
Vs5f	120	28	77	21	64.17%	75.00%
Vs6m	120	28	69	20	57.50%	71.43%
Vs7m	120	28	72	23	60.00%	82.21%
Vs8m	120	28	55	12	45.83%	42.86%
Average					**60.42%**	**71.45%**

Table 5.3: *Speakers Dependent Audio Only Speech Recognition Result*

Sample Speakers	Total Number of Test Audio speech		Number of Recognized		Recognition Percentage	
	Words	**Vowels**	**Words**	**Vowels**	**Words**	**Vowels**
Vs1m	120	28	82	21	68.33%	75.00%
Vs2m	120	28	80	20	66.67%	71.42%
Vs3f	120	28	78	23	65.00%	82.14%
Vs4f	120	28	79	22	65.84%	78.57%
Vs5f	120	28	77	22	64.17%	78.57%
Vs6m	120	28	81	23	67.50%	82.14%
Vs7m	120	28	76	20	63.33%	71.42%
Vs8m	120	28	74	20	61.67%	71.42%
Average					**65.31 %**	**76.34%**

Table 5.4: *Speakers Dependent Audio-Visual Speech Recognition Result*

Sample Speakers	Total Number of Test Audio-Visual speech		Number of Recognized		Recognition Percentage	
	Words	**Vowels**	**Words**	**Vowels**	**Words**	**Vowels**
Vs1m	120	28	87	24	72.50%	85.71%
Vs2m	120	28	87	25	72.50%	89.28%
Vs3f	120	28	84	25	70.00%	89.28%
Vs4f	120	28	84	23	70.00%	82.14%
Vs5f	120	28	85	22	70.83%	78.57%
Vs6m	120	28	86	23	71.67%	82.14%
Vs7m	120	28	81	25	67.50%	89.28%
Vs8m	120	28	79	21	65.83%	75.00%
Average					**70.10%**	**83.92%**

Table 5.5: *Speakers Independent Visual Only Speech Recognition Result*

Sample Speakers	Total Number of Test Visual speech		Number of Recognized		Recognition Percentage	
	Words	**Vowels**	**Words**	**Vowels**	**Words**	**Vowels**
Vs1m	300	70	189	47	63.00%	67.14%
Vs2m	300	70	191	45	63.67%	64.28%
Vs3f	300	70	201	49	67.00%	70.00%
Vs4f	300	70	199	55	66.33%	78.57%
Vs5f	300	70	195	51	64.00%	72.86%
Vs6m	300	70	184	44	61.33%	62.86%
Vs7m	300	70	165	47	55.00%	67.14%
Vs8m	300	70	143	43	47.67%	61.43%
Average					**61.00%**	**68.04%**

Table 5.6: *Speakers Independent Audio Only Speech Recognition Result*

Sample Speakers	Total Number of Test Audio speech		Number of Recognized		Recognition Percentage	
	Words	**Vowels**	**Words**	**Vowels**	**Words**	**Vowels**
Vs1m	300	70	193	49	64.33%	70.00%
Vs2m	300	70	191	51	63.67%	72.86%
Vs3f	300	70	204	51	68.00%	72.86%
Vs4f	300	70	200	53	66.68%	75.71%
Vs5f	300	70	196	55	65.33%	78.57%
Vs6m	300	70	190	45	63.33%	64.28%
Vs7m	300	70	171	49	57.00%	70.00%
Vs8m	300	70	180	50	60.00%	71.43%
Average					**63.54%**	**71.96%**

Table 5.7: *Speakers Independent Audio-Visual Speech Recognition Result*

Sample Speakers	Total Number of Test Audio-Visual speech		Number of Recognized		Recognition Percentage	
	Words	**Vowels**	**Words**	**Vowels**	**Words**	**Vowels**
Vs1m	300	70	217	54	72.33%	77.14%
Vs2m	300	70	195	55	65.00%	78.57%
Vs3f	300	70	213	55	71.00%	78.57%
Vs4f	300	70	211	56	70.33%	80.00%
Vs5f	300	70	203	55	67.67%	78.57%
Vs6m	300	70	199	50	66.33%	71.43%
Vs7m	300	70	188	53	62.33%	75.71%
Vs8m	300	70	185	52	61.67%	74.29%
Average					**67.08%**	**76.79%**

5.6 Discussion

The major challenge in this work was data collection. Since there is no available audio-visual data corpus for Amharic language, the researcher had to prepare new data corpus. There was also lack of willing people for the visual data collection.

In this work, two experiments were conducted as part of the pilot study to evaluate the visual words approach. The first was a speaker-dependent (SD), and the second was a speaker-independent (SI) experiment.

The speaker-dependent experiments also show that females provide better visual speech recognition than males while talking, and the average word recognition rate for female

speakers is (63.61% for word recognition and 78.57% for vowels), which is significantly higher than it is for male speakers (58.5% for word recognition and 67.15% for vowels recognition). This could be due to several factors, the main one being the clear appearance of the female lip area (the lack of facial hair on female faces) and the use of makeup. This allows for more lip-detection accuracy, and more detail appears on the facial features. This contrasts with the appearance of facial hair in males, where some details are hidden by this obstacle. For example, as shown in Figure 5.4, the moustache of Male-8 hides his upper and that is probably why the word recognition rate of Male-8 (Vs8m) is the worst (42.86% for vowels recognition and 45.82% for visual word recognition).

Figure 5.3: *Male have Moustache*

As shown in Table 5.2 and Table 5.5, there are visual speech result variation between speakers. In visual speech, this happened because of the way the speakers follow to read the word/phone. Figure 5.5 shows visual speechless person (VSP) for speakers who do not provide visual signals while talking, or at least provide incomplete visual signals, causing the lip-reading process to misunderstand the speech.

Figure 5.4: *Samples of Visual Speechless Person*

The other issue that we encounter during experimentation is the wrong recognition result for those word or phone (vowels) which have similarity. This problem is encountered due to the reasons the basic units of visual speech are the visemes, the most conspicuous component of the visual speech is the oral movement. However, many basic sounds have the same sequence of movement of the lips. The mapping between phonemes and visemes is not one-to-one but many-to-one. For example, phonemes ፕ [pe], ብ [be], ጵ [pe’], and ም [me] are all produced

with a closed mouth and cannot be distinguished visually one phoneme form the other phonemes. The vowels ኦ[o] and ኡ[u] also cannot distinguished visually one from the other.

Generally, in this work we used two main experimental sets these are SD and SI. The result found for audio-visual speech recognition on the first set was 70.10% on word recognition, 83.92% on vowels recognition and the result found for audio-visual speech recognition on the second set was 67.08% on word recognition and 76.79% on vowels recognition. In both cases (SD and SI), recognition audio-visual speech is better than the audio only recognition and visual only recognition.

Chapter Six: Conclusion and Future Work

6.1 Conclusion

Audio is used as principal source of speech information in automatic speech recognition systems, but their performance degrades in presence of noise. Not only this, some phones are acoustically ambiguous. To compensate, a number of approaches have been adopted in the ASR literature, of which the use of the visually modality is probably the most suitable candidate being supported by both human speech perception studies and the work reported on AVSR systems.

The purpose of this study is to develop an automatic audio-visual speech recognition for Amharic language using the lip movement which include face and lip detection, region of interest (ROI), visual features extraction, visual speech recognition and integration of visual with audio. The architecture of the system that we adopted in our study is the decision fusion architecture. As a result of this architecture, we used three classifiers. The first one is the HMM classifier for audio only speech recognition, the second one is HMM classifier for visual only speech recognition and the third one was CHHM for audio-visual integration.

For implementation we use python programing language and OpenCV. For face and mouth detection we use Viola-Jones object recognizer called haarcascade face detection and haarcascade mouth detection respectively, after the mouth detection ROI extracted. Extracted ROI used as an input for visual feature extraction. Appearance-based (low-level) DWT is used for visual feature extraction and LDA is used for reduce visual feature vector. For audio feature extraction we use MFCC this is implemented by using a python package called librosa.

The system has been tested using the videos and audios which were captured for testing and training purpose. We used two main evaluation criteria for both phone (vowels) and word recognition, these are speakers dependent and speakers' independent. Based on the first evaluation criteria (speaker dependent) we found overall word recognition 60.42% on visual only, 65.31% on audio only and 70.1% for audio-visual. We also found overall vowels (phone) recognition 71.45% on visual only, 76.34% on audio only and 83.92 % on audio-visual speech based on speakers' dependent evaluation criteria.

Based on the second evaluation criteria called speaker independent we got overall word recognition 61% for visual only, 63.544% for audio only and 67.08 % for audio-visual. We also found the overall vowels (phone) recognition 68.04% for visual only, 71.96% for audio only and 76.79 % for audio-visual speech.

6.2 Contribution to Knowledge

This research work has contributed the following:

- Extended ASR architecture for Amharic speech recognition.
- ROI method from detected mouth for visual speech recognition.
- Mouth detection method.
- Visual Speech Modeling for Amharic.
- Base data corpus called AAVC for Amharic Audio-visual speech recognition.

6.3 Future Works

This work shows that audio-visual speech recognition is a wide area that has been explored by many researchers. In this thesis work, we achieved a result in recognizing isolated word and phones of Amharic language. However, there are gaps which should be filled by future works. Since this research cannot be used as a full fledge speech recognition system of Amharic language, we recommend future works to incorporate the following components.

- We used a static background while we capture videos of the speakers. But in real world we cannot find such static background in every corner. Therefore, future Amharic lip-reading researches should consider dynamic backgrounds.
- Researches should be expanded to the recognition of continuous speech by using the outputs of the isolated word and phone recognitions.
- In this work, we used frontal face for visual speech recognition, we recommend that future researches to consider side face to recognize visual speech.
- The AAVC corpus contains 18 speakers and more should be included to improve the statistical reliability of results obtained. In addition, only three female speakers are currently in the database and more need to be added in order to allow the detection of characteristics that may differ between the two sexes and so potentially consider the development of separate models in order to improve the overall performance. The

AAVC data corpus could also be extended to include sentences that cover additional parts of the phone space that involve different phone contexts.

- The current visual feature extraction is still dependent on the light conditions. It would be desirable to obtain an extraction algorithm that would be independent of these conditions in such a way that the training and test could be mismatched without result degradations. As well another visual extraction algorithm could extract more information about the visemes and, therefore, reduce the word error rate of visual only system improving the performance of the whole system.
- The results achieved from audio-visual is small compared with other previous literature for other language thus, more work is needed on the integration of visual speech and audio speech.
- More experiments on using the whole face is needed. Instead of using only the mouth region: Some papers state that the face expression may give a more clear idea about the speech, like the jaw motion, we thing this point deserves a research.
- Although the shape of the mouth and the positions of visible articulators in individual frames of video provide useful information about the utterance, they fail to capture the speech dynamic information necessary for distinguishing certain phonemes. As speech is inherently a dynamic phenomenon, the motions of the various articulators is likely to add additional information which may not be captured by features extracted from individual frames. For instance, the position of the tongue appears similar when uttering ል [l] or ድ [d], and can only be differentiated by observing the motion of the tongue during the articulation. While the mouth shape provides information for recognizing a set of visemes, the mapping from phoneme to viseme is not one-to-one and several phonemes may correspond to a single viseme. Such phonemes can often be differentiated by utilizing dynamic information obtained from the lips and other visible articulators. Consequently, a suitable representation of the motions of the articulators may potentially improve the overall recognition performance of the AVASR systems.

References

[1] C. Vimala, V. Radha, "A Review of Speech Recognition Challenges and Approaches", *Journal of World of Computer Science and Information Technology (WCSIT)*, Vol. 2, No. 1, pp. 1-7, 2012.

[2] Alin G and Leon J. M., "Visual speech recognition: automatic system for Lip reading of Dutch" *collaborative information system ICIS project supported by the Dutch Ministry of Economic Affairs*, BSIK03024, Mar 2009.

[3] Guoying Zhao, Mark Barnard and Matti Pietik¨ainen, "Lip-reading with Local Spatiotemporal Descriptors*", IEEE Journal of Transactions on Multimedia,* Vol. 11, No. 7, November 2009.

[4] Ayaz A. Shaikh, Dinesh K. Kumar, Wai C. Yau, and M. Z. Che Azemin, "Lip Reading using Optical Flow and Support Vector Machines", in *Proceedings of 3rd International Congress on Image and Signal Processing*, Yantai, China, 2010.

[5] Peter Roach, "English Phonetics and Phonology a practical course", Fourth edition, Cambridge university press, New York, 2009.

[6] http://clas.mq.edu.au/speech/phonetics/phonetics/consonants/place.html, Last accessed on 10/31/2017.

[7] P.Ladefoged, *A Course in Phonetics*, Harcourt Brace Jovanovich College Publishers, Third edition, 1993.

[8] Sharma R., Pavlovic V., Huang T. S., "Toward Multimodal Human-Computer Interface", in *Proceeding of IEEE*, Vol. 86, No. 5, pp. 853-869, 1998.

[9] Namrata Dave and Narendra M. Patel," Phoneme and Viseme based Approach for Lip Synchronization" ,*International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 7, No. 3, pp.385-394, 2014.

[10] Solomon Berhanu, "Isolated Amharic Consonant-Vowel (CV) Syllable Recognition. An experiment using the Hidden Markov Model", Unpublished Masters Thesis, Department of Computer Science, Addis Ababa University, 2001.

[11] Kinfe Tadesse, "Sub-word based Amharic speech recognizer: An experiment using Hidden Markov Model (HMM)," MSc Thesis, School of Information Studies for Africa, Addis Ababa University, Ethiopia, June 2002.

[12] Martha Yifru. “Automatic Amharic Speech Recognition System to Command and Control Computers”, Masters Thesis, School of Information Studies for Africa, Addis Ababa, 2003.

[13] Molalegn Girmaw. “An Automatic Speech Recognition System for Amharic”, Masters Thesis, Department of Signals, Sensors and Systems, Stockholm, Sweden: Royal Institute of Technology, 2004.

[14] Solomon Teferra, “Syllable-Based Speech Recognition for Amharic”, in *Proceedings of the 5th Workshop on Important Unresolved Matters, Association for Computational Linguistics: Prague,* Czech Republic. pp. 33–40, 2007.

[15] S. Tamura, H. Ninomiya, N. Kitaoka, S. Osuga, Y. Iribe, K. Takeda, and S. Hayamizu, “Audio-Visual Speech Recognition Using Deep Bottleneck Features and High-Performance Lipreading” in *Proceedings of APSIPA Annual Summit and Conference*, Asia-Pacific, December 2015.

[16] A. Davis, M. Rubinstein, N. Wadhwa, and William T. Freeman, “The Visual Microphone: Passive Recovery of Sound from Video*”, Journal of ACM Transactions on Graphics (TOG),* Vol. 33, No. 4, July 2014.

[17] Vibhanshu Gupta, and Sharmila Sengupta, “Automatic speech reading by oral motion tracking for user authentication system”, *International Journal of Software Engineering Research & Practices*, Vol. 3, No.1, April, 2013.

[18] Ming-Hsuan Yang, David J. Kriegman and Narendra Ahuja, “Detecting Faces in Images: A Survey”, *Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 1, pp. 34-58, January 2002.

[19] Solomon Teferra, “Automatic Speech Recognition for Amharic”, PhD Thesis, der Universität, Hamburg, 2006.

[20] Baye Yimam, *አጭርና ቀላል የአማርኛ ሰዋሰው (ačirna qälal yä’amrña säwasw/Short and Simple Amharic Grammar),* Addis Ababa: Alpha, 2010.

[21] Getahun Amare, *ዘመናዊ የአማርኛ ሰዋሰው በቀላል አቀራረብ (Zemenäwi yä’amrña säwasw beqäläl akärärb/Modern Amharic Grammar Simple presentation*), Addis Ababa: Alpha , 2016.

[22] Simon Luce, "Audio-visual Speech Processing", Published PhD Thesis, School of Electrical & Electronic Systems Engineering, Queensland University of Technology, Brisbane, April 2002.

[23] Hussien Seid and B. Gambäck, "A Speaker Independent Continuous Speech Recognizer for Amharic", Published Masters Thesis, Computer Science & Information Technology, Arba Minch University, 2005.

[24] Timothy J. Hazen, "Visual model structures and synchrony constraints for audiovisual speech recognition", *Journal of IEEE Transactions on Audio, Speech and Language Processing*, Vol. 14, No. 3, pp. 1082-1089, 1 May 2006.

[25] Stéphane Dupont and Juergen Luettin, "Audio-Visual Speech Modeling for Continuous Speech Recognition", *IEEE Journal of Transactions on Multimedia*, Vol. 2, No. 3, September 2000.

[26] Paul Duchnowski, Martin Hunke, Dietrich Busching, Uwe Meier and Uwe Meier, "Toward movement-Invariant Automatic Lip-Reading and Speech Recognition", in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Detroit, Michigan, USA, May 08-12, 1995.

[27] T. Chen and R. Rao, "Audio-visual integration in multimodal communication", in *Proceedings of the IEEE*, Vol. 86, No. 5, pp. 837–852, May 1998.

[28] G. Potamianos, H. P. Graf, and E. Cosatto, "An Image Transform Approach for HMM Based Automatic Lip-reading" in *Proceeding of International Conference on Image Processing*, Vol. 3, pp. 173-177, 1998.

[29] Alan Wee-Chung Liew and Shilin Wang, *Visual Speech Recognition: Lip Segmentation and Mapping*, Medical Information science reference, Hershey, New York, 2009.

[30] Jesús Fernando Guitarte Pérez , "Improvements in Speech Recognition for Embedded Devices by taking Advantage of Lip Reading Techniques", PhD Thesis, Departamento De Ingeniería Electrónica y Comunicaciones, Universidad De zaragoza, March 2006.

[31] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Journal of Transactions on Acoustics, Speech, and Signal Processing* ,Vol. 28, No. 4, pp 357-366, Aug 1980.

[32] B. S. Atal and S. L. Hanauer, “Speech Analysis and Synthesis by Linear Prediction of the Speech Wave,” *The Journal of the Acoustical Society of America,* Vol. 50, pp. 637–655, 1971.

[33] Farooq, O., and Datta, S., “Speech recognition with emphasis on wavelet based feature extraction”, *IETE Journal of Research*, Vol. 48, No. 1, pp. 3-13, 2002.

[34] Junda Dong, “Designing a Visual Front-End in Audio-Visual Automatic Speech Recognition System”, Published Master Thesis, Faculty of California Polytechnic State University, USA, June 2015.

[35] S. Gurbuz, Z. Tufekci , E. Patterson, and J. N. Gowdy, “Application of affine-invariant Fourier descriptors to lip-reading for audio-visual speech recognition”, Proc. ICASSP, pp. 177-180, 2001.

[36] S. Dupont, and J. Luettin, “Audio-Visual Speech Modeling for Continuous Speech Recognition”, in *Proceeding of IEEE Transactions on Multimedia*, Vol. 2, No. 3, 2000.

[37] J.W. Picone, “Signal modelling techniques in speech recognition”, in *Proceedings of the IEEE,* Vol. 81, No. 9, pp. 1215-1247, 1993.

[38] P.Viola, M. Jones, “Robust Real-Time Face Detection”, *International Journal of Computer Vision* ,Vol. 57, No. 2, pp.137-154, 2004.

[39] J. Kamarainen and V.kyrki, “Invariance Properties of Gabor Filter-Based Features-Overview and Applications”, Journ*al of IEEE Transaction on Image Processing*, Vol.15, No. 5,pp. 1088-1099, May 2006.

[40] Ivana Arsic and Jean-Philippe Thiran, “Mutual Information Eigenlips for Audio-Visual Speech Recognition”, in *Proceedings of 14th European Signal Processing Conference (EUSIPCO),* Florence, Italy, September 4-8, 2006.

[41] A. B. Hassanat, “Visual Words for Automatic Lip-Reading”, Published PhD Thesis, Department of Applied Computing University of Buckingham, United Kingdom, December 2009.

[42] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, “A survey of skin-color modeling and detection methods”, *Journal of Pattern Recognition*, Vol. 40, No. 3, pp. 1106 –1122, Mar. 2007.

[43] Syed Ali Khayam , *The Discrete Cosine Transform (DCT): Theory and Application*, Michigan State University, March 10, 2003.

[44] Zhanyu Ma and Leijon A. , "A Probabilistic Principal Component Analysis Based Hidden Markov Model for Audio-Visual Speech Recognition", in *Proceedings of 42nd Asilomar Conference on Signals, Systems & Computers*, Pacific Grove, CA, USA, 26-29 Oct. 2008.

[45] L. Smith, "A tutorial on Principal Components Analysis", *Cornell University, USA*, February 26, 2002.

[46] Yu, H., and Yang, J., "A direct LDA algorithm for high-dimensional data with application to face recognition", *Journal of Pattern Recognition Society,* Vol. 34, pp. 2067-2070, 2001.

[47] J. Shlens. "A Tutorial on Principle Component Analysis", Systems Neurobiology Laboratory, University of California at San Diego, Version 2, December 2005.

[48] E. D. Petajan, "Automatic lip-reading to enhance speech recognition", in *Proceedings of IEEE Computer Vision and Pattern Recognition,* pp. 44-47, 1985.

[49] Nasir Ahmad, "A Motion Based Approach for Audio-Visual Automatic Speech Recognition", Published PhD Thesis, Department of Electronic and Electrical Engineering Loughborough University, United Kingdom, May 2011.

[50] Ole Helvig Jensen, "Implementing the Viola-Jones Face Detection Algorithm", Published Master's thesis, Image Analysis and Computer Graphics, Department of Informatics and Mathematical Modeling, Technical University of Denmark, 2008.

[51] M. Heckmann, F. Berthommier, and K. Kroschel, "Noise Adaptive Stream Weighting in Audio-Visual Speech Recognition," *Journal of EURASIP* Applied Signal Processing, Vol. 2002, No. 11, pp. 1260–1273, 2002.

[52] I. Matthews, T. Cootes, J. Bangham, S. Cox, and R. Harvey, "Extraction of Visual Features for Lipreading", *Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 2, No. 24, pp.779-789, 2002.

[53] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision,* Vol. 1, No. 4, pp. 321–331, Jan. 1988.

[54] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *Journal of* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23,No .6, pp 681 - 685, Jun 2001.

[55] E. Petajan , B. Bischoff ,and D. Bodoff, "Automatic lipreading to enhance speech recognition," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Washington, D.C., USA , May 15 - 19, 1988.

[56] C. Bregler and Y. Konig. "Eigenlips for robust speech recognition," in *Proceeding of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-94*, Adelaide, SA, Australia, 19-22 April 1994.

[57] G. Krone, B. Talle, A. Wichert, and G. Palm. "Neural architectures for sensor fusion in speech recognition," in *Proceeding of ESCA Workshop on Audio-Visual Speech Processing (AVSP'97),* Rhodes, Greece September 26-27, 1997.

[58] M. Gordan, C. Kotropoulos, and I. Pitas. "A support vector machine-based dynamic network for visual speech recognition applications," *EURASIP Journal on Applied Signal Processing*, Vol., 11, pp., 1248-1259, 2002.

[59] F.V. Jensen, *Introduction to Bayesian networks*. Springer -Verlag New York, Inc. Secaucus, NJ, USA, 1996.

[60] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book, Cambridge,* Entropic Ltd., 2002.

[61] J. Luettin, G. Potamianos, C. Neti, and A.S. AG, "Asynchronous stream modeling for large vocabulary audio-visual speech recognition," *in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'01),* Salt Lake City, UT, USA, 7-11 May 2001.

[62] C. Neti, G. Potamianos, J. Luettin, I. Matthews, H. Glotin, D. Vergyri, J. Sison, A. Mashari, and J. Zhou, "Large-vocabulary audio-visual speech recognition: a summary of the Johns Hopkins Summer 2000 Workshop," in *Proceeding of IEEE Fourth Workshop on Multimedia Signal Processing*, Cannes, France , 3-5 Oct. 200.

[63] A.V. Nefian, L. Liang, X. Pi, X. Liu, and K. Murphy, "Dynamic Bayesian networks for audio-visual speech recognition," *EURASIP Journal on Applied Signal Processing*, Vol.,2002, No.,1, pp., 1274-1288, January 2002.

[64] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden Markov models for complex action recognition," in *Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, USA, 17-19 June 1997.

[65] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A.W. Senior, "Recent advances in the automatic recognition of audio-visual speech," in *Proceedings of the IEEE,* Vol., 91, No.,9, Sept. 2003.

[66] U. Meier, R. Stiefelhagen, J. Yang, and A. Waibel, "Towards unrestricted lip reading," *International Journal of Pattern Recognition and Artificial Intelligence,* Vol., 14, No., 5, pp., 571-585, 2000.

[67] G. Potamianos, J. Luettin, and I. Matthews, "Audio-Visual Automatic Speech Recognition: An Overview," *Issues in Visual and Audio-Visual Speech Processing,* MIT Press, Ch 10, 2004.

[68] R. Goecke, "Audio-Video Automatic Speech Recognition: An Example of Improved Performance through Multimodal Sensor Input," in *Proceeding of NICTA-HCSNet Multimodal User Interaction Workshop,* Vol., 5, pp., 25-32 Sydney, Australia, 2005.

[69] Mustapha A. Makkook, "A Multimodal Sensor Fusion Architecture for Audio-Visual Speech Recognition", Published Master's Thesis, Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada, 2007.

[70] Asratu Aemiro, " Pronunciation Modeling Based on Pattern Identification of Amharic Phonemes for Automatic Speech Recognition" Msc Thesis, Department of Computer Science Addis Ababa University, Ethiopia, 2015.

[71] Rodomagoulakis Isidoros, "Feature extraction optimization and stream weight estimation in Audio-visual speech recognition", A Thesis presented for the degree of Electronic and Computer Engineer, University of Crete, Chania, Greece, October, 2008.

[72] Kuniaki Noda, Yuki Yamaguchi, Kazuhiro Nakadai, Hiroshi G. Okuno and Tetsuya Ogata, "Audio-visual speech recognition using deep learning*"*, Springer Science Business Media, New York, 2014.

[73] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proceedings International Conference on Image Processing 2002 (ICIP2002),* Rochester, New York, USA ,2002.

[74] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, Vol. 55, No. 1, pp. 119–139, Aug. 1997.

[75] C. Zhang and Z. Zhang, "A Survey of Recent Advances in Face Detection," *Microsoft Research*, Rep. MSR-TR-2010-66, 2010.

[76] H. Lee, Y. Kim, A. Rowberg, and E. Riskin, "Statistical Distributions of DCT Coefficients and their Application to an Inter frame Compression Algorithm for 3-D Medical Images," *IEEE Journal of Transactions of Medical Imaging*, Vol. 12, No.3 , pp. 478-485, 1993.

[77] Baum LE, Petrie T, Soules G, Weiss N. "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The annals of mathematical statistics,* Vol.41 No. 1, pp. 164-171, Feb. 1, 1970.

[78] D.W. Massaro and D.G. Stork, "Speech recognition and sensory integration," *Journal of American Scientist,* Vol. 86, No. 3, pp 236-244, 1998.

[79] https://conda.io/docs/faq.html, Last accessed on 4/29/2017.

[80] https://sqlitestudio.pl/index.rvt, Last Accessed on 4/29/2017.

[81] http://opencv.org/ , Last accessed on 4/11/2017.

[82] https://github.com/tyiannak/pyAudioAnalysis/, Last accessed on 4/12/2017.

[83] https://pywavelets.readthedocs.io/en/latest/, Last accessed on 4/12/2017.

[84] https://pythonhosted.org/spyder/editor.html, Last accessed on 4/11/2017.

[85] https://docs.python.org/3/tutorial/, Last accessed on 4/10/2017.

[86] http://www.numpy.org/ , Last accessed on 4/13/2017.

[87] https://www.scipy.org/ , Last accessed on 4/13/2017.

[88] https://matplotlib.org/ , Last accessed on 4/13/2017.

[89] https://hmmlearn.readthedocs.io/en/latest/, Last accessed on 4/13/2017.

[90] https://librosa.github.io/librosa/ , Last accessed on 4/13/2017.

[91] https://github.com/wavelets/bayesian, Last accessed on 4/13/2017.

Appendix A: Python Packages Management to Prepare Development Environment

Installing packages

1. How can we install a specific package, such as SciPy?

```
conda install scipy
```

2. How can we install a package such as SciPy, in a specific version?

```
conda install scipy=0.15.0
```

3. How can we install more than one package at once, such as SciPy and *cURL*)?

```
conda install scipy curl
```

4. How can we install many packages at once and specify the version of the package?

```
conda install scipy=0.15.0 curl=7.26.0
```

5. How can we install a package for a specific Python version?

```
conda install scipy=0.15.0 curl=7.26.0 -n py34_env
```

Updating packages

1. How can we update *conda* itself?

```
conda update conda
```

2. How do we update the *Anaconda* meta package?

```
conda update conda
conda update anaconda
```

3. How can we update a specific package, such as SciPy?

```
conda update scipy
```

Removing packages

1. How can we remove a specific package, such as SciPy?

```
conda remove scipy
```

2. How can we remove multiple packages at once, for example, SciPy and *cURL*?

```
conda remove scipy curl
```

Appendix B: AdaBoost Algorithm

This section describes an algorithm for constructing a cascade of classifiers which achieves increased detection performance while radically reducing computation time. The key insight is that smaller, and therefore more efficient, boosted classifiers can be constructed which reject many of the negative sub-windows while detecting almost all positive instances (i.e., the threshold of a boosted classifier can be adjusted so that the false negative rate is close to zero). Simpler classifiers are used to reject the majority of sub-windows before more complex classifiers are called upon to achieve low false positive rates.

Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.

Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.

For $t = 1, \ldots, T$:

Normalize the weights,

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$

so that w_t is a probability distribution.

For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$.

Choose the classifier, h_t, with the lowest error ϵ_t.

Update the weights:

$$w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$$

where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha_t = \log \frac{1}{\beta_t}$

Appendix C: Visual Preprocessing

```
# This code extract visual features
# First it detect the face
# Then it divided the face in to two parts i.e, the upper and the lower
face.
# Then using the lower face part it detect the mouth
# Finally it extract the ROI

"""
This Code is a prototype for AAVSRS(Amharic Audio Visual Speech
Recognition System)
"""
"""
In code, is the preprocess part of the visual and extract ROI.
"""

import cv2
import numpy as np
import sqlite3
face_cascade=cv2.CascadeClassifier('haarcascades/haarcascade_frontalface_d
efault.xml')
```

```
mouth_cascade=
cv2.CascadeClassifier('haarcascades/haarcascade_mcs_mouth.xml')
ROIh= 40; # regine of interest height
ROIw= 70; #regine of interest width
newx=300;
newy =350; # set the size for frame size
sampleNum=0;# the number of sample

def insert_or_update(lableId,speakerId,lableName):

    conn=sqlite3.connect("../VD/Visual_speech_db.db")
    cmd= "SELECT * FROM visual_speech WHERE
    lableId="+str(lableId)+" AND
    speakerId="+str(speakerId)
    cont=conn.execute(cmd)
    isRecordExist=0;
    for row in cont:
        print(row)
        isRecordExist=1
        ID=row[1]
        if(isRecordExist==1):
            cmdd="UPDATE visual_speech SET
            lableId="+str(lableId)+","+"speakerId="
            +str(speakerId)+","+"lableName='"
            +str(lableName)+"' WHERE id="+ str(ID)
        else:
             cmdd ="INSERT INTO visual_speech
            (speakerId,lableId,lableName) values
            (" + "'"+str(speakerId)+"'"+","+"'"+str(lableId)
             +"'"+","+"'"+str(lableName)+"'"+")"
        #cmdd ="INSERT INTO visual_speech(speakerId,lableId,lableName)
         values ("+str(speakerId)+","+str(lableId)+","+str(lableName)+")"
    conn.execute(cmdd)
    conn.commit()
    conn.close()

Spekers=input('Enter the speakers ID ')
Id=input('Enter The lable ID of the Visem  this must be number ')
Lablename=input('Enter the lable Name ')


while str(Id).isdigit()=='False':
    Id=int(input('Enter The lable this must be number'))

insert_or_update(Id,Spekers,Lablename)


#video_path=input('Enter the vidio path that you want to add your dataset
')
cap= cv2.VideoCapture('../AAVC/VC/Viwel/tranning_data/speaker1/1.avi')
while True:
    ret , img =cap.read()
    #cv2.imwrite("out.jpg", img)
    #print (cv2.arcLength(cnts[0],True))
    if(not ret):
        print ('Process completed')
        break ;
```

```
            #break ;
        else:
            img = cv2.resize(img,(newx,newy))
            gray = cv2.cvtColor(img,cv2.COLOR_RGB2GRAY)
            gray=cv2.equalizeHist(gray)
            faces = face_cascade.detectMultiScale(gray,1.3,5)
            for (x,y,w,h) in faces:
                # make rectange for the hole face
                #cv2.imwrite(""+Id +'.'+ Spekers +'.'+ str(sampleNum) +
".jpg", img)
                cv2.rectangle(img, (x,y), (x+w,y+h),(255,0,0),2)
                # make rectangle for upper  face
                #cv2.rectangle(img, (x,y), (x+w,y+(h//2)),(255,0,0),2)
                sampleNum=sampleNum+1;
                #make rectangle on lwer face part
                cv2.rectangle(img, (x,y+(h//2+20)), (x+w,y+h),(0,0,255),2)
                # convert the lower face from RGB to gray
                lower_face_gray=gray[y+(h//2+15):y+h,x:x+w]
                lower_face_color= img[y+(h//2+15):y+h,x:x+w]
                mouth= mouth_cascade.detectMultiScale(lower_face_gray,1.3,8)
                for (mx,my,mw,mh) in mouth:
                    # get the cencer co-ordinate of the mouth
                    Mcx= mx + (mw//2)
                    Mcy= my + (mh//2)
                    ROI= lower_face_gray[Mcy-(ROIh//2):Mcy+(ROIh//2),
                    Mcx-(ROIw//2):Mcx+(ROIw//2)]
                    ROIc= lower_face_color[Mcy-(ROIh//2):Mcy+(ROIh//2),
                    Mcx-(ROIw//2):Mcx+(ROIw//2)]
                    Mouth= lower_face_color[Mcy-(ROIh//2):Mcy+(ROIh//2),
                    Mcx-(ROIw//2):Mcx+(ROIw//2)]
                    cv2.rectangle(lower_face_color,(mx,my)
                    ,(mx+mw,my+mh),(255,0,0), 1)
                    #res = cv2.resize(img,None,fx=2, fy=2,
                    interpolation = cv2.INTER_CUBIC)
                    #dataSet holds preprocessed data this used
                    to ferrther extraction
                    #this hold a secunce of image
                    cv2.imwrite("Visemes/viwels/"+str(Id)+'.'+
                    str(Spekers +'.'+ str(sampleNum) + ".jpg", ROI)
            ROI = cv2.resize(ROI,(160,90))
            ROIc = cv2.resize(ROIc,(160,90))
            cv2.imshow('Detected Face and Mouth',img)
            cv2.imshow('Regine of Interest',ROI)
            cv2.imshow('ROI3',ROIc)
            k=cv2.waitKey(30) & 0xff
            if k==27:
                break
cap.release()
cv2.destroyAllWindows()
```

Appendix D: Visual Speech Recognizer

```
import cv2
import numpy as np
import sqlite3
import codecs
detector1=
cv2.CascadeClassifier('haarcascades/haarcascade_frontalface_default.xml')
detector= cv2.CascadeClassifier('haarcascades/haarcascade_mcs_mouth.xml')
cap = cv2.VideoCapture('../AAVC/VC/Viwel/tranning_data/speaker1/4.avi')
ROIh= 40;
ROIw= 70;
rec =cv2.createRecognizer()
rec.load('TrainedValue/trainngDataforvawels.yml')
id=0
fontFace = cv2.FONT_HERSHEY_SIMPLEX
fontScale = 1
fontColor = (255, 255, 255)
lst=[];
value=""

def get_predict_value(lableId):
    conn=sqlite3.connect("../VD/Visual_speech_db.db")
    cmd= "SELECT * FROM visual_speech WHERE lableId="+str(lableId)
    cont=conn.execute(cmd)
    resualt=None;
    for row in cont:
        resualt=row
    conn.close()
    return resualt;
while(True):
    ret, img = cap.read()
    if(not ret):
        if(id==0):
            print ('Silet Mode')
        else:
            print 'Predicted as=',max(lst,key=lst.count)
        print ('Process completed')
        break ;
    else :
        newx,newy = 300,350;
        img = cv2.resize(img,(newx,newy))
        gray = cv2.cvtColor(img,cv2.COLOR_RGB2GRAY)
        gray=cv2.equalizeHist(gray)
        #cv2.imshow('gray',gray)

        faces = detector1.detectMultiScale(gray, 1.3, 5)
        #if(len(faces)!=0):
        for (x,y,w,h) in faces:
            cv2.rectangle(img, (x,y), (x+w,y+h),(0,0,0),2)
            # make rectange for the hole face

            #cv2.rectangle(img, (x,y), (x+w,y+h/2),(255,0,0),2)
            # make rectangle for upper  face
            cv2.rectangle(img, (x,y+h//2+20), (x+w,y+h),(255,255,255),1)
            #make rectangle on lwer face part
            lower_face_gray=gray[y+h//2+15:y+h,x:x+w]
```

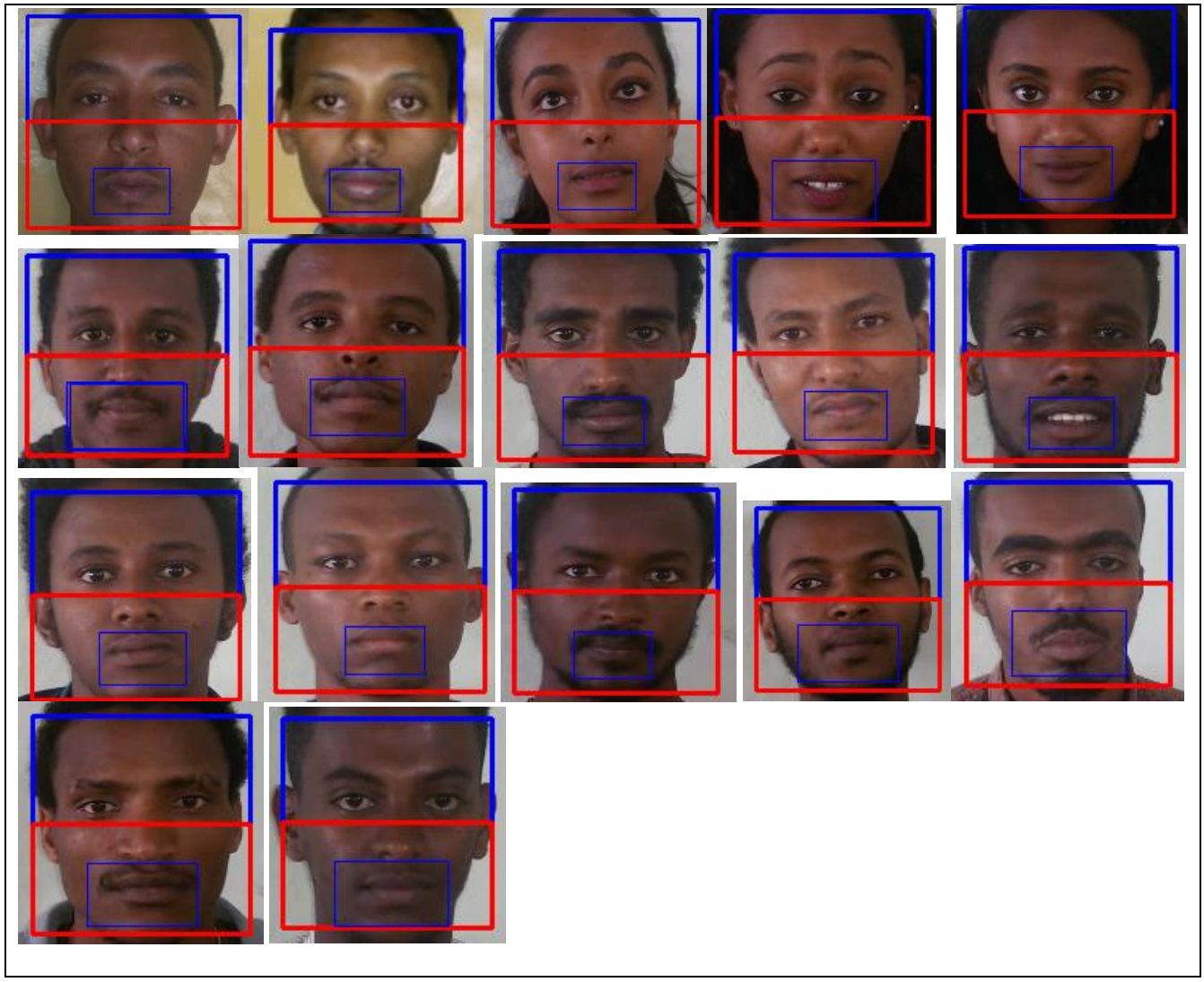
```
            # convert the lower face from RGB to gray
            lower_face_color= img[y+h//2+15:y+h,x:x+w]

            mouth= detector.detectMultiScale(lower_face_gray,1.3,8)
            # apply the cascade classifier to detect
                     the mouth on the lower mouth part
            #if(len(mouth)!=0):
            for (mx,my,mw,mh) in mouth:
                # get the cencer co-ordinate of the mouth
                Mcx= mx + mw//2
                Mcy= my + mh//2

                # co- ordinate of the ROI
                if((ROIh//2)>Mcy or (ROIw//2)>Mcx):
                    print("wrong mouth Detection")
                    ROI= lower_face_gray[my:my+ROIh,mx:mx+ROIw]
                    ROIc= lower_face_color[my:my+ROIh,mx:mx+ROIw]
                else:
                    ROIyi=Mcy-(ROIh//2);
                    ROIyf=Mcy+(ROIh//2);
                    ROIxi=Mcx-(ROIw//2);
                    ROIxf=(Mcx+ROIw//2);
                    ROI= lower_face_gray[ROIyi:ROIyf,ROIxi:ROIxf]
                    ROIc= lower_face_color[ROIyi:ROIyf,ROIxi:ROIxf]
                    cv2.rectangle(lower_face_color,(ROIxi,ROIyi),
                    (ROIxi+ROIw,ROIyi+ROIh),(255,0,0), 2)
                    #result = cv2.face.MinDistancePredictCollector()
                    id,conf=rec.predict(
                    lower_face_gray[ROIyi:ROIyf,ROIxi:ROIxf])
                    #id = result.getLabel()
                    #conf = result.getDist()
                    pridict_value=get_predict_value(id)
                    if(pridict_value!=None):
                        value=pridict_value[3]
                        lst.append(value)
                            #cv2.putText(img, value,(x-40,y+h-100),
                            fontFace, fontScale, fontColor)
        #cv2.cv.PutText(cv2.cv.fromarray(img),str(id),(x,y+h),font,255)
        cv2.imshow('frame',img)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

cap.release()
cv2.destroyAllWindows()
```

Appendix E: Face and Mouth Detection Sample Result from AAVC Speakers



Appendix F: Visual Feature Extraction Using Discrete Wavelet Transform (DWT)

```
# -*- coding: utf-8 -*-
"""
@Author: Befkadu
This Code About

The DWT transform decomposes the input image into a
low-frequency sub band (known as the approximate image)
and high-frequency sub-bands (known as detailed images),

"""

import numpy as np
import matplotlib.pyplot as plt
import cv2
import pywt
import pywt.data


# Load image
original = cv2.imread('sampleROIinages.jpg')
original = cv2.cvtColor( original,cv2.COLOR_RGB2GRAY )

# Wavelet transform of image, and plot approximation and details
titles = ['Approximation', ' Horizontal detail', 'Vertical detail',
'Diagonal detail']
coeffs2 = pywt.dwt2(original, 'bior1.3')
LL, (LH, HL, HH) = coeffs2

""" low frequency components are known as approximate coefficients"""
cv2.imwrite('transformed2.jpg', LL)
img_transformed = cv2.imread('transformed1.jpg',0)
img_transformed = cv2.resize(img_transformed,(160,90))
cv2.imshow("Approximation",img_transformed)

""" high frequency components are known as detailed coefficients """
for i, a in enumerate([LH, HL, HH]):
    a = cv2.resize(a,(160,90))
    cv2.imwrite(titles[i+1] +'.jpg',a)
    #cv2.cv.SaveImage('transformed1.jpg', LL)
    cv2.imshow(titles[i+1],a)



cv2.waitKey(0)
cv2.destroyAllWindows()
```

Appendix G: Database Tables of The System

Table name: visual_speech | WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
1	id	INTEGER							NULL
2	speakerId	INTEGER							NULL
3	lableId	INTEGER							NULL
4	lableName	STRING (100)							NULL

Table name: audio_speech | WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
1	id	INTEGER							NULL
2	speakerId	INTEGER							NULL
3	lableId	INTEGER							NULL
4	lableName	STRING (100)							NULL

Table name: audio_visual_speech | WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
1	id	INTEGER							NULL
2	speakerId	INTEGER							NULL
3	lableId	INTEGER							NULL
4	lableName	STRING (100)							NULL

Appendix H: Visual Speech Test Code

```
# -*- coding: utf-8 -*-
"""
Created on Sat Jun 03 11:14:21 2017

@author: befkadu
"""

import cv2
import os
import numpy as np
from PIL import Image
import sqlite3
```

```
path ='../AAVC/VC/Viwel/testing_data/speaker2/' # the pash changed manualy
basd on your test
recognizer = cv2.createRecognizer()
detector1=
cv2.CascadeClassifier('haarcascades/haarcascade_frontalface_default.xml')
detector= cv2.CascadeClassifier('haarcascades/haarcascade_mcs_mouth.xml')

Sample_number=0;
ROIh= 40;
ROIw= 70;

total_number_of_recognized=0; # hold the number of recognized word or hone


rec =cv2.createLBPHFaceRecognizer()
rec.load('TrainedValue/trainngDataforvawels.yml')
id=0
fontFace = cv2.FONT_HERSHEY_SIMPLEX
fontScale = 1
fontColor = (255, 255, 255)

def get_testresualt_value(lableId):
    conn=sqlite3.connect("../VD/Visual_speech_db.db")
    cmd= "SELECT * FROM visual_speech WHERE lableId="+str(lableId)
    cont=conn.execute(cmd)
    resualt=None;
    for row in cont:
        resualt=row
    conn.close()
    return resualt;


testing_vedios=[os.path.join(path,f) for f in os.listdir(path)]


for sample_vedio in testing_vedios:
    lst=['si'];
    value=""
    sample_lable_id=int(os.path.split(sample_vedio)[1].split('.')[0])
    sample_lable_number=int(os.path.split(sample_vedio)[1].split('.')[1])

    print(sample_vedio)
    print("Sample for Lable",sample_lable_id,"Sample
Number=",sample_lable_number)
    Sample_number=Sample_number+1;
    cap= cv2.VideoCapture(sample_vedio)
    print 'File Number=',Sample_number;


    ret , img =cap.read()
    if(not ret):
        print "Opss ! you enter wrong path make shor the path"
    else:
        newx,newy = 300,350;
        img = cv2.resize(img,(newx,newy))
        gray = cv2.cvtColor(img,cv2.COLOR_RGB2GRAY)
        gray=cv2.equalizeHist(gray)
```

```
        #cv2.imshow('gray',gray)
        faces = detector1.detectMultiScale(gray, 1.3, 5)

        for (x,y,w,h) in faces:
            cv2.rectangle(img, (x,y), (x+w,y+h),(0,0,0),2)# make rectange
for the hole face

            #cv2.rectangle(img, (x,y), (x+w,y+h/2),(255,0,0),2)# make
rectangle for upper  face
            cv2.rectangle(img, (x,y+h//2+20),
(x+w,y+h),(255,255,255),1)#make rectangle on lwer face part
            lower_face_gray=gray[y+h//2+15:y+h,x:x+w] # convert the lower
face from RGB to gray
            lower_face_color= img[y+h//2+15:y+h,x:x+w]
            mouth= detector.detectMultiScale(lower_face_gray,1.5,10)

            for (mx,my,mw,mh) in mouth:
                Mcx= mx + mw//2
                Mcy= my + mh//2

                ROIyi=Mcy-(ROIh//2);
                ROIyf=Mcy+(ROIh//2);
                ROIxi=Mcx-(ROIw//2);
                ROIxf=(Mcx+ROIw//2);
                ROI= lower_face_gray[ROIyi:ROIyf,ROIxi:ROIxf]
                ROIc= lower_face_color[ROIyi:ROIyf,ROIxi:ROIxf]
                cv2.rectangle(lower_face_color,(ROIxi,ROIyi),
                (ROIxi+ROIw,ROIyi+ROIh),(255,0,0), 2)
                #result = cv2.face.MinDistancePredictCollector()

id,conf=rec.predict(lower_face_gray[ROIyi:ROIyf,ROIxi:ROIxf])
                #id = result.getLabel()
                #conf = result.getDist()
                pridict_value=get_testresualt_value(id)
                if(pridict_value!=None):
                    value=pridict_value[3]
                    lable_id=pridict_value[2]
                    lst.append(value)
                    if(lable_id==sample_lable_id):
                             total_number_of_recognized=
                             total_number_of_recognized+1;
                    else:

total_number_of_recognized=total_number_of_recognized

                    print 'Predicted as=',max(lst,key=lst.count)
                    print ("Ok")

        cap.release()
        cv2.destroyAllWindows()

print"Total Number of Test Visual speech =",Sample_number
print "Number of recognized word=",total_number_of_recognized
print "Persentage=" ,total_number_of_recognized*100//Sample_number,'%'
```

Appendix I: Audio Feature Extraction

```
# Generate mfccs from a time series

y, sr = librosa.load(librosa.util.example_audio_file())
librosa.feature.mfcc(y=y, sr=sr)
# array([[ -5.229e+02,  -4.944e+02, ...,  -5.229e+02,  -5.229e+02],
# [  7.105e-15,   3.787e+01, ...,  -7.105e-15,  -7.105e-15],
# ...,
# [  1.066e-14,  -7.500e+00, ...,   1.421e-14,   1.421e-14],
# [  3.109e-14,  -5.058e+00, ...,   2.931e-14,   2.931e-14]])

# Use a pre-computed log-power Mel spectrogram

S = librosa.feature.melspectrogram(y=y, sr=sr, n_mels=128,
                                   fmax=8000)
librosa.feature.mfcc(S=librosa.power_to_db(S))
# array([[ -5.207e+02,  -4.898e+02, ...,  -5.207e+02,  -5.207e+02],
# [ -2.576e-14,   4.054e+01, ...,  -3.997e-14,  -3.997e-14],
# ...,
# [  7.105e-15,  -3.534e+00, ...,   0.000e+00,   0.000e+00],
# [  3.020e-14,  -2.613e+00, ...,   3.553e-14,   3.553e-14]])

# Get more components

mfccs = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=40)

# Visualize the MFCC series

import matplotlib.pyplot as plt
plt.figure(figsize=(10, 4))
librosa.display.specshow(mfccs, x_axis='time')
plt.colorbar()
plt.title('MFCC')
plt.tight_layout()
```

Signed Declaration Sheet

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all source of materials used for the thesis have been properly acknowledged.

Declared by:

Name: Befkadu Belete Frew

Signature: _____

Date: _____

Confirmed by advisor:

Name: Yaregal Assabie (PhD)

Signature: _____

Date: _____