Delft University of Technology

Master Thesis

Limits on Modeling Compensation in Multimodal DNNs for Audio Visual Speech Recognition

Author:
Sreejith Chandrasekharan Nair

Supervisors:
ir. Alessio Bazzica
prof. dr. Martha Larson

Exam committee:
dr. Cynthia Liem
dr. Jan Van Gemert
prof. dr. Martha Larson

A thesis submitted in fulfilment of the requirements for the degree of Master of Science

in the

Mutlimedia Computing Group
Intelligent Systems Department

July 2017

Declaration of Authorship

I, Sreejith Chandrasekharan Nair, declare that this thesis titled, 'Limits on Modeling Compensation in Multimodal DNNs for Audio Visual Speech Recognition' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a Master degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"The highest goal in life is to inquire and create, to search the riches of the past, to internalize the parts of them that are significant to you, carry that quest for understanding further in your own way. The purpose of education from that point of view is just to help people determine how to learn on their own. It is you, the learner who is going to achieve in the course of education and it is really up to you what you will master, where you go, how you use it, how you will go on produce something new and exciting for yourself, maybe for others."

Noam Chomsky

DELFT UNIVERSITY OF TECHNOLOGY

Abstract

Electrical Engineering, Mathematics and Computer Science (EEMCS) Faculty
Intelligent Systems Department

Master of Science

Limits on Modeling Compensation in Multimodal DNNs for Audio Visual Speech Recognition

by Sreejith Chandrasekharan Nair

Speech is a natural way of communicating that does not require us to develop any new skills in order to be able to interact with electronic devices. With the evolution of technology, speech has become one of the primary means of communication. Speech recognition is a form of multimedia content analysis, where the information carried in a speech signal is transcribed into a character string. Any information in the real world is perceived via several input channels. Each modality conveys some additional information about a real world concept. Likewise, the perception of speech in a human brain is bimodal in nature. We combine information from both visual and audio modalities to disambiguate speech. The system studied in here is a multimodal speech recognition system, where the features are generated by correlating visual and speech modalities using a multimodal Deep Belief Network. This thesis reproduces this system, and explores several aspects of its performance related to real-life conditions under which speech must be recognized. Since the limitations of multimodal deep learning approaches are not well comprehended, we would like to gain insights into the resemblance of such systems to humans in their ability to level multimodality. The experiments carried out by our study demonstrate that the visual modality complements speech modality, providing information such as place of articulation. Further studies are performed on the system to shed light on the limits of such a multimodal Deep Neural Network for Audio-Visual speech recognition. In real-life, Audio-Visual speech recognition systems will come across several perturbations such as reverberation and visual occlusion. The behavior of this system is analysed in a simulated environment replicating such real-life surroundings. Further, a study is performed to see the effect of the visual modality on recognition of phonemes, which are basic building blocks of speech. The study conducted in this thesis supports the conclusion that the multimodal Deep Neural Network is far from achieving human-like performance in the presence of perturbations. This demonstrates the necessity to conduct more research on the robustness of the multimodal Deep Neural Networks in real-life scenarios.

Contents

List of Figures

List of Tables

Abbreviations

ASR	**A**utomatic **S**peech **R**ecognition
AVSR	**A**udio **V**isual **S**peech **R**ecognition
PCA	**P**rincipal **C**omponent **A**nalysis
SVM	**S**upport **V**ector **M**achine
RBM	**R**estricted **B**oltzmann **M**achine
DNN	**D**eep **N**eural **N**etwork
DBN	**D**eep **B**elief **N**etwork
CUAVE	**C**lemson **U**niversity **A**udio **V**isual **E**xperiments
ROI	**R**egion **O**f **I**nterest
MNIST	**M**ixed **N**ational **I**nstitute of **S**tandards and **T**echnology

This thesis is dedicated to mom, dad and my dearest friends...
For their endless love, support and encouragement

Chapter 1

Introduction

This section gives an introduction to certain important concepts for this thesis project. It presents the key of multimedia content analysis and one of its most important research areas — speech recognition. The evolution of speech recognition systems to their current form is discussed. Finally, research questions are proposed based on the current literature that will be investigatedin this thesis. These questions allow the thesis to achieve its goal of shedding lights on the limits of multimodal Deep Neural Networks for Audio-Visual speech recognition.

1.1 Background

Content analysis can be defined as "any technique for making inferences by objectively and systematically identifying specified characteristics of messages" [9]. This definition of content analysis is applicable for several different areas such as video analysis, textual or document analysis and analysis of drawings. Technologies in the area of content analysis allow a system to distinguish and separate out large volumes of data, in a comparatively easier and systematic way. This help in discovering and depicting the interests of an individual, groups and institutions. Inferences made from content analysis is useful for many real life applications.

One such important content analysis is the video or multimedia content analysis, which is a multidisciplinary technology [10]. Calling this a multidisciplinary approach is justified, as a digital video is always a multimodal data stream. It usually contains visual, audio and textual modalities and each of these modalities contains a part of the whole information carried. Video content analysis comprises techniques for information extraction from images, sounds and text, for the interpretation of the content. Thus, the content

analysis in this sector includes image processing, audio and speech processing, textual information retrieval etc. Video content analysis benefits from research in fields of image processing, speech and audio processing, psychology, pattern recognition, digital coding, statistics and natural language processing.

Multimedia content analysis is the task of understanding the semantics of a multimedia document using computers — for example, a video with associated audio track [11]. Techniques of multimedia content analysis is quite essential in this multimedia digital information era, to retrieve, access and digest any such digital information. A multimedia document's semantics are enclosed in multiple forms which are complementary. As an example, a live TV coverage of a natural calamity such as earthquake communicates more information than what we just hear through reporting. However, watching video of the report alone will not convey the information about the impact of the earthquake in statistical terms. Therefore it is important to analyze all different types of data available with the stream such as video frames, soundtracks, textual information and spoken words, recognized from the soundtracks. Such an analysis involves segmentation of the document into meaningful chunks based on semantics and classification of these chunks into predefined classes. Then these chunks can be indexed and summarized for future retrieval.

For a multimedia stream, segmentation means splitting the whole video into scenes, that represents one particular story or context [11]. For the purpose of achieving coherence, the scenes can further be split into shots. The stages that follow segmentation are task dependent. For example, for a speech recognition task, the modules following segmentation generates features suitable for recognition of speech. The classification of the scenes in multimodal content analysis can be at different levels. A high level classification of the scene can be an 'opera performance', the mid level can be a 'music performance' and low level can be the 'audio corresponding to a music'.

1.2 Problem

Speech recognition, which is one of the ways to analyze multimedia content, can be defined as decoding of the information carried by a speech signal and its transcription into a character set [12]. These characters are then used in different applications such as generating a written form of the speech signal, controlling an application or a machine and database access. One real life example of such an application is in 'Siri' that runs on Apple iOS and mac OS, is an intelligent personal assistant and a knowledge explorer.

Automatic Speech Recognition has to do with the symbol transcriptions, unlike speech understanding, which aims at meaning extraction from the speech signal. Recognition of speech is a knowledge-intensive process where all the available information regarding the speech such as model, rules and references should be taken into consideration. The quantity of knowledge required in each case is task dependent. Less linguistic knowledge is required in recognition of digits than complete sentences containing many words in a language comprising of thousands of words. The Automatic Speech Recognition (ASR) task itself is multidisciplinary task, comprising disciplines such as acoustics, pattern recognition, signal processing, linguistics, neuroscience and psychology. Thus, the research and development groups that develop ASRs are supposed to have knowledge in these several different disciplines.

With the traditional computers, the speech technology was never a primary means of interaction [1]. It was mainly because the other means of communication with machine such as keyboard and mouse outperformed speech in terms of reliability, efficiency and accuracy in communication. Also, the computing power available a few decades ago, without modern day technologies such as multi-core processors, Graphical Processing Units (GPUs) and Central Processing Unit (CPU)-GPU clusters was inferior.

Today, the technology has improved so much that the computing power available is several orders of magnitude more than what was available previously [1]. The advancements in speech technology started changing the way we interacted with the electronic devices and thus, the way we live and work. There are several reasons for the popularity of speech recognition in this era, first of which is the increase in computation power. This helps in creating 'computation-intrusive-models' which reduce the error rates of ASRs. This progress resulted in development of robust ASR systems suitable for real life applications. Moreover, the access to data has improved in terms of volume, because of the advancement in technology, specifically in cloud computing. This helps in avoiding assumptions regarding the model by minimization of parameters, making the system more robust. Also, mobile devices such as phones, wearable devices and in-vehicle infotainment devices became very popular for which the traditional input devices were not convenient. Thus, speech, which is a natural way of communication, which does not require developing any special skill, have become a more popular means of interaction.

1.3 Traditional methods

In every speech recognition system, given an acoustic observation sequence corresponding the word sequence is obtained [13]. For this purpose, a probabilistic model of speech

production is assumed, through which a word sequence, W, generates an acoustic observation, Y, with a probability indicated by P(W,Y). The goal of the system is then to decipher a word sequence from the acoustic observation sequence. Here, the selected word sequence will have the maximum a posteriori (MAP) probability among all the sequences, as shown below:

P(A—W) is the acoustic model that gives the probability of an acoustic observation sequence, which is conditioned on a string of words. For large vocabulary speech recognition, statistical models of speech units are built. Word models are in turn built from these statistical models of speech units, where a lexicon describes the composition. Then, word sequences are created and, by concatenation methods acoustic model probabilities are computed. The term P(W) in the equation is known as the language model. This model comprises of syntactic and semantic rules of the language and the speech recognition task.

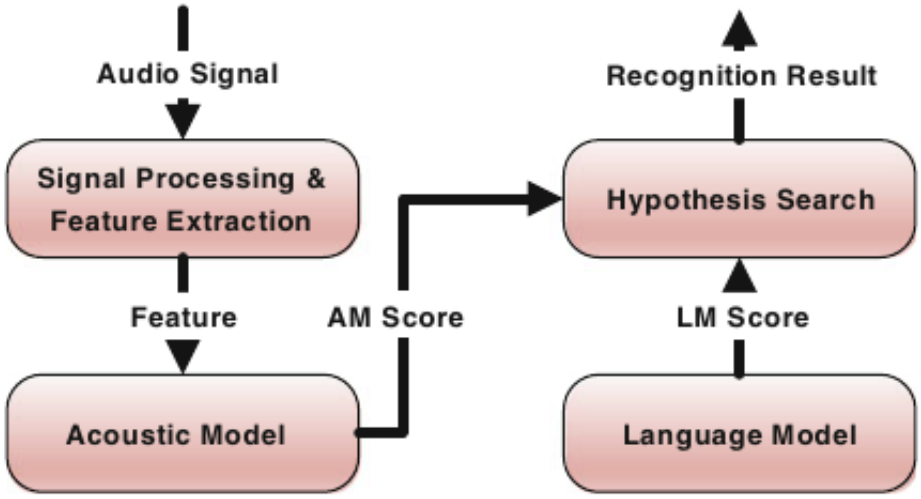

FIGURE 1.1: Architecture of a typical ASR system [1]

The architecture that is typical for ASRs is shown in the figure. It has four key components as shown in the figure: Signal Processing and feature extraction unit, Acoustic Model (AM), Language Model (LM) and hypothesis search. The input in the form of speech signal is given to the Signal Processing and Feature Extraction unit. The signal processing unit will remove the noise and distortion in a channel to some extent. Also, it usually converts the signal into frequency domain from time domain. The feature extraction unit extracts features that are suitable for the acoustic model that follows this unit. The AM unit generates AM scores for the variable length feature sequence. It uses the knowledge of acoustics and phonetics to do this task. LM unit that follows the AM unit gives LM score which is the probability of the speculated word string. To do this task LM learns correlations between words using a training set, which typically

contains text. The accuracy of an LM unit in generating LM score can be improved by prior knowledge of the task. The hypothesis search unit combines the above mentioned two scores for each of the feature vector and corresponding predicted word string to recognize the speech. The word string with the highest combined score is the recognized as the speech at the output of the system. One of the key reasons for the complexity in speech recognition is the complex interplay of several components. These include speaker characteristics (gender, illness that affects tongue, lips or any other issues with vocal organs or the psychological conditions), rate and style of the speech, noise in the background (such as environmental noise and side talk), channel distortion due to differences in microphone, dialects and non-native accents. For real world applications, it is important that the system is robust enough to deal with the above-mentioned variability.

Traditional methods use a generative statistical model, which computes the joint probability of the input feature vector and the labels [14]. Later on, the joint probability is decomposed into prior probability of the labels such as speech class tags and the probability of the class-conditioned data such as acoustic features. Using Bayes rule, the posterior probability of the labels for new, unseen input speech data is obtained. For such generative methods, it is important to have a good model to the true data distribution. The Hidden Markov Model (HMM) is one of the important generative methods and has been popular since 1980s. It has been proved to be a good model for the statistical distribution of time series.

The Markov chain is a stochastic model that captures a chain of possible events. Here, the probability of occurrence of any event depends only on the state when previous event occurred [15]. The key concept of HMM is also state, which is a random variable that takes discrete values for the states. Thus, HMM is an extension of Markov chain where uncertainty is added to each state. One of the basic problem, in speech recognition before HMM was the variable length of acoustic feature sequences. Speech has an elasticity in temporal dimension, and the features generated for recognition should have a correlation with this elasticity. As a consequence, even if two word sequences are identical, the resultant acoustic features sequence can have different lengths at different times, even if the speaker is the same. This peculiarity in the temporal dimension and its correlation to the frequency domain properties posed challenges for the speech recognition domain. The challenges have been addressed by HMMs to a considerable extent. Use of HMM for the representation of quasi stationary speech signals and the use of Expectation Maximization algorithm for the training of the HMM parameters has led to reliable recognition of speech. This success is considered to be one of the key turning points in speech recognition research domain.

Till the early period of 21st century, the shallow, generative approach of HMMs with Gaussian Mixture Models (GMMs) was the most popular way of recognizing speech [16]. The GMM Models were used as the state output distribution of the HMMs. As discussed in [16], the research in that era was mostly focused on constructing deep and dynamic structures using human speech production knowledge. Early work in HMMs generalize and extended the HMM-GMM model with dynamic constraints on its parameters. Later on, hidden layers were added onto this dynamic model. These additions captured to comprise target directed, articulatory like attributes in human speech generation. The latest in HMMs was using these deep structures with hidden dynamics by the addition of non-recursive Finite Impulse Response (FIR) filters. The use of these filters improved the efficiency of the traditional system.

1.4 Critical Challenges of the Traditional Methods

Though traditional methods such as HMM and its variations addressed a few basic requirements of speech recognition, there were many drawbacks [14].

- Knowledge of the articulatory human speech dynamics and the deep control mechanisms of those are still imprecise and incomplete. For the purpose of having efficient computation in training and decoding of speech with the available resource, the available knowledge itself was simplified. This resulted in the diminution of modeling power and precision.
- All of the important traditional methods were using generative learning. These have goal of providing a minimal parameter set for speech variations as a result of coarticulation and environmental factors. This affected the performance with respect to variational inference and resulted in a low quality model for practical applications.
- The traditional models have used isolated characteristics of speech dynamics based on detailed models of human speech production mechanisms. However, the models made the dynamic systems simple for standardization. This simplification was performed without a clear knowledge of the structure and it has many unknowns in it. This is a disadvantage which affects the system especially when inference is made.
- Features used by in these models needed to be statistically independent. [1] This goal arises from the requirement of minimizing the number of parameters. Thus, the flexibility in terms of crafting features was compromised as the features were supposed to be manually crafted.

- Human speech perception uses visual modality to disambiguate information received through the auditory system [17]. The use of visual modality and other modalities were not considered in traditional speech systems. This affects the robustness of such speech recognition systems.

1.5 Overcoming Limitations

The above mentioned limitations are overcome by the recent developments in speech recognition. The major advancements were due to the usage of deep neural networks and multimodality, as explained below.

1.5.1 Evolution of Neural Networks

Recent development in speech recognition has been mostly in the area of deep neural networks which combine generative and discriminative learning [16]. Neural Networks made use of large number of parameters unlike HMMs where the parameters were limited. A typical limitation of GMMs on input features is that they should be statistically independent. This makes the covariance matrix diagonal, reducing the number of parameters in the model. Neural networks do not have any such constraints on the input features. Thus, there is more flexibility in using arbitrary features.

Deep neural networks have several layers of nonlinear processing, which transform raw input features to a representation that is invariant and discriminative [16]. Such a representation can be classified using a log-linear model. Deep neural networks have a hierarchy of features. The features generated by lower layers are able to retrieve local patterns, which are susceptible to changes in raw features. However, as the network becomes deeper, the higher layers produce more invariant and discriminative features. Several layers of simple nonlinear processing generate a complicated nonlinear transform at the end. These features will be more abstract and the variability of raw feature will have less effect on these. This makes it robust to variations in environment and speakers. Also, in HMM-GMMs, the variations in environment and speakers were supposed to be handled separately, which is not required with the deep neural networks. Besides, the higher the layer is, the more the saturation for neurons (high confidence), which means the associated features are sparse. The sparse features also help in adapting varying levels of information.

It can be seen in Figure 1.2 that the higher level features are more abstract and invariant. Imagenet database[1] shown in this figure is an image database. In GMM-HMM systems,

[1]More details : http://image-net.org

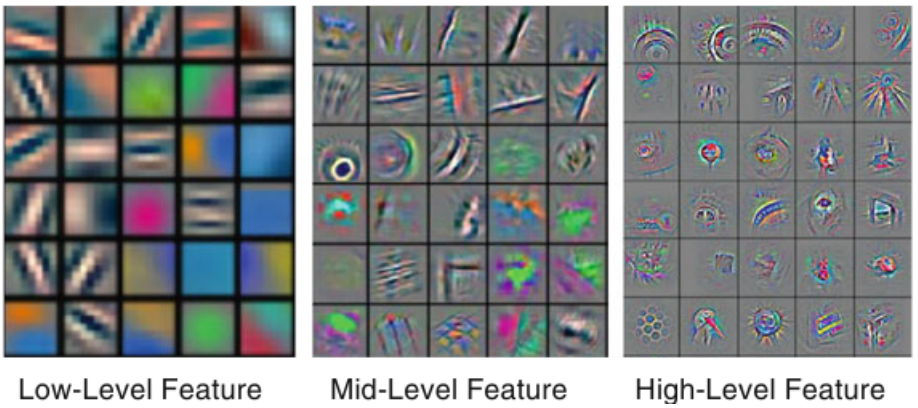

FIGURE 1.2: Feature hierarchy learned by a network of many layers in imageNet

the speaker variability was handed with by two different methods, namely vocal tract length normalization (VTLN) and feature space maximum likelihood linear regression (fMLLR) [16]. These additional techniques are not required in neural network systems since the robustness of these systems towards speaker variation is better compared to the traditional systems. GMM based systems are also known be very sensitive to environmental variability. Typically to deal with these issues Vector Taylor Series adaptation (VTS) or maximum likelihood linear regression (MLLR) are the technologies used with GMM-HMMs. In contrast, deep neural networks have the capability to create internal representations that are robust in the presence of environmental differences in the training data. Also, with different speaking rates and noise levels, deep neural networks have comparable performance that of GMM-HMMs.

One of the important improvements with deep neural networks is their capability to extract feature from the raw signals [16]. This is considered to be an important improvement for real life systems as GMM-HMMs require hand-crafted features. In certain deep neural network architectures, it was seen that the raw spectrogram of linear filter bank features give an edge over the Mel-frequency cepstral coefficient (MFCC) features which was typically used in traditional systems. This advantage is mainly due to raw features' ability to retain more information. Also, it permits the use of convolution and pooling processes to represent and deal with speech invariance and variability in the frequency domain. Finally, deep learning learns raw features by going all the way down to the speech sound waveforms. The learning is performed through an unsupervised task and still outperforms MFCC features. Recently, in 2015, it was found that the raw features become comparable to filter banks if trained through supervised tasks [16].

1.5.2 Evolution of multimodal systems

Another important evolution was the use of multiple modalities for performance improvement in many multimedia content analysis systems [18]. Modality means mode of communication for human senses and types of inputs for a computer. For the human senses modes are sight, touch, hearing, smell and taste. Computer input devices exist for each modalities similar to human senses, such as camera for sight (visual modality), microphones for speech (speech modality), haptic sensors for touch and so on. There are more input devices that do not directly map human senses such as keyboard, mouse, writing tablet, motion sensors that also produce additional modalities. A multimodal system is a system that uses a combination of these modalities. For example, a system that uses a camera and a motion sensor for tracking, utilizes two modalities and can be called as a multimodal system.

Video content typically consists of synchronized audio and visual elements. Both of these modalities contribute to the high level semantics [19]. The presence of one has significant effect on the perception of the other. For example hearing a dog's sound we expect a dog in the video frame. While a person's lips are moving we expect the corresponding speech and so on. Through psychological experiments, the synergistic integration of various modalities in human perception systems have been proved. McGurk effect which will be explained later on, is a well known example of this synergy. With traditional video content analysis applications, these modalities were treated separately. However, with this separation of modalities, important information about the whole event or object is lost. Taking into consideration the inherent association of audio and visual modalities integration is advantageous for content analysis. The following section will explain the human perception speech. This helps in understanding the effect of visual modality on speech with more clarity.

1.5.2.1 Human Perception of Speech

Perception of speech with human brain is bimodal by nature. Human beings combine information from both visual and audio modalities to disambiguate speech [20]. This was quantified by various studies from old times such as [21]. It was shown [21] that the facial information is helpful in improving the intelligibility of speech, in cases where noise is present. Contribution from the visual modality can be of extensive use when speech to noise ratio is low. A clear demonstration of the collaboration of both modalities can be perceived in McGurk effect that is explained in the next section. The benefits of visual modality on speech can be summarised in the following points:

1. Visual modality is useful in the localization of the speaker
2. The segmental information of speech is contained in the visual signal that adds on to the speech intelligibility
3. Visual signal provides information about the place of articulation that is complementary to speech. This is possible as articulators such as tongue, teeth and lips are seen.
4. Jaw and lower face muscle movements also help in disambiguation of speech for human beings

Phonemes and visemes

In the context of spoken languages, a phone is defined to be "the smallest segmental unit of sound which generates meaningful contrast between utterances" [1]. The sounds that constitute phonemes are called phones or allophones. The /p/ sounds in pit or tip constitute a group of difficultly distinguishable sounds that are assumed to hold same function by the speakers of the language. So, these can be considered to belong to the same phoneme. For small vocabulary speech recognition tasks, word-level systems are commonly used. However, when it comes to the traditional speech recognition systems, phonemes constitute the building blocks of the system and each phoneme belongs to a recognition model in such a system.

Visemes are similar to phonemes;they are the reflex of phoneme's visual modality and, it is present in the context of visual modality [1]. Visemes are used in the context of assessing the human capability to recognize phonemes in the absence of auditory signal. They are a set of identical phonemes in the context of visual modality and not to be confused with the phonemes in speech modality. There is not a one to one mapping between phonemes and visemes. Also, the ease of distinguishing utterances in the visual modality is reduced and the expectation of the performance of a lip reader is less compared to a listener of speech.

McGurk effect

In [17], it is observed that the syllable /ba/ dubbed onto the video of lip motions of /ga/ results in human perception of the syllable as /da/ thus, effect is called the McGurk effect. As previously mentioned it is an important instance demonstrating the multimodal nature of speech perception in human beings. This effect made it clear

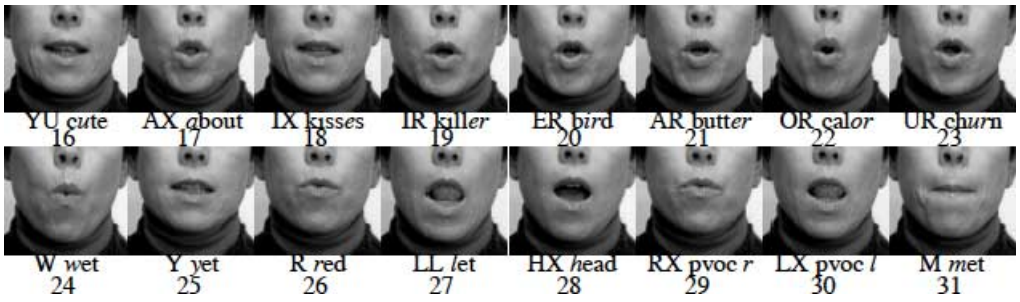

FIGURE 1.3: a partial viseme table [1]

that contemporary mono-modal speech recognition is inadequate in such cases where the visual modality disambiguates similar sounding syllables.

In human speech perception, the information from both modalities is fused into a new component [17]. This component is something that is not present in either modality, still, is necessary for human speech perception. This idea of using visual modality which captures the place of articulation and muscle movements to disambiguate between syllables of similar acoustics, led to the development of multimodal speech recognition systems in recent times.

1.5.2.2 Multimodal Speech Recognition Systems

According to information theory, mutual information of two random variables signifies the quantity of mutual dependence between those two variables. In the context of multimodal speech recognition systems, the variables are the modalities involved in video content analysis. In [22], the mutual information between the audio and time-shifted visual features were plotted as a function of temporal offset. It was found that the mutual information has always a maximum between offsets of 0 to 120 milliseconds. These high values of mutual information indicate a strong correspondence between these two modalities, even when they are non-linearly correlated.

In [23], it was shown that the use of multimodality can support mutual disambiguation of the input signals in speech recognition. It also yields higher overall recognition rates than in the traditional spoken language processing using single modality. Use of multimodality through fusion was also shown to give higher performance in a noisy environment than in a less error-prone or quiet environment. Further performance was improved for accented speech with challenging user groups and usage contexts. These improvements were due to the mutual disambiguation, which is helpful even when one of the modalities failed in identifying the intended meaning. The same result was obtained over different setups such use of several microphones types.

It was demonstrated in [24] that the adaptive fusion of multiple modalities helps in removing the uncertainty that exits when single modality is used. With this proposed system, highly adaptable multimodal fusion rules were used. These rules were based on the weights that naturally emerge from each of the feature stream, which, in turn are based on the noise uncertainty. This model illustrates the effectiveness of use of multimodality in ASRs. Cross modal association, which uses extraction of heterogeneous features from multiple modalities, was shown to be more effective than traditional single modal systems [19]. With cross modality, unlike fusion, an associated visual feature is used when the audio is corrupted. Thus, here the query from one modality can be used with the content of a different modality.

The above-mentioned architectures shows that, with both cross modal association and multimodal fusion, the use of multiple modalities improve the robustness and performance of the system. With multimodal systems, DNNs also have an edge over the traditional methods. This advantage has two aspects [25]. Firstly, DNNs have the ability to model multiple dependencies within its nodes. Secondly, DNNs are capable of integrating the temporal dynamics of the multimodal data. Most of the multimedia tasks including the speech recognition must process time-series data. It follows that the use of multimodality along with DNNs is naturally one of the best choices for speech recognition. The study of state-of-the-art techniques in the next chapter, helped in understanding the limitations of the techniques, which motivated the research questions formulated in this thesis.

1.6 Research Questions

As the study of state-of-the-art techniques in the next chapter reflects, a sizable number of multi-modal deep learning methods have been proposed in current literature. However, until now the limits on the modalities to compensate for each other has not been studied. The following research questions look into a few important aspects of modality compensation in speech recognition. The intention of this study is to understand the limits of such a system and its ability to compensate in several real life scenarios in a selected multimedia task — speech recognition.

1. As the audio signal gets less reliable, is the video signal able to help compensate?

   Method: Artificially reduce the quality of the audio signal by adding reverberation.

2. Does multi-modal speech recognition compensate for different levels of noise differently? (If so, what is the minimal amount of information needed in the video?)

   Method: Use occlusion to reduce the available information in the video, in steps.

3. What is the effect of the visual modality on classification of individual phones?

   Method: Compare the performance of the following two scenarios:

   (a) When the audio is silenced for each of the phoneme

   (b) When both the modalities are silenced for each of the phoneme

1.7 Outline

This section gives an outline of this thesis report.

1.7.1 Overview

This thesis studies a state of the art method for multimedia spoken digit recognition in depth. Here, the attempt is to answer the research questions which are motivated through a study of the state-of-the-art techniques. Experiments are performed via simulation of various conditions of missing information and noise. These experiments put the robustness of the multi-modal DNN speech recognition system to the test. This study will help in understanding the interplay of multiple modalities in a deep neural network and the limits that are faced by multimodal DNN systems.

1.7.2 Thesis Layout

The Chapter wise structure of the document is as follows:

- Chapter 1 this chapter introduces the basic concepts of multimedia content analysis, speech recognition and the traditional and the contemporary systems in abstract, which also explains the motivation, goals and the outline of the thesis document. Here, a few research questions to be investigated in this thesis are formulated based on the understanding of the current literature.
- Chapter 2 briefly discusses the state of the art techniques in Automatic Speech Recognition, multimodal systems and Deep Neural Networks (DNNs).
- Chapter 3 introduces all the datasets that are used in this thesis and explains in which contexts they are useful
- Chapter 4 gives a detailed view of a state of the art multimodal Deep Neural Network, details the system architecture and explains the preprocessing techniques used on different input modalities.

- Chapter 5 gives a detailed explanation of the experiments performed and the results obtained from those.
- Chapter 6 concludes the thesis document by drawing the conclusion of the work performed in here and outlining the future research work in this area.

Chapter 2

State of the art

This section explains the state of the art technologies in the area of focus of this thesis project. It gives a summary of the current developments in the area of deep neural networks and multi-modal systems.

2.1 Deep neural networks

Note that deep neural networks are structures used for machine learning, which learns several representations of the input data in a hierarchy of features or concepts, where each level corresponds to a level of abstraction [26]. The depth of deep learning depends on the task of interest and a particular depth can be suitable for a particular task.

The key aspects of deep learning are the following:

- Models involving multiple layers of non-linear information processing
- Supervised or unsupervised learning of representations of input data at more abstract layers

In deep learning the output from a lower stage is given to the input of the next non-linear stage and so on [27].

2.1.1 Hierarchical learning

Human information processing mechanisms work in a hierarchical manner. These mechanisms extract complex structures in the inputs via internal representations in the hierarchy [26]. The human auditory system has a proper hierarchical structure, with the

inputs received in the form of waveforms, converted into linguistic level at a higher level representation. The visual system also has a hierarchical structure similar to the auditory system.

The following section explains how this knowledge of hierarchical learning motivated in replacing traditional systems with deep neural networks.

2.1.2 Shallow to deep neural networks

Architectures that contain just one or two nonlinear layers for feature transformation are called shallow structures. Commonly used Support Vector Machines (SVM), Multi Layer Perceptrons (MLPs) with just one hidden layer, Gaussian Mixture Models (GMMs) and Maximum Entropy models are shallow models [26]. Shallow models perform well with well-constrained problems that are uncomplicated. However, when it comes to human information processing applications such as speech and visual processing, their limited representational ability will pose problems. On the other hand, DNNs work well with hierarchical nature of such signals since their hierarchical structure deals with this level of complexity. A DNN that has a MLP with multiple hidden layers is a good example of such a deep structure.

2.1.3 Deep neural networks and Multimodality

Information in real world is obtained through several input channels [28]. Each modality conveys some additional information about a real world concept. For example, when it comes to the human auditory system, the visual and speech modalities are correlated since lip movement has a significant role in producing speech. Here, speech is usually represented as audio signal of low frequency or spectrogram, whereas the visual modality is represented as pixels. This difference in representation makes it complex to find the correlation between different modalities representing the same concept, compared to the inputs of same modality. However, as previously mentioned, it is not wise to exclude any modality as each modality carries some unique information which is not present in the other ones. For example, an image with a caption can contain scenery not reflected in the caption or the caption can convey things which are not evident in the scene. The caption can name the place in which the picture was taken, name of a person in the image etc. Along with other complexities, noise and missing input make the multi-modal processing a difficult task.

With multi-modal learning, one of the key ideas is to find modality friendly latent variables with which a low order representation is learned [28]. This representation can

be joined with other representations of other modalities or used independently to learn a higher order representation. In one of the higher order representations, it is quite possible to correlate the inputs which were not possible to correlate at a lower order representation.

2.1.4 Restricted Boltzmann Machines

In 2006, Hinton proposed Restricted Boltzmann Machine (RBM) which aims to simulate the behaviour of human brain analysis through the use of neural networks [29]. One important aspect of RBMs is their ability to generate representation through nonlinear dimensionality reduction, when given high dimensional data. RBMs are also good at reconstructing input when given features. Structure-wise, an RBM is a Markov random field with one layer visible units which are typically Bernoulli or Gaussian valued and one layer of hidden units which are typically Bernoulli units [27]. Having Bernoulli unit implies the expected values for the units are binary. Whereas, with Gaussian units the values of the unit can be real-valued. There will not be any visible to visible or hidden to hidden connections and, for this reason an RBM can be represented as a bipartite graph.

An RBM can be represented by a joint distribution p ($\mathbf{v}$,$\mathbf{h}$; θ), where $\mathbf{v}$ are visible unites, $\mathbf{h}$ are hidden units and θ represents the model parameters. The probability of a state ($\mathbf{v}$,$\mathbf{h}$) for an RBM is defined in terms of energy function as below:

$$p(\mathbf{v}, \mathbf{h}; \theta) = \frac{exp(-E(\mathbf{v}, \mathbf{h}; \theta))}{Z} \tag{2.1}$$

,where Z is a normalization factor or partition function.

Further details of the steps of learning in RBM is discussed in the appendix, in the section Contrastive Divergence.

2.1.5 Deep Belief networks

By stacking a number of RBMs so that the learning happens layer by layer from bottom to top the network, Deep Belief Network is created [27]. The first layer is a Gaussian-Bernoulli RBM in case of speech related tasks where the features have continuous values. From the next layer, it is Bernoulli-Bernoulli RBM where the activation probabilities of previous layers are used as the visible data input for training. This is repeated over more deeper layers for the network. Under this model, stacking will improve the lower bound

on the likelihood of the training data. Thus, this greedy process can achieve approximate maximum likelihood learning. This kind of learning does not require labelled data as it is unsupervised.

2.2 State of the Art Techniques

This section describes various advancements in the area of speech recognition research. It can be seen that most of the recent advancements are using one or another form of Deep Neural Networks. Another key development in the area of speech recognition is the use of multiple modalities. This study of the recent advancements will allow the formulation of research questions for this thesis project.

Multimodal Deep Learning Architectures

One of the important developments in the deep neural network research was from Ngiam et al in [8]. This paper proposed multimodal Deep Neural Networks which can generate features correlating the audio and video modalities. So, these features are useful for both multimodal fusion and cross modal applications. For generating multimodal features, first audio and video are given to two separate shallow RBMs. Then, using the hidden layers, a mutlimodal layer is trained and multimodal features are generated. Speech recognition performed on the CUAVE database using a combination of multimodal features and audio RBM features gave impressive results. It outperformed all the shallow features when it comes to noisy audio. From these results, it can be understood that the video modality complements the audio modality, providing information such as place of articulation. This information helps in distinguishing similar sounding speech. It was also shown that the bimodal Deep Autoencoder shows McGurk effect just like the human speech perception.

Another recent, interesting and popular proposal was given in [30], which uses the deep learning architectures for speech feature and visual feature extractions. A Deep Denoising Autoencoder was used to extract the audio features and a Convolutional Neural Network was used to extract features from the raw images. Finally, a Multi-Stream HMM (MSHMM) was used for feature-level integration of the multiple modalities. This framework was used for isolated word recognition tasks and was shown to give a promising result.

In a more recent development, in [31], a Multimodal Deep Network using bilinear softmax layer was proposed. It is shown in here that the earlier Deep Neural Networks (DNN)

are still showing remarkable performance in both audio and visual classification tasks even after the introduction of Convolutional Neural Networks, hybrid architectures etc. This study also shows that the use of visual modality is helpful not just in the noisy speech case, moreover, it is useful in the case of clean speech. Here, similar to the case of Ngiam et al's architecture [8], two separate networks of DNNs are used for video and speech. The final representation is given to a bilinear bimodal DNN for the generation of representation, which is to be used in classification tasks.

Automatic Speech Recognition Systems and Reverberation

Deep learning systems helped Automatic Speech Recognition (ASR) systems have phenomenal improvement in performance [32]. Still, ASR systems are very sensitive to speech degradations including reverberation, which results in considerable reduction in accuracy of the speech. The vulnerability of ASR systems are revealed in non-matching surroundings, where the test set contain noise such as reverberation. Reverberation is one of the important sources of degradation and the performance degradation usually happens with increase in reverberation time, expressed as RT60. The time taken by the reflections of a direct speech to decay to a level of 60 dB is expressed as RT60. For higher RT60 values, the higher will be the reverberation and, correspondingly the greater will be the degradation in performance. There is much research happening in the area of reverberation, with importance given to microphone-array processing, echo cancellation, robust signal processing and speech enhancement. Previous ASR systems tried to improve robustness of the system against reverberation by using signal processing techniques and dereverberation strategies. Recently, the deep learning techniques redefined the ASR acoustic modeling. The researchers came up with discriminative learning technique including neural networks to replace generative learning techniques such Gaussian Mixture Models (GMMs), which were traditional. Still, the use of recent technologies such as multimodal neural networks in reverberation scenarios is yet to be explored.

It has been seen that Deep Neural Networks are able to learn speaker-invariant representations of the data. It was found in [33] that Convolutional Deep Neural Networks (CDNNs) are more robust to degradations compared to DNNs. In this paper, a conventional CDNN, which uses two separate convolution layers, namely time-frequency convolution neural network (TFCNN), improves performance compared to DNNs. In this neural network, one layer is a made to operate across frequency and the second layer operates across time.

In [34], an improved Deep Neural Network (DNN) acoustic model was used for speech recognition in a reverberant environment. A method was used in here, which parameterize the reverberant environment obtained from the input signal to train a room-aware DNN. This paper implemented multi-task learning, where the model is trained to deal with multiple tasks at the same time, with the help of shared parameters. The improvement in robustness against reverberation is achieved via a unique framework that adds on auxiliary information to a DNN model. A vector of room specific features is given to the network which distinguish the utterances in a reverberant environment.

In [35], a denoising autoencoder is used on reverberant environment as a front-end which is capable of reconstructing clean speech spectrum given the noisy input. Context specific effects of speech are captured via a window of several short windowed spectral frames that are concatenated as a single input vector. Also, short and long term spectrum combinations are observed to deal with the long impulse response of reverberation still retaining the necessary time resolution for speech recognition. These ways of dealing with spectra results in better performance of the system. The authors of [35] also proposed an extension of the framework that uses multiple microphones and multi-modality.

A spatial diffuseness feature for Deep Neural Networks (DNNs) is proposed in [36] and is shown to be effective in improving accuracy in reverberant environments. The feature is calculated in real-time using signals from multiple microphones. It does not require information regarding the direction of arrival. This feature represents the proportional quantity of diffuse noise in every time-frequency bin. The system can also use information from multiple microphones as an additional feature input for the DNN speech recognition system. The diffuseness features are normalized as it reaches the microphone array geometry. Thus, this feature can be extracted from several different multichannel recording devices avoiding the need for adaptation of the acoustic model. Also, its computation can be performed in real-time and direction of arrival estimates are required. These properties make it suitable for practical applications such as speech recognition on smartphones using cloud systems.

Other important work in this research area was a Deep Denoising Autoencoder (DDA) framework that generates robust speech features for recognition in reverberant environments [37]. First step for this approach is to pre-train Restricted Boltzmann machines (RBMs) in an unsupervised manner. To get an autoencoder, the framework is unrolled. Fine tuning is performed on the DDA using clean features, which helps in learning stochastic mapping from corrupted features to the clean features. Acoustic models are re-trained using these clean features for performing speech recognition.

Visual Modality in Speech Recognition Systems

Visual occlusion is arguably the least explored area in speech recognition system. Most of the recent literature regarding visual occlusion is around face recognition systems. [38] It is is seen that an accurate localisation of lips or face itself is a hard task. This is due to the differences in lighting condition, backgrounds, quality of the sensors, dynamics of the lips, shadowing of facial regions, head pose, expressions, head rotations and occlusion due to objects and hand movements. Also, faces have a lot of variability due the factors such as facial expression, shape differences and aging resulting in variation of accuracy based different conditions.

The approaches in recognition systems to handle occlusion can be seperated into two categories [39]. In one approach, occlusion is treated as a reconstruction problem, whereas in the second approach it is treated by a patch matching method. In the reconstruction approach, the occluded image is assigned to a class of enrolled images with minimal reconstruction error. For this approach, it is necessary that there be enough samples of images from the same person to do the reconstruction. This makes the reconstruction approach less useful in real-life scenarios. In patch-matching approach, the occluded images are partitioned into sub patches from which features are extracted. Patching is performed depending on the similarity between the patch pairs. Face recognition using dynamic image-to-class warping described in [39] is one such recently developed approach using patch matching.

Another recent development is the Robust Boltzmann Machine or RoBM. [40]. This is a Boltzmann Machine that is made to be robust against corruptions. This is a remarkable advancement in visual recognition as it is able to deal with occlusions in the right way. This robustness is attributed to the use of multiplicative gating, which results in a scale mixture of Gaussians over the pixels. This Boltzmann Machine can be trained in an unsupervised manner using unlabeled and noisy data. With this method, it will be able to learn the spatial structure of the occluding objects and make it flexible as parameters are learned from the data.

Additional researches has been carried out on the occlusion problem in recognition, especially in the domain of artificial neural networks [41]. One such popular method is Double Channel Stacked Sparse De-noising Auto-encoder (DC-SSDA), which is designed to deal with the occlusion removal problem. A Stacked Sparse De-noising Auto encoder uses a greedy layer-wise algorithm for learning representations. The deep architecture of SSDAs helps in learning features that are represented as the previous layer's output. Here, one SSDA is trained for noisy data and encoding parameters are transferred to

another one for clean data. A Noisy node is a node for which the average relative activation difference is high. Such a node's activations are replaced with the corresponding clean node's activations to lower the activation or to make such nodes inactive. By this method, it is possible to get a good indicator of the noise and clean image regions which is the difference in activation values of the above mentioned two SSDAs. Thus, unlike the previously mentioned methods, this method can precisely sense occlusions, without having any prior knowledge of the occlusions. To remove occlusions, it is possible to replace the noise nodes with the clean node's output and the effect of occlusion will be removed.

Another new neural network architecture namely DeepID2+, is a high performance deep convolutional network [42]. It has properties such as sparsity, selectiveness and robustness resulting in a better performance. Also, without having the occluded patterns in the training set, it is robust to occlusions.Unlike older systems where robustness against occlusion is achieved through adding components and regularizations to the models, such properties are naturally presennt in the Deep Neural Networks due to large scale training. Thus DeepID2+ showcases the ability of deep learning for scenarios such as occlusion handling.

Phonetic Classification and Speech Recognition Systems

There are several researches recently performed on the use of visual modality for the phonetic classification. One important work in phone recognition is the Context-independent and pre-trained Deep Neural Network HMM (DNN-HMM), which is a hybrid architecture that achieved a promising performance and it performed even better than the conventional Gaussian mixture model GMM-HMM. Also, in [43], the same architecture was applied to large vocabulary speech recognition context and achieved a reliable performance. Another important development was the Tensor Deep Stacking Network (T-DSN), in which several layers are stacked and a bilinear mapping from hidden representations to the output in each layer incorporate higher order statistics of the input [44]. The T-DSN performs equivalently to DNNs in phonetic classification. With continuous phonetic recognition, T-DSN's performance is similar to that of a DNN, still, there is no requirement of a difficult to scale and sequential fine tuning. The performance improvement is largely due to its capability to model complex correlations in speech by the DNNs. However, in 2014, Abdel-Hamid et al in [45], showed that even further error rate reduction is possible by the use of convolutional neural networks (CNNs). CNNs were able to reduce the error rates by around 6-10% compared to the DNNs on both large vocabulary speech recognition and TIMIT phone recognition.

Again in [31], deep multimodal learning is used for phonetic classification from speech and visual modalities. It was shown that even for the clean acoustic condition, the use of the visual modality along with speech provides a considerable improvement in the performance of phonetic classification. There are two models introduced in this literature. One approach is to train two separate networks DNNa for the speech modality and DNNv for the visual modality. Then, a joint Audio-Visual representation is formed by concatenation of the outputs DNNa and DNNv. The paper also introduces another bilinear bimodal DNN, that uses the correlation between the speech and visual modalities resulting in even further error reduction. It recommends deep multimodal learning for the task of phonetic classification from speech and visual modalities. This architecture which was published in 2015 is very similar to the architecture in [8]. In another development, a deep denoising autoencoder used in [30] for acquisition of noise-robust speech features and convolutional neural network (CNN) was used for acquisition of visual features from raw mouth area images. Then, a multi-stream HMM (MSHMM) integrates the speech and visual modality trained with the corresponding features, resulting in a promising performance for phonetic recognition.

Chapter 3

Multimodal Speech Recognition System

This section discusses a Multimodal Speech Recognition System, that takes Audio-Video speech signals as the inputs, preprocesses the inputs, generates representations using neural networks and classify spoken digits in the signal. Ngiam et al's proposal in [8], which is a multimodal speech recognition system, is taken as the basic framework to answer the research questions formulated in this thesis project. In a more recent development in [31], it is shown that the architecture proposed by Ngiam et al is still relevant. With a small modification to Nigam et al's system, and continued use of multimodal features, it yields better results for clean speech. The difference between the architectures [8, 31] is that the system in [31] has a bilinear softmax layer as the multimodal joining layer for video and audio modalities, in the place of multimodal DBN in [8]. However, the architecture in [31] is used for multimodal deep learning from audio and visual modalities in phonetic classification.

In this chapter, the architecture of Ngiam et al in [8] is detailed at the implementation level. No source code or details of the algorithm implementation are available for this architecture. As the reference paper [8] did not contain specific details, the knowledge acquired from other sources is used in the detailed implementation level. Also, open source libraries available for implementing various features are used. These are modified to suit the functionalities required for this thesis project.

3.1 System Level Architecture

There are 3 different systems that will be discussed in the following section of this chapter. The 3 systems are video-only system, audio-only system and multimodal system. The system level architecture is shown in the block diagram, Figure 3.1. This provides an overview of the flow within the framework for the video-only system. The module-wise functionalities will be discussed in detail in the following sections. Speech and multimodal systems have similar architectures that will be discussed in detail later.

The database used for this thesis project consists of videos that contain the audios and visuals of ten digits spoken by several subjects. As the initial step, audio and video are preprocessed. Preprocessing stage includes segmentation, PCA whitening and generation of temporal derivatives [8]. The preprocessed input, in turn, is provided to two different RBM models (video-only RBM and audio-only RBM). Thus, the input to the neural network models are contiguous audio in the form of spectrogram and video frames (for audio-only and video-only systems respectively). The RBMs are trained separately for each modality. By training the RBMs, the posteriors of the audio and video input data are obtained at the hidden layers. A greedy layer-wise training of a sparse RBM by Lee et al [46] is used for this architecture. This RBM is already known to learn meaningful features for the natural images and digits. For audio-only and video-only systems, the hidden representations generated by the RBMs are used by the classifier.

For a multimodal system, the posteriors from the audio RBM and the video RBM are provided as the input to a bimodal Deep Belief Network (DBN). The generated bimodal features learn the relationship between the modalities involved. These multimodal features along with the features generated by the shallow audio RBM is used to represent the training and test data provided to the classifier. The classifier used for this thesis project is a linear Support Vector Machine (SVM) as proposed by [8]. The linear SVM classifies the provided input into the ten possible classes of digits.

The following sections explain the information about the stages of architecture provided in [8]. These sections detail the implementation used to achieve the same and an attempt is made to understand the motivation behind the choices. As there is no open source code available for the architecture for reference or reuse, the implementations are performed based on [8] and the communication with the authors of the paper. Choices of libraries for each stage is made based on the availability of the source code, stability and the ease of use. From now on, the term 'reference architecture' refers to the architecture of the system proposed in [8] and 'reference stack' refers to the implementation (source code) based on the reference architecture.

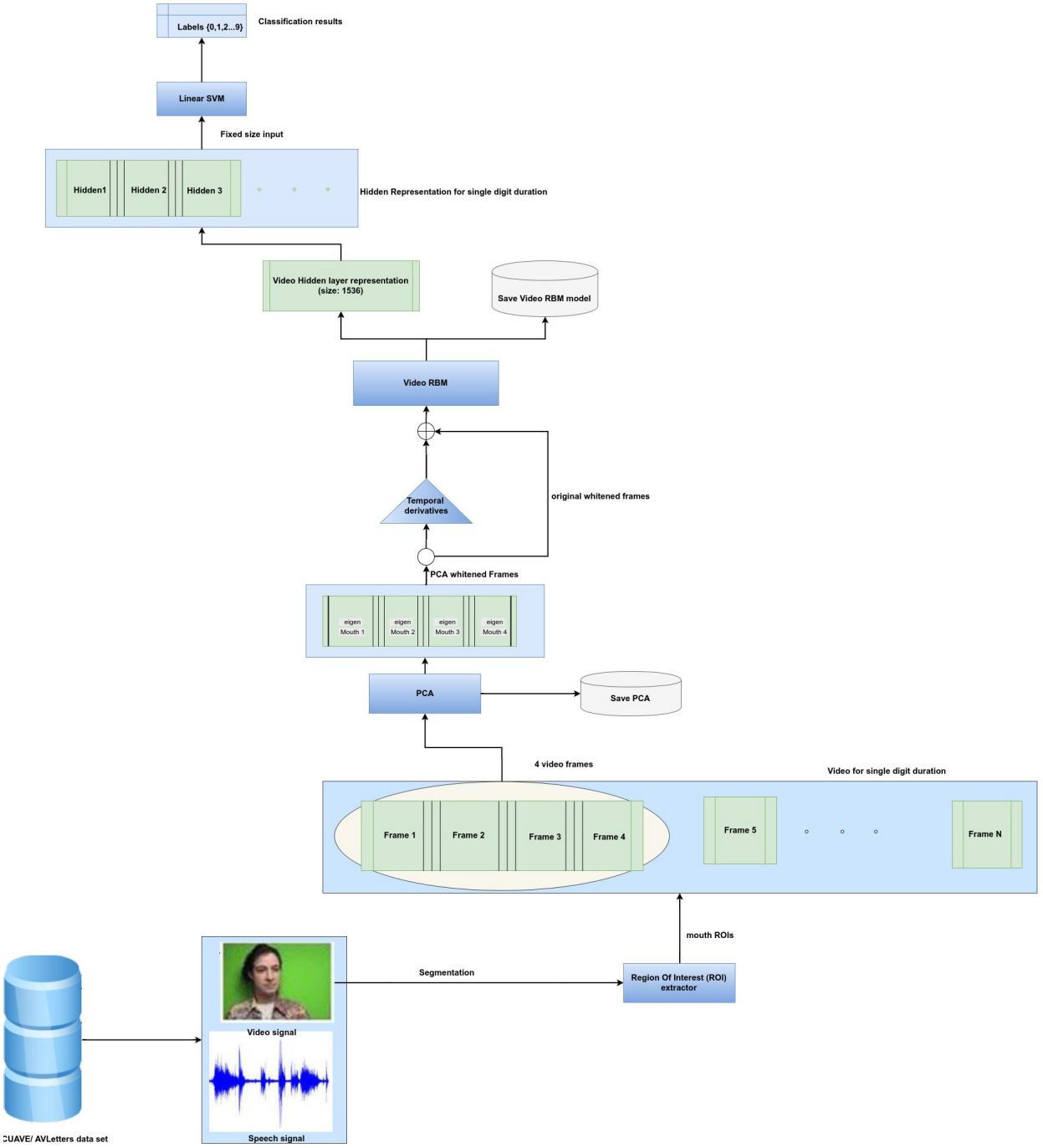

FIGURE 3.1: Top level flow diagram for video-only system

3.1.1 Preprocessing Stage

Preprocessing is the first stage of this multimodal speech recognition system. An audio-visual database is used to provide the inputs to the preprocessing stage. The output of the preprocessing stage is provided as the input to a Neural Network. This section details the databases used for the speech recognition and the stages of preprocessing performed on this data, before providing it as the input to the Neural Networks.

Out of all the preprocessing modules for the input data, a few are exclusive to audio or video. A few of the preprocessing modules are used for both the modalities. The details of these preprocessing modules are explained in the following sections. In both the cases, segmentation is the first step of preprocessing. After segmentation, video-only and audio-only systems have a preprocessing module specific for each of the modalities. Video-only system has a Region Of Interest extractor which generates the required frames for the later stages of preprocessing. Whereas, audio-only system has a spectrogram application that generates the required frames for the later stages. The later stages of preprocessing are similar for video-only and audio-only systems, though, the order of the modules are different. There are no preprocessing modules for the multimodal system as it uses the outputs of audio and video RBMs.

3.1.1.1 Database for Speech Recognition Experiments

As per Ngiam et al, for the reference architecture, the following databases were used: CUAVE, AVLetters, AVLetters2, Stanford database and TIMIT, for the unsupervised training of the auto encoders [8]. For the purpose of training RBM, Multimodal DBN and the supervised classification, only the CUAVE database was used. No fixed protocol has been defined for the division of training set and test set, so, as per the author's suggestion, cross validation was used on the CUAVE database. Since there are 36 subjects, a 6-fold cross validation should provide a reliable experimental setup. The following section details the features of the CUAVE database.

The CUAVE database is suitable for investigating the research questions. It is a speaker-independent database of isolated digits with raw audio and, video frames showing shoulder and head of the subjects. As raw audio is available, it is possible to add reverberation or any other types of noises to the audio, unlike AVLetters, in which MFCC features represent the audio. This is suitable for the simulation of different experimental conditions such as reverberant rooms. The visual channel depicts a standard speaker where the head of the subject face the camera and there is no deliberate movement of the subjects. This made it possible to focus on the speech recognition task, without increasing the complexity of the face and the mouth recognition tasks.

Clemson University Audio Visual Experiments (CUAVE) database

The CUAVE database is a speaker-independent database of isolated and connected digits of high quality video and audio [47]. It is flexible, comprehensive and contains a corpus of around 7000 connected and isolated utterances of digits. The CUAVE database has two sections, a section containing only individual speakers and another section containing speaker pairs. For this thesis project, only the section containing individual speakers is used. There was no strict control on selection of individuals for the part containing individual speakers. In total, 36 speakers were chosen with an even representation of male and female speakers. The database contains 19 male and 17 female subjects. It was made sure that there is a rarity in the database. The database has subjects with different skin tones, accents and visual features such as facial hair and glasses. Speakers were framed to include shoulders and head. Although, different speaking styles including side to side, back and forth and tilting the head movements are present in the database, only the natural stand still in frontal facing is chosen for the experiment. This makes possible validation of the AV speech recognition system with more tightly controlled experimental conditions. Video is recorded in full color and includes no aids for the face and lip segmentations. The Figure 3.2 shows sample visual representations of a few CUAVE subjects.

The recording of the subjects was performed in an isolated sound booth. The video was recorded at a resolution 720x480 at 29.91 fps, which is NTSC standard [47]. 1 megapixel CCD MiniDV camera and the on-camera microphone was used for recording the subjects. As recording was performed through on-camera mic, noises due to the movement of the subjects are clearly heard in the audio. The recording was made under controlled lighting with a green background. This green background mainly helps in color-keying of different backgrounds and also can be used as an aid for segmenting the face region. Though disruptive mistakes were removed from the video, mistakes in speech were kept for making the database realistic. The data is stored in MPEG-2 file format for each of the speakers in the database. These files have data rates of 5000 kbps and multiplex with 16 bits, 44KHz stereo audio.

3.1.1.2 Audio Preprocessing Modules

A block diagram that provides an overview of the flow of the audio-only system is shown in Figure 3.3. Note that the flow is similar to that of the video-only system in Figure 3.1. After segmentation, audio-only system has a spectrogram application in place of mouth ROI extractor for video-only system. This generates the frames required for the

FIGURE 3.2: CUAVE subjects' sample visual representation

next stage of preprocessing. The number of frames provided as input at a time is 10 for audio, whereas, it is 4 for video. Another difference is that the temporal derivatives are calculated before PCA in audio-only system. Whereas, the temporal derivatives are calculated after PCA in video-only system. Further details will be discussed in the following sections.

The audio present in the CUAVE database is raw audio signals. These audio signals as per [8] should be represented as spectrograms with temporal derivatives. Each of the spectrogram frames is supposed to have a 20 millisecond window, with 10 milliseconds overlap. Each of the frames resulted in 161 frequency bins. This, in turn, results in a 483 dimension vector for each of the audio frames. A 100 dimensional vector is created from this vector using PCA whitening. Totally, 10 contiguous frames are used to create the input vector to the neural network models used in this architecture.

This implementation, which was organized in stages, is explained in the following sections. Further details of the implementation are available in the appendix.

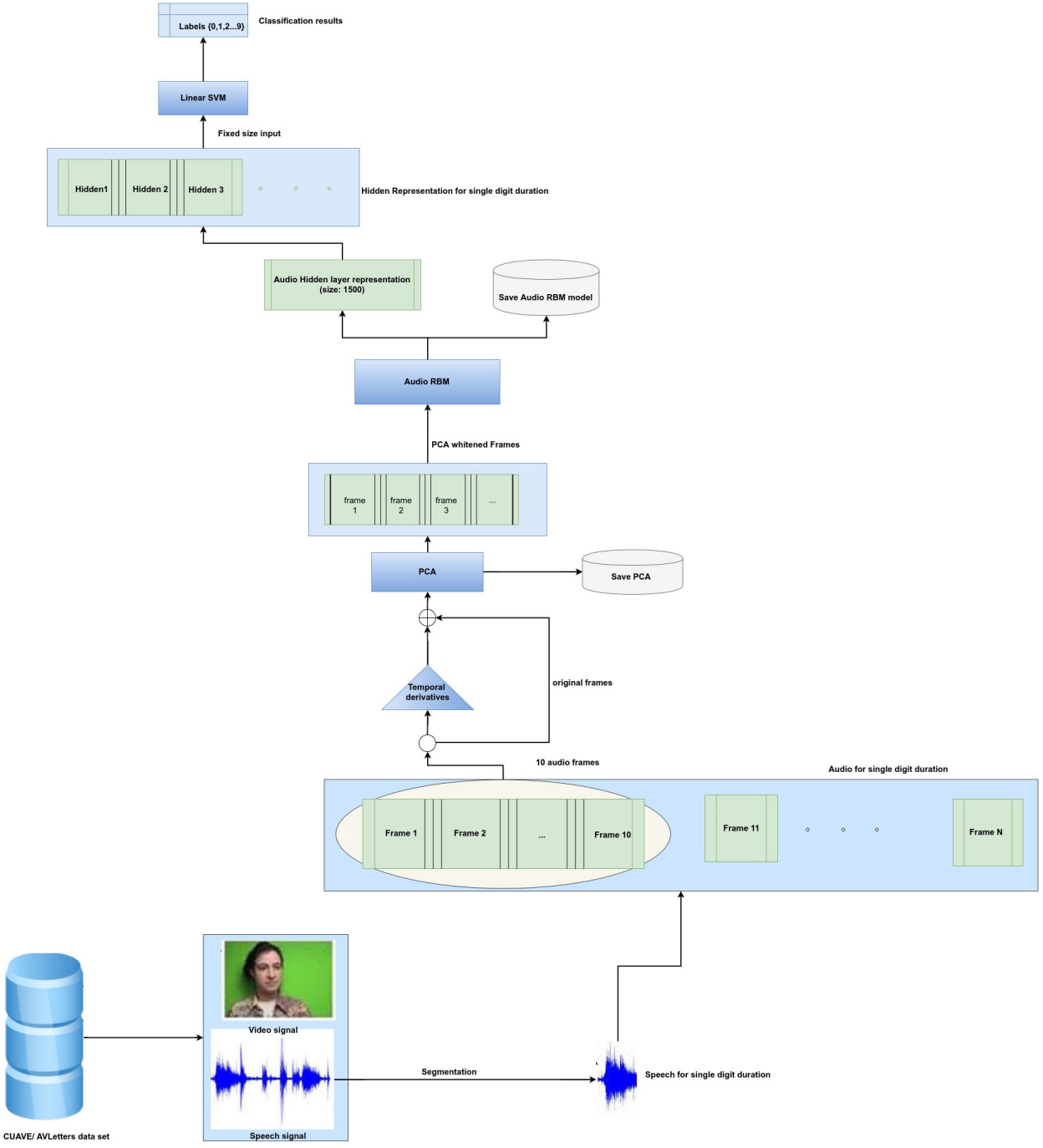

FIGURE 3.3: Top level flow diagram for audio-only system

Digit segmentation

In CUAVE dataset, the digits are spoken separately rather than in a continuous manner. Digit segmentation in this context refers to the separation of digits spoken in the video, removing any silence present between the digits. Since there are 36 different subjects of different ethnicity, age and gender, the variability of the speech is high. This means the digits can be spoken fast or slow, with long or short pauses between digits and loudly or softly. Thus, for answering the research questions, it is important that an automated digit segmentation algorithm is developed. Automated digit segmentation will make sure that even with the variability in speech, we get a reliable segmentation of the data. This helps in obtaining a reliable segmentation even when reverberation is present. Such a reliable segmentation algorithm is important as adding reverberation results in expansion of speech in the temporal domain, resulting in a longer duration for spoken words than in the original data.

The key functionality of this module is to extract the relatively-silent and non-silent sections of the audio. For this purpose, the total duration of the 'frontal-face' is found through manual observation and is set in the code. For the total duration of the frontal face, a search for the non-silent regions of 'sufficient duration' is performed over the audio. The sufficient duration is selected based on manual observation of pauses between the digits. The sufficient duration for relatively-silent region is decided to be anything more than 200 milliseconds. This duration is set based on the analysis performed on selected digits' waveforms, such as six and eight. For these digits, it was seen that there is a silent region within the digit as shown in Figure 3.4 [2]. Please ignore the vertical lines in the diagram as it is not important in the current context.

Also, for the detection of silence, it is very important to know the volume level of silent regions. This can be defined in terms of dBFS, which is the volume expressed in a logarithmic scale [48]. The dBFS level of 0 indicates the peak level of amplitude of sound which can be present in a digital system. Typically the sound is kept at a level less than -10 dBFS in audio signals to ensure the signal quality. The normal sound level can be around -20 dBFS as per [48]. The silence is typically minus infinity dBFS in theoretical case and around -80 dBFS for complete silence in a studio for recording [49]. The typical sound level for a quiet room is known to be around -40dBFS which is set as the default value [50]. So, a few values of silence threshold between -50 dBFS and -30 dBFS are tried and it is found through fine tuning that -45 dBFS provides the most appropriate results for the CUAVE data.

To implement the above-said features, AudioSegment Application Programmable Interface (API) of the pydub Python library is used. With CUAVE subjects, the minimum

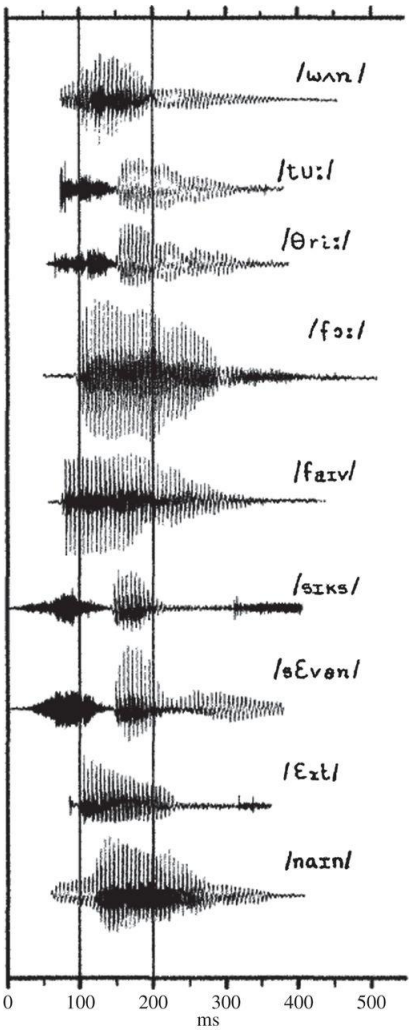

FIGURE 3.4: Shows the typical waveforms of speech digits with the phonemes [2]

silent duration is set to be 200 milliseconds and the silence threshold is set to -45dB which are the values found and fine tuned heuristically. After the detection with AudioSegment, the non-silent regions are extended by a small factor on either side to avoid clipping of edges of speech digits. It was seen that with the above mentioned silence threshold and sufficient duration of silent regions, a part of the digit at the beginning and end are clipped on segmentation. This is because the start and end of the digits have energy levels similar to that of the noise. So, the segments of each digit are extended by a factor based on the actual duration of the digits. The value of the factor is decided based on the observations in waveforms of different subjects. This factor is multiplied with the actual duration and thus extended on either side of the digits.

Manual digit segmentation

It was seen that with the CUAVE subjects, in many cases, the digits were spoken continuous without typical silent region expected in discontinuous speech. When reverberation is added, the overlaps between digits increased and the gaps between the digits are reduced. This made the segmentation through automated means complex for certain subjects or certain regions of the speech signal of many subjects. For these, segmentation is performed manually to achieve a precise digit segmentation. It is performed through Audacity software, by manually adding labels for the segments based on the observed non-silent regions. The digits are segmented with reference to the provided waveforms in Figure 3.4. These waveforms acted as a reference for the typical waveform shapes for the digits 0 to 9. For those subjects for which the energy levels were low, listening to the audio is also helpful in segmenting the digits reliably.

Spectrogram and mat file creation

The Spectrogram is a way to analyse the content of a part of speech signal over a given time [3]. Spectrograms were created by taking the short time Fourier transformation (STFT) of the speech signal. The magnitude of the STFT of speech segments placed in the columns of a matrix and the matrix is shown as an image, called spectrogram. The color scales are used to indicate the magnitude of speech at different frequencies. The temporal resolution of the spectrogram is improved by overlapping windows for the segments of speech [51]. The segments discussed in here are windows within the digit's duration and is different from the segments (which is one digit) in the previous section. Here the spectrogram contains only the magnitude of STFT which means the frequency information. The phase information is not taken into consideration when spectrogram is used.

In Figure 3.5, the spectrogram shows voiced, unvoiced and silent speeches [3]. If vocal cord vibrate during the pronunciation of a phoneme, it is called voiced speech. Unvoiced speech does not make use of the vocal cords. For example, difference between the phonemes /s/ and /z/ or /f/ and /v/ is that /z/ and /v/ use the vocal cords, whereas the other two do not. The harmonics of speech are the horizontal red lines on top of each other in voiced speech. The patterns of concentration of such harmonics are called formants, which is a very unique characteristic of the human speech. Unvoiced signals are almost like noise. Silence is the low energy area of the spectrogram, which appears in the figure as blue.

For the speech, one important assumption that is made is — the speech signal is constant over the window in which spectrogram is calculated. This is due to the quasi stationary nature of the speech, which is supposed to be stationary for around a 20 ms window. In

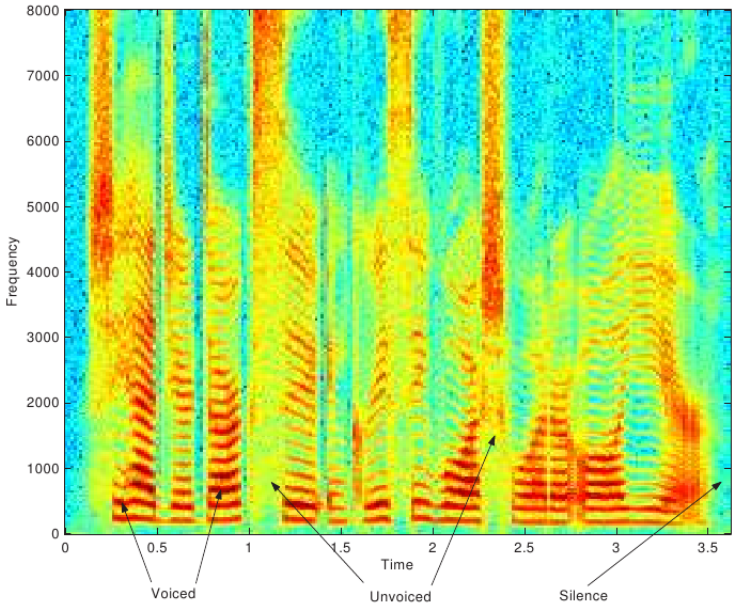

FIGURE 3.5: Spectrogram of "She had your dark suit in greasy wash water all year" [3].

reality, speech is non stationary. Still, this assumption is made in most of speech systems. Along with this assumption, we also assume that some slowly varying parameters of vocal tract carry important information of speech. This is mostly true for non tonal language such as English, where pitch does not carry information.

For this project, the spectrogram is performed on the segmented digits using 'specgram' functionality of the matplotlib Python library. The audio is read from the provided CUAVE data files and is converted to an array. The algorithm is set to traverse through the whole duration of speech in steps of 10 milliseconds. In each iteration of the algorithm, 20 milliseconds of speech are read. It was found by analysis that centering the signal to the mean and scaling to unit variance provides a better performance. This step makes sure that the dynamic information is preserved while variation in volume is compensated for. This is important as the energy of the speech signal differs between the subjects and within the same subject's audio to a large extent. In experiments conducted in this project, it was found that this energy fluctuation has a considerable effect on the accuracy of the system. Thus, the time input signal is scaled to have a normalized signal at the input of the spectrogram module. After the scaling, the spectrogram is calculated and a detrend on mean is applied on the input signal internally to spectrogram. The effect of normalization is shown in Figures 3.6 and 3.7.

Figure 3.8 shows the initial processing stages of audio frames to generate spectrograms. In [8], it is mentioned that 161 frequency bins are used in the spectrogram. Considering how the magnitude spectrum is computed, each audio frame should have 320 samples. As per [8], it is given that each frame lasts 20 ms. Thus, the sample rate of the audio

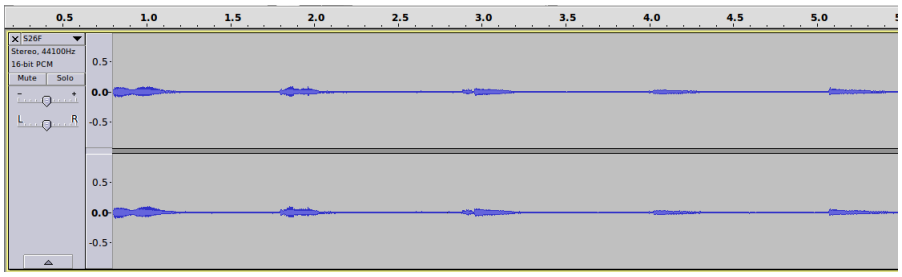

FIGURE 3.6: Speech of a subject with low energy level, which makes it necessary to apply normalization

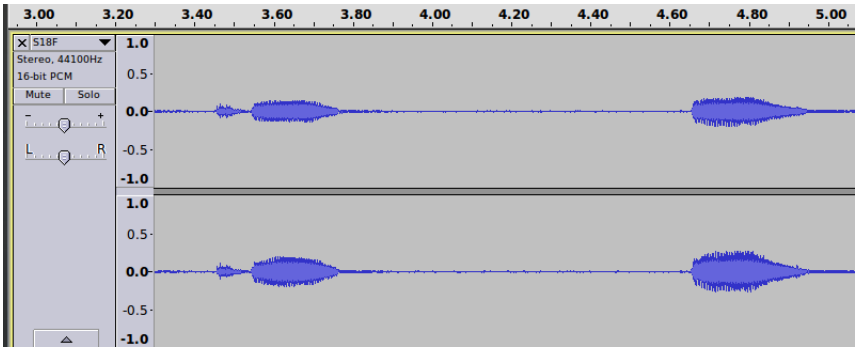

FIGURE 3.7: Speech of a subject with good energy level, which makes it easy to detect even without normalization

has to be 320 samples / 0.02 seconds = 16 kHz. So, the audio present in the original CUAVE video data needs to be downsampled to 16kHZ from 44.1kHz. The window for the spectrogram is chosen to be 'Hanning' window which is helpful in achieving a very low aliasing. Spectrogram generates 161 dimensional vector for each frame. Each frame has two temporal derivative frames. These are obtained by computing the element-wise difference of current frame with the previous and the next frames. This results in a 483 dimension vector for each of the audio frames.

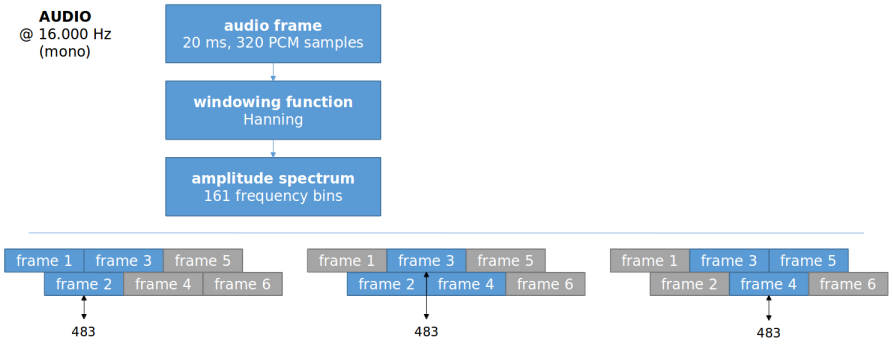

FIGURE 3.8: Audio preprocessing initial stages

The spectrogram application generates a matrix, a mat file, for each of the subjects for

the whole duration of the natural stand still in frontal facing. This mat file is generated one time for each subject and is reused for the whole cross-validation experiment. For experiments requiring changes in the audio modality, the spectrogram files are regenerated for the corresponding test subjects.

3.1.1.3 Video Preprocessing Modules

The first step in video processing is to extract the mouth regions, which is the Region of Interest (ROI) of the video modality. In [8], an off-the-shelf mouth ROI extractor from Dalal & Triggs, 2005 is used with a median filtering over time to extract the relevant mouth regions. The mouth regions are rescaled to 60 x 80 pixels before those are provided to the later stages of preprocessing. This stage is unique to video preprocessing section.

The mouth region extractor from Dalal & Triggs, 2005 is not available openly for use. So, a Region of Interest extractor is developed from a face recognizer that is available openly, for this project. The following section explains various aspects of the Region of Interest extractor. The implementation is further detailed in the corresponding appendix.

Region of Interest Extraction

ROI extraction is the first step in the visual feature extraction stage of an Audio-Visual Speech Recognition system [52]. One important requirement for this stage is low computational load on the system. To have useful visual features for the recognition system, it is important to have a dependable and robust estimation of mouth region of interest from the video stream of the input data. The resultant visual features from the preprocessing stage should acquire the cues used by the human brain in distinguishing speech. There are two major categories of ROI extraction for mouth ROIs: a) methods using image processing techniques, and b) methods using statistical modeling. Here we are going to use one reliable mouth ROI estimator which consumes comparatively reasonable resources.

For this project, the off-the-shelf object detector that we use is from Dantone et al [53]. It uses Conditional Regression Forest, which learns relations conditional to global face properties. Each tree in the forest is trained on a random sub-sampled training set, the use of which avoids over-fitting. When provided with a face image from the training set, the detector will learn conditional probabilities in parameter space. For detection, the probability of the global properties is estimated and is used to select a set of pre-trained conditional regression trees. The trees are selected in such such a way as to predict the

facial features. The algorithm for detection starts with finding the position and size of the face and then 'patches' are sampled inside the bounding box of the face. These patches are then fed to the tree mentioned above and are passed through the tree until each patch ends at a set of leaf nodes. Later, the probability of a leaf to be a positive leaf for a facial feature point is used for density estimation for the feature point at a pixel location. A mean shift for each point will provide the final facial feature points. As an example, consider regression trees that are trained conditional on several head poses. The probability for a head pose is calculated from the provided image and the corresponding tree is used for estimation of the facial features. Thus, the tree selected for each image can be different based on the properties.

Using the above mentioned facial feature detector, the mouth ROI coordinates are also determined. This is used later on in the program flow to extract the mouth ROI from the frame array.

Stabilization of Mouth ROI

As per [8], over the extracted mouth regions, a median filter is used. In addition to the median filter, it is important to resize the mouth ROI to make it a 60 x 80 pixel array. Since the mouth shape and the relative size vary between subjects and based on what they are saying, it is important to fix the aspect ratio. Once the aspect ratio is fixed the ROI can be resized to 60 x 80 element array as required by the project. The median filter and its application are explained in the section below.

Median Filter

The median filter is needed at the output of ROI detector to ensure that any noise in ROI values, which corresponds to a wrong ROI is filtered. The median filter is a discrete time process in which a window is stepped through an input signal [4]. The window should have n frames, where n is any integer and is selected based on the scenario. On each iteration, the values within the window are ranked. The output of the filter is the median value of the corresponding set of values within the window. For this project, the mouth ROI is a rectangular box which is defined by x and y (the coordinates of top left corner of the ROI), h, the height and w, the width of the corresponding ROI. The median filter is applied on each of these variables x, y, h and w. For each the variable, including the current value and the previous values, total n values are chosen. Initially, when the number of available values is less than one, the window size is made equal to

the number of available values. It was verified that this does not affect the accuracy of ROI by displaying the ROI for all the subjects.

For the selection of value n, a few properties of the median filter need to be considered [4]. The window size decides the flexibility of the median filter with the changes in the input. If the window size is big, then the impulses that correspond to wrong values of ROI will be filtered efficiently. Still, along with that, the actual variations in the position of mouth will be removed. If the window size is small, then the impulsive changes in ROI will have an effect on the stability of the output. By checking various values of window size n, for the set of frames where head movement is more, it is found that n=5 provides a reliable median filter. The effect of window size of median filter is explained in Figure 3.9.

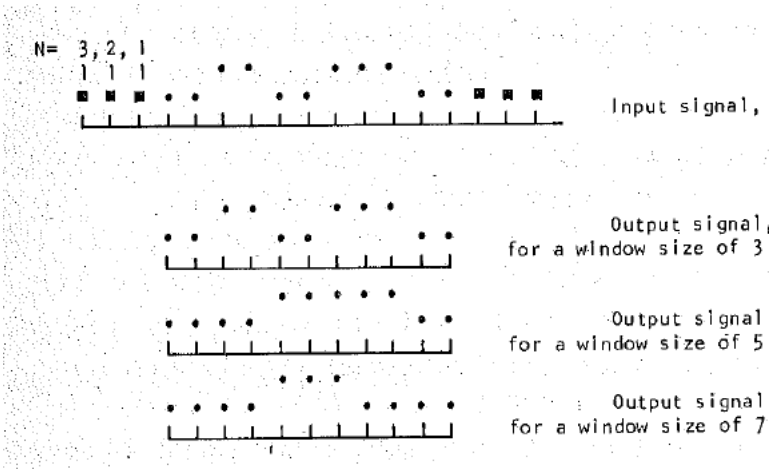

FIGURE 3.9: Median Filter: Effect of various sizes of n on an arbitrary input signal [4].

As can be seen in Figure 3.9, with value of n=3, the output signal is just equal to the input. Also, with values of n=5 and n=7, it can be seen that the changes in input signal reflect at the output, still, not in an impulsive way. For n=11, all such variations are filtered out and the output signal will be a constant.

3.1.2 Common Modules of Preprocessing Stage

Following modules are used for both audio and video preprocessing. These are the final few stages of preprocessing and deliver the input vector to the RBM/DBN. The output of the RBM/DBN is further processed by average pooling after which, it is provided to the classifier.

Note that, though the modules are common, the order in which these are present is different for audio-only and video-only systems, as explained before.

3.1.2.1 Temporal derivatives

As per [8], the audio signal is represented using a spectrogram and the temporal derivatives. The resulting vector for audio has 483 elements. For video, the mouth ROI that is extracted as explained in the previous section is reduced in dimension using PCA and temporal derivatives are calculated on that. In either case, one set of features are directly calculated from the input data and then the temporal derivative features are computed for every two frames to have the frame level features.

For the video modality, temporal derivatives are calculated after converting the image (frame) to gray [54]. So, the temporal derivatives are just pixel by pixel gray level difference between frame T_f and T_{f+1}. These features obtained via taking differences are also called delta features. In [54], it is shown that the use of delta features improve the performance of a visual speech recognizer. Also, the simplicity of these features reduces the complexity of the speech recognizer to a large extent. It was shown in [55] that the visual features combining pixels and pixel by pixel differences between consecutive frames perform well in conjunction with linear SVMs.

A key property of natural systems is that it cannot change its state instantaneously [56]. So, its dynamics carries information. The characteristics of time variations of speech spectra is called speech spectral dynamics. For human speech production, inertia in vocal organs produces coarticulation, causing dependencies between adjacent phones which is seen in its short term spectrum. The dynamics of temporal derivatives of the speech power spectrum acts as a carrier conveying information. The information is contained in the modulation spectra of the speech signal. Linguistic information of speech is depicted in the dynamics of envelopes of the frequency subbands. This property was validated by physiological and psychophysical experiments conducted on human speech perception. Without the use of such spectral dynamics of speech, the conventional ASRs were sensitive to linear distortions, reverberation in closed rooms, frequency localized noises and speaker characteristics. Human speech perception can handle these issues well because it makes use of speech dynamics.

As per [57], psychophysical studies show the significance of temporal structure for intelligibility of speech. Intelligibility depends on the integrity of amplitude of the modulation spectrum especially between 3 to 16kHz. In this paper, it is shown that the complete elimination of temporal modulation resulted in speech intelligibility to be 5%. The temporal envelope has variations at different rates which are different speech features. Such speech features include phonetic segments, syllables and phrases which ranges between 50Hz to 0.5Hz.

3.1.2.2 Principal Component Analysis (PCA)

PCA is a multivariate statistical technique for reducing the feature space and helpful in data compression [58]. This is achieved by generating linear weights of the original input feature vector by which a transformed feature space of the following attributes is created:

- Every PCA vector (principal component) has a maximum variance of the input feature that is not modelled by any other PCA vector
- Sequentially, PCA vectors represent a reduction in the magnitude of input feature variance
- PCA vectors have no correlation with each other

PCA is computed by taking covariance matrix of the input feature set [58]. The PCA weights are the eigenvectors of the covariance matrix. The eigenvectors and the eigenvalues can further be used to transform any input feature vector with arbitrary covariance matrix. This provides a reduced input feature vector that has no correlation among components. Such a process is called PCA whitening.

It has been noted in [59] that PCA whitening can reflect the statistics of speech in very reliable manner. PCA was found apt for applications such as speech enhancement and dealing with room reflections in microphone array systems, where it was used on FFT spectrum. With the use of PCA the curse of dimensionality issues are less of a problem as, dimensionality reduction can be performed at the very front end of a speech recognition system [60]. It has also been shown that using PCA on FFT spectrum of speech makes the system robust in noisy environments.

With the visual modality, PCA whitening generates features known as 'eigen-mouths' which are eigenvectors from a set of mouth ROIs [61]. This projection using PCA, makes it easy to compare a mouth ROI as it is represented as a weighted sum of the eigen-mouth features. So, here for the classification purpose, the weights of the ROI under consideration are compared with the ones that are already known. This approach provides the ability to learn and later recognize the mouths in an unsupervised manner. In [61], it is shown that the eigenface technique has high-accuracy and is well suited for application such as face recognition, human-computer interaction and security systems. Similar to the eigenface technique, here also the test data is taken into the same subspace spanned by eigen-mouths and is provided to the system for recognition of speech.

Following Ngiam et al [8], PCA is used for dimensionality reduction for this architecture. As mentioned before, after generating temporal derivatives, the combined vector of audio

signal and temporal derivative make a vector of 483 elements. This is further reduced to 100 element vectors by the use of PCA. On the other hand, with video, the reduced vector is obtained on the original raw video frames. Later, temporal derivatives are taken on the reduced vector. In case of visual input, [62] shows that most of the variance is in the original data is preserved by PCA along with providing reduced dimensionality of the data. It was also shown in this research that reducing mouth ROI to a vector containing 30 to 60 principal components achieves peak performance. For further experiments, 32 principal components were used in this research. Ngiam et al [8] used the same number of principal components for representing mouth ROI for the architecture discussed in here.

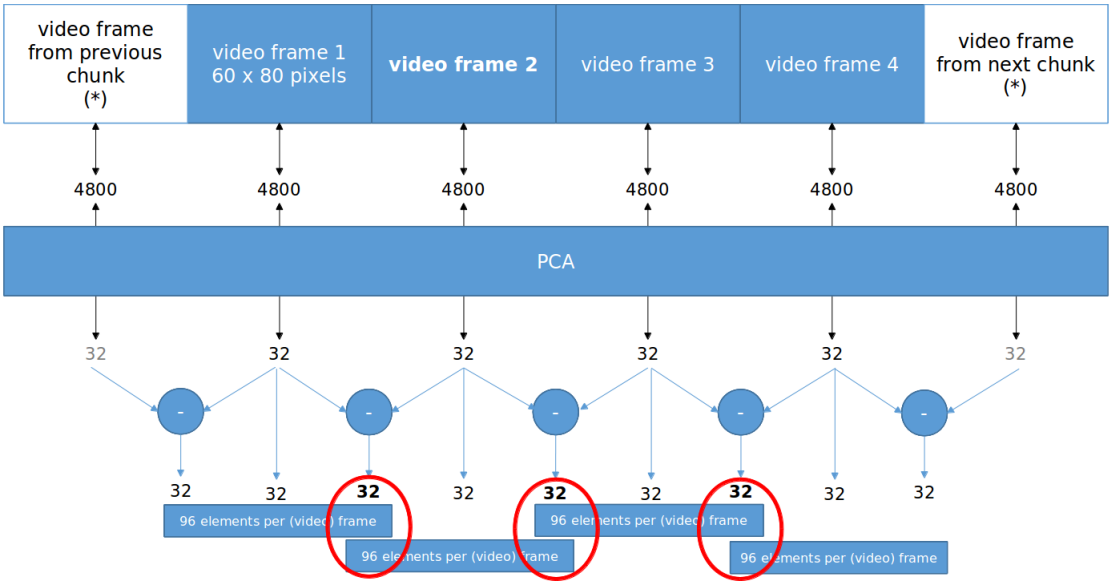

FIGURE 3.10: Video preprocessing initial stage

Figure 3.10 shows the initial stages of video preprocessing. It can be seen that at first PCA is performed on the video frames. Temporal derivatives are obtained from these whitened frames. The PCA whitened frames, along with the temporal derivatives are combined to created an input array for the video. This is further passed into later stages of the input preprocessing. The red circle shows the temporal derivatives shared by the consecutive frames. Note that since the temporal derivaties are calculated for each of the frames seperately, a 32 x 12 (384 element) element vector is generated from 4 video frames.

Figure 3.11 shows the stages of audio preprocessing after the spectrogram. As can be seen, for each frame, temporal derivatives are found before PCA whitening and whitening is performed on the combined vectors of frames and temporal derivatives.

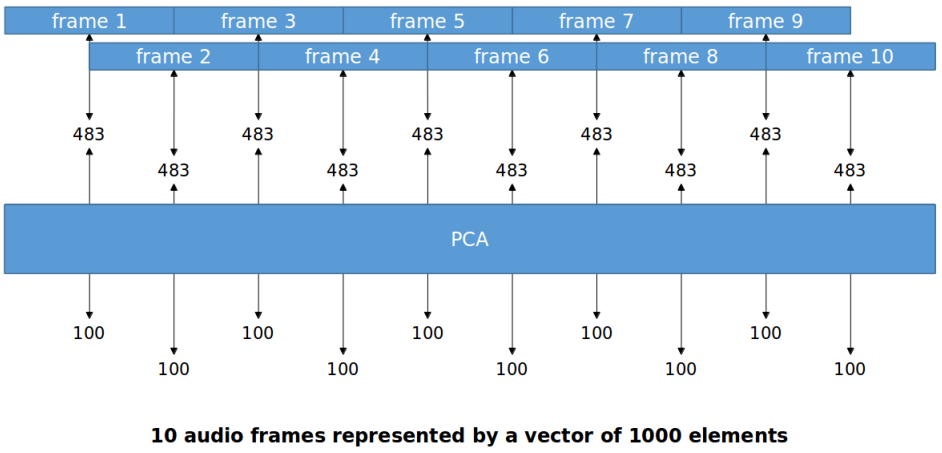

FIGURE 3.11: Audio preprocessing stages after spectrogram

3.1.2.3 Feature Mean Normalization over Time

Feature Mean Normalization (FMN) is the simple subtraction of feature mean calculated on the full length of an utterance, in this case, the digits [63]. This removes the DC component from the data, which does not carry information in the case of audio and video [8]. It is helpful in improving performance in both visual and audio speech recognition tasks. In [64] and [65], it is shown that FMN improves performance in the case of multi-speaker and speaker-independent lip reading tasks.

Feature Mean Normalization over time is performed on the whitened output from video and audio preprocessing stages. A sklearn library function is used to normalize the data over the mean. After FMN, the training and test data is provided to the RBM or DBN to generate representations. These representations are fed to the next stage, average pooling, before it is provided for classification.

3.1.2.4 Setting Up Input for RBM/DBN

It is seen that the duration of video frames is equivalent to the duration of 10 audio frames [8]. So, the input to the input layers of the Video and Audio RBMs will have preprocessed data of 4 and 10 frames respectively. With video, 4 frames, which are PCA whitened along with the temporal derivatives of these constitute the input. So, at the input of the Audio RBM, we have a 1000 element vector whereas for Video RBM input it is a 384 element vector. It is observed that for the same duration of data, audio is represented by a larger vector compared to the visual. This lesser size of the visual

modality vector in spite of more dense data for the visuals is justified, since speech modality is of the prime importance for speech perception.

As per [57], the visual cues of articulation are loosely bounded to the auditory input. Also, the visual cues can precede the auditory signals by around 100 milliseconds. This justifies the choice of grouping 4 visual frames with 10 frames of audio for bimodal learning. Such a strategy will provide the system an ample amount of information to correlate the visual modality with speech modality.

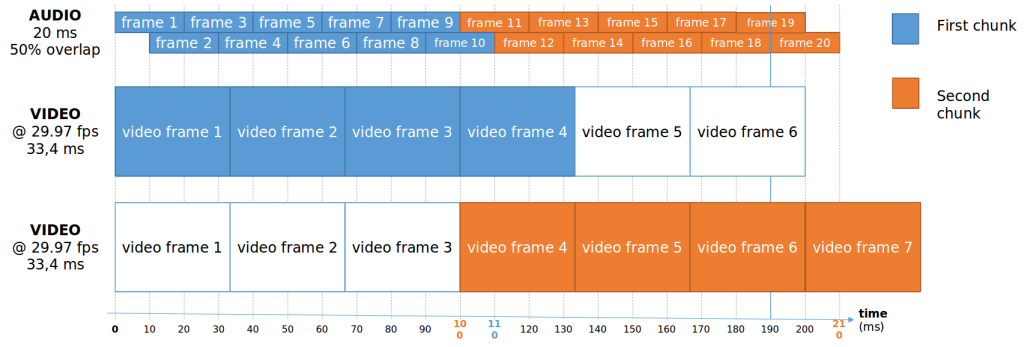

FIGURE 3.12: A visual representation of the segmentation

As can be seen in Figure 3.12, duration of 4 video frames is approximately equal to that of 10 audio frames. Since the frame rate of CUAVE video for video frames is 29.97 fps (frames per second), the length of each frame is approximately 33.4 milliseconds. With 10% overlap the duration of one audio frame is 20 milliseconds. In figure 3.12, it is clearly seen that the first 4 frames of video approximate duration of the audio. Similarly, from almost the end of fourth frame of video where the 5th frame is displayed, till the end of 7th frame where the 8th frame is just displayed,video matches with next the 10 frames of audio. Thus, 4 frames of video constitute one column in the input array for RBM/DBN. Similarly 10 frames constitute the input array for audio. Thus, input array will be a 1000 x n array for audio and 384 x n array for video, where 'n' is the number of sets of frames, for a single digit. All such digits can be concatenated to form two arrays, one for training and one for test. The RBM/DBN is initially trained with the training data generating models that are further used for creating training and test representations.

3.2 Neural Networks

From [66], it is known that the speech signals are sparse. This results in amplitude modulated noise having less masking effect on speech than any stationary noise, such

as cafeteria noise which is naturally occurring. The sparseness of speech is described in terms of the discrete harmonics contained in the speech and the periods of relative silence on closure of vocal tracts. This implies that the speech spectrogram, which is a matrix in frequency-time, has low probability for a frequency-time cell to contain speech. So, the matrix is sparsely populated with speech energy. This effect is seen in the spectrogram in Figure 3.5 which was discussed in the previous section. In computational neuroscience, sparse neural code signifies sensory stimuli that are encoded by powerful activation of a small subset of a large population of neurons. In matrix algebra a sparse matrix is a matrix that is mostly populated with zeros. The above three definitions are overlapped in terms of relevance in the current context. As per Ngiam's [8] architecture, a sparse RBM is used, which is known to learn significant features with digits and natural images. The details of the RBM is discussed in the next chapter.

A RBM has a layer of visible units v, that receives input and is connected to a layer of hidden units h, by symmetric connections represented by a weight matrix W. For a Gaussian-Bernoulli RBM, that is used in this architecture, a k dimensional real-valued input data is modelled using a undirected graphical model that has n binary (Bernoulli) units. The negative log probability of any state of the RBM can be represented using the energy function that follows:

$$-logP(\mathbf{v},\mathbf{h}) = \frac{1}{2\sigma^2}\Sigma_i v_i^2 - \frac{1}{\sigma^2}(\Sigma_i c_i v_i + \Sigma_j b_j h_j + \Sigma_{i,j} v_i w_{ij} h_j)$$

Here v_i and h_j are ith and jth unit of visible and hidden units respectively. σ is a parameter, b_j and c_i are the biases respectively for the hidden units and the visible units. w_{ij} is the symmetric weight between the i-th hidden and the j-th visible units. This maximum likelihood estimation problem learns w_{ij}, c_i and b_j in order to minimize the energy for the states from the data distribution. On the other hand, it increases the energy of the states that are not probable. The goal of this problem is maximizing the log likelihood of the data.

From this model, it is possible to learn the conditional probability distributions, preserving h or v fixed. It is represented in the following mathematical expression:

$$P(v_i|h) = N(c_i + \Sigma_j w_{ij} h_j, \sigma^2)$$
$$P(h_j|v) = logistic(\frac{1}{\sigma^2}(b_j + \Sigma_i w_{ij} v_i))$$

Here, N(.) is the Gaussian density function and *logistic* represents the logistic function. In this problem, to have sparse hidden unit activations, a regularization term is introduced. This term penalizes any divergence from a fixed level of activation, expected for the hidden units. From this, the following optimization problem is posed for the training set comprising of m elements:

$$minimze(w_{ij}, c_i, b_j - \Sigma_{l=1}^{m} log \Sigma_h P(v^{(l)}, h^{(l)}) + \lambda \Sigma_{j=1}^{n} |p - \frac{1}{m} \Sigma_{l=1}^{m} \mathcal{E}(h_j^{(l)} | v^{(l)}|^2))$$

Here, $\mathcal{E}$ is conditional expectation given the data, λ is a regularization constant, p controls the sparseness of hidden units and is a constant. Here an algorithm, namely, contrastive divergence, is used for computing the gradient of the log-likelihood term, that is expensive with gradient descent algorithm. This algorithm provides a reliable approximation of the the gradient of the log-likelihood. On each iteration, contrastive divergence update rule is applied. After the update, gradient descent is performed using the gradient of the regularization term. The algorithm is detailed in the appendix.

Using the sparse RBM explained above, it is possible to learn the features of audio and video separately. This is the strategy used by [8], in this architecture, as the first step. On learning, the posteriors of the hidden variables given the visible variables are obtained. These posteriors are used by the classifier as a representation of data for learning and classification.

For the implementation of neural networks for this project, the Deepnet library is used [67]. The reference stack used for this project is built using Deepnet library developed by Nitish Srivastava in 2012. It contains open source implementations for commonly used neural networks such as Feed-forward Neural Networks, Restricted Boltzmann Machines, Deep Belief Networks and Autoencoders. It is built on the top of cudamat library — that provides matrix operations on CUDA enabled GPUs — by Vlad Mnih and cuda-convnet library by Alex Krizhevsky. All these implementations are performed in Python which makes it easy to use and adapt for the requirements of the rest of the system, which is implemented in Python. Details of the implementation of neural networks using this library is detailed in the appendix.

The output of RBM/DBN goes to an average pooling module which provides a fixed size output. Thus, this module interfaces the variable size output of the neural networks to the fixed size input of the linear SVM classifier.

3.2.0.1 Average Pooling

Pooling in general provides invariance, more compact representations and robustness to noise and clutter in images [68]. There are several different types of pooling such as average pooling, max pooling and some combinations rules of these. Pooling transforms feature representations into one that retains important information and disposes of any irrelevant information. In [68], it is demonstrated that by selecting the right kind of pooling, it is possible to make simple systems of features and classifiers competitive to the complex ones.

The first step after the creation of a representation through RBM or DBN is to perform an average pooling [8]. It is a must to use pooling, as, the linear SVM for classification can handle only fixed sized input vector as training and test data. For any digit, the frames are overlapped and average is taken which is called average pooling. This results in a fixed size input for the SVM. Performing pooling along each row of the input vectors transforms it to a single vector of equal dimension. This also makes the vectors of different sizes of speech or visuals compact and provides a unified dimension. Each digit is sliced into three and for each of the slices, average pooling is performed. These four sets of averaged data are concatenated to form the input. This is applicable to all different forms of representation handled in this thesis, as, the classifier is always a linear SVM. Thus, the average pooling is used in audio-only, video-only and multimodal systems.

In the implementation level, average pooling was performed based on the implementation suggested in [69]. The representation that comes out of RBM or DBN contains a two dimensional array. This array has rows representing the features while each column represents each set of frames of the digit. Averages of all rows are taken as a representative of all the frames in the digit, resulting in a single column vector. This reduces the size of the data and thus, the overall computational complexity of the classification system. Pooling summarizes the feature distribution of the data under consideration into a statistical representation. This results in rejection of noisy elements as the noise is diluted due to averaging.

Figure 3.13 shows the FMN performed in this investigation. Here 'FULL' indicates the whole digit and A1, A2 and A3 are the three equally sized slices of the digit. Using a sliding window on the digit array, average pooling is performed. As shown in the figure, if the digit has 9 sets of frames, then each of the slice will have three frames each. After average pooling, the 4 sets - FULL, A1, A2 and A3 will generate F, F1, F2 and F3 correspondingly. These are concatenated linearly to form the input vector for the classifier.

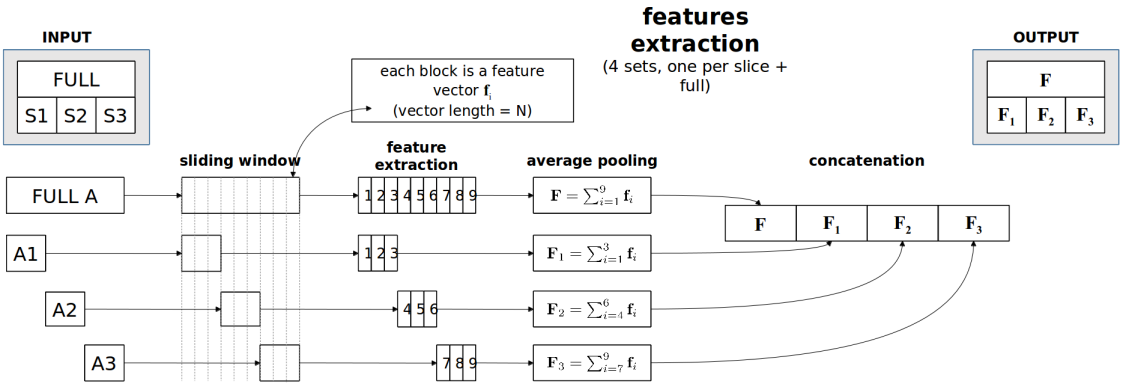

FIGURE 3.13: Feature mean normalization on video and audio

An important advantage of average pooling over other pooling methods is its robustness towards intrinsic variance. In the distribution of input data, intrinsic variance is caused by sampling from a finite pooling of data [69]. Pool cardinality represents the number of set of frames used for averaging. Since pool cardinality is large enough for this project, average pooling makes the system robust to intrinsic variance compared to any other pooling. This happens as the variance of average always decreases with increase in pool cardinality.

3.2.0.2 Setting Up Input for Classifier

As explained in the previous section, the preprocessed data is normalized and averaged to get fixed size chunks of data [8]. One chunk represents the whole digit whereas other three chunks represents three equal slices of the digit. For video, each chunk has 1536 elements. On concatenation as shown in Figure fig:classification, the chunks forms a 6144-sized vector, which acts as the input from video RBM to the classifier. For audio-only system, correspondingly, the vector size is 6000. For the multimodal representation, the vector size is 24216. This large size for the multimodal data is reflected in the time consumption of the classification task using multimodal representation.

The classifier takes this fixed size vector in the previous stage along with the corresponding labels and uses these for the training of classifier and classification of the test data. The labels are output from the linear SVM, which can be used for the computation of the performance externally. The results of these classification tasks are discussed in the forth coming sections. The following figure 3.14 explains the vector size of data in each stage. From top to bottom, it shows the input vectors for linear SVM from audio RBM, video RBM and multimodal DBN.

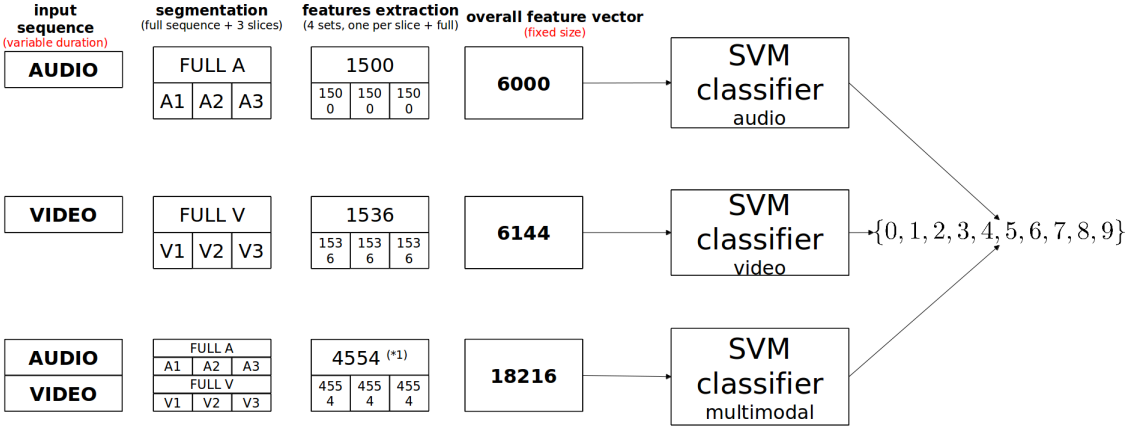

FIGURE 3.14: Setting up input for classification

As per [56], the dynamic features that reflect the dynamic trends of speech is to be calculated from spectral envelopes of larger segments of the signal. With a segments around 100 milliseconds it is possible to calculate the dynamic features. It is seen in certain systems that the syllable level spectral dynamics can be learnt from rather long speech segments (around 200 milliseconds). By choosing three slices for each digit, the speech segment size for each slice is around 200 milliseconds, for typical duration of the digits. Modulation spectrum is defined to be the structure of temporal curvature of components of spectral envelopes of the given speech. This reflects the temporal motions of vocal organs. By choosing this strategy, the modulation spectrum may also be learnt by the feature learning process.

3.3 Classifier

This is the final stage of all the three the systems (audio-only, video-only and multimodal). In early researches it was found that linear SVMs are promising classifiers for visual speech recognition tasks [54].

In many of the deep learning models, softmax activation function was used for prediction tasks. But, it is shown in [70] that replacing softmax layer with linear SVMs have several benefits. With linear SVMs, the learning process discovers the maximum margin between the data points of the given classes. In that place, softmax layer minimizes cross-entropy or maximizes the log-likelihood. In this research, it is shown that various types of linear SVMs yield significant gains in prediction over softmax layer. This was tested against several popular deep learning databases such MNIST.

In [71], it is shown that the tractability of large scale training is not promising with non-linear kernel machines. This is due to their computational complexity. In wang2013towards,

it is shown that the use of linearly separable and discriminative features obtained from raw acoustics and linear SVM is much faster and simpler than non-linear kernel machines. This proposed systems was tested in various acoustic conditions and was shown to be reliable.

One of the most popular packages for linear SVM is made available in the libSVM library. Such libraries implement linear SVM using one of the popular multi-class SVM methods . There are mainly three such methods, namely one-against-all, one-against-one and DAGSVM (Directed Acyclic Graph Support Vector Machine) [72]. One-against-all is a bit archaic method, which was popular once.

The one-against-one method is used in the linear-SVM library used for this project. One-against-one method constructs $\frac{k(k-1)}{2}$ classifiers and each of the classifier is trained on data from two different classes [72]. For testing, based on the sign of the decision function, voting is performed for the classes. One vote is given to one of the two classes under consideration based on the sign. Once the voting is over for all the classifiers, the class with the maximum vote represents the class of the test data. If there are equal number of votes for multiple classes, the one with lower index is selected.

The training phase of DAGSVM is similar to that of one-against-one method [72]. Whereas, in the case of testing, a direct acrylic graph is used. Based on the results of a binary evaluation function, a path in the graph is chosen till a leaf node is reached, which represents the predicted class.

Algorithm for the classifier using libSVM for this project is explained in the appendix A.

This chapter explained the detailed structure of the reference system and implementation of the reference system at a top level. In the next chapter, experiments that are conducted based on the research questions, are explained.

Chapter 4

Experiments and Results

4.1 Experimental Framework

The following sections describe the experimental setup used for this thesis project. An MNIST object recognition experiment is used to validate the ability of the Deepnet library networks to generate reliable features. Then, a few experiments based on [8] are performed on the reference stack to create a benchmark. Grid search experiments are performed on this reference stack to fine tune the network, to the get the required level of performance. Finally, experiments based on research questions are performed and the results are tabulated and discussed.

4.1.1 MNIST experiments for Deepnet library setup

The Deepnet library, which is used in the reference architecture, is downloaded and set up. However, the reference architecture mentioned here has several stages of preprocessing and several layers of the neural networks. Thus, it is wise to verify the library setup using a simple experiment. With this step, it is possible to verify if the RBMs generate a reliable representation and is able to learn the features from one of the concerned modality.

For this purpose, a framework, namely 'Restricted Boltzmann Machines with SVM for Object Recognition' is used [29]. With this framework, a reduced representation of the data is obtained using an RBM and classification of the generated representation is carried out using an SVM. The SVM estimates the classification ability of the features generated by the RBM, where the RBM is set up using the Deepnet library. The framework is meant to show the powerful learning ability of the RBM and its ability to attain a high recognition rate.

4.1.1.1 RBM Dimension Reduction

An important virtue of the RBM is its ability to non-linearly reduce the dimensionality of any high dimensional input data [29]. Such dimensionality reduction can promote classification, visualization, communication and the storage of high dimensional information. The reduction in dimensionality depends on the number of visible and hidden units of the RBM. The set of states of the visible layer is regarded as the sample observation. The hidden layer is a feature representation layer that generates features of the given samples using the hidden states. Thus, in this context, the number of hidden states is less than the number of visible states to obtain a reduced number of features.

RBM is an unsupervised learning technique that produces a new, unknown distribution denoted as P(v) of the data from the given samples [29]. P(v) is learned through the following steps:

1. Modelling of P(v) using the hidden parameters, θ.
2. Calculate increments for parameters, namely, $\Delta\theta$
3. Calculate value of θ by the use of gradient descent combined with maximum log likelihood. Thus the parameter θ is given by,

$$\theta^{t+1} = \theta^t + \tau\Delta\theta \tag{4.1}$$

Here, τ is the learning rate and $\Delta\theta$ is the increment calculated in step 2.

As the theory behind RBM is explained in the previous chapter, this section only abstracts the learning algorithm.

4.1.1.2 Database for Validation of RBM-SVM Setup

For the purpose of validation of the RBM with SVM — RBM for the creation of representations and SVM for the classification purpose — the MNIST database is used [29].The MNIST database contains 60,000 training samples and 10,000 test samples. Each of the images has a size of 28x28 pixels which is centered and size-normalized. Here, for this experiment, the whole test set is used. However, for the training, 1000 randomly chosen images from the whole 60000-digit training set is used. Cross-validation is performed by randomly choosing ten 1000-digit training sets. The accuracy is calculated by averaging the accuracies of all the sets. The MNIST images are not in any standard format. Thus, there needs to be a program to read the images. Figure 4.1 shows a visual representation of of MNIST digits.

FIGURE 4.1: A visual representation of MNIST digits

MNIST, which is a diverse dataset, was formed by modification of a parent dataset. For this, images in the dataset NIST, which is the superset of MNIST, are normalized and fitted in 20x20 pixel box. Then, the center of mass of these images is calculated. The images are translated into 28x28 by positioning the center of mass at the center through translation. These images contain equal number of handwritten digits collected from NIST's Special Database 3, and Special Database 1, referred to as SD3 and SD1 respectively [29]. SD3 contains handwritten digits from the Census Bureau, whereas, SD1 contains handwritten digits from high school students. Since MNIST is a mix of these two datasets, it is possible to reach reasonable conclusions in experiments compared to the original database (NIST database). The MNIST training dataset comprises 30,000 digits from SD3 and the same number of digits from SD1. Similarly, the test set of MNIST contains 5000 digits each from SD3 and SD1. The digits in the training and test sets are disjoint. The training set consists of writings from 250 different people.

4.1.1.3 RBM with SVM

SVM is one of the widely used methods of machine learning and it uses statistical learning theory [29]. The strong generalization ability of the SVM is helpful for the kind of task performed here. Note that the theory behind SVM is detailed in the previous chapter. The framework here uses a SVM nonlinear classifier for the task under consideration. Two key features of SVM include the capability to map data into a higher-dimensional

space using a linear or nonlinear kernel function, and its ability to make non-linear data at the input, linearly separable.

At the implementation level, libSVM is used as the classifier in this framework for MNIST digit recognition. In the case of RBM with SVM, non-linear kernel is used for the classification purpose. The recommended non-linear kernel for libSVM in this case is a radial basis function. The block diagram in figure 4.2 explains the module-wise flow of this framework.

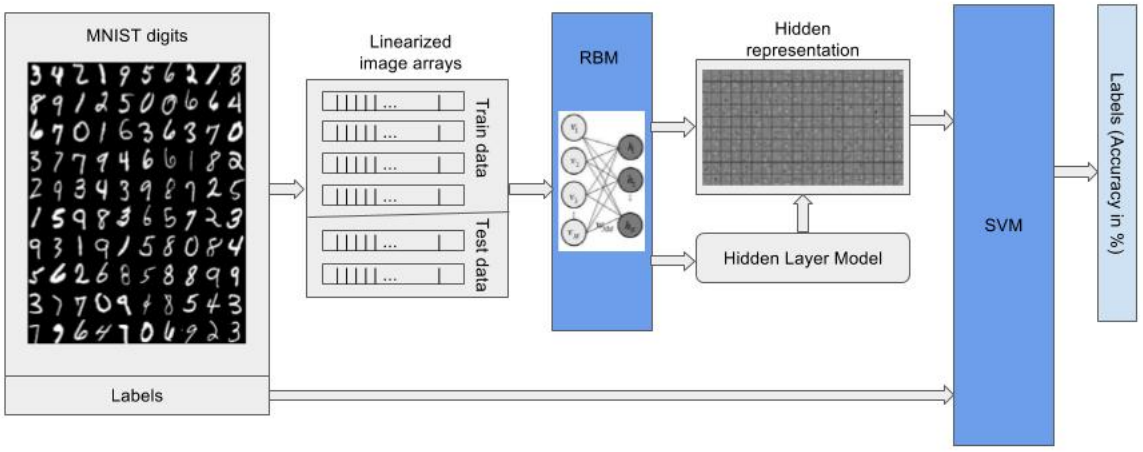

FIGURE 4.2: Modules involved in generation of representation and classification using RBM-SVM

Since RBM uses unsupervised learning, labels are used only with SVM when classification task is performed. The hidden layer model is generated by the RBM using the training data. Later, this model is used to generate transformed representations for both training and test data. Finally, libSVM predicts labels for the given test data using the inputs — transformed training data, its labels and the transformed test data. It also computes the accuracy when the test labels are provided. The number of training digits used for the training of the RBM and the classifier is 1000, whereas, the number of test digits used is 10000.

4.1.1.4 Validation

On execution of the RBM-SVM on MNIST digits, the representations were generated and the results of classification are captured in Table 4.1.

TABLE 4.1: RBM-SVM performance of MNIST dataset

Index	Accuracy (%) (linear kernel)	Accuracy (%) (non-linear kernel)
1	88.66	88.59
2	89.17	83.78
3	90.38	86.52
4	89.65	84.27
5	89.21	82.61
6	89.17	83.7
7	89.98	84.06
8	89.36	84.22
9	89.31	83.01
10	89.66	85.64
Average	89.46	84.64

It is seen from the result that the average accuracy over cross validation is close to the reference result that is 91.3% [29]. The difference (6%) in performance can be attributed to the lack of fine tuning of the RBM for this specific task. Also, with the linear kernel, which is used for answering thesis questions in this project, the performance seems to be promising. This validation confirms that the RBM is able to generate reliable representations from the image modality. When video is given to this network, the frames are treated as images. This implies that its performance can be expected to be good with the video modality of the reference architecture. As explained in the previous chapter, the pipeline of the reference architecture has average pooling between RBM and SVM stages. This step makes the pipeline for the reference architecture different from this experiment. Still, this experiment assures that the setup of RBM and SVM is reliable, though, fine tuning may be required for better performances.

As, the setup for RBM-SVM is confirmed to generate reliable representations, the next is to set up the unimodal-systems (Audio-only and video-only systems).

4.1.2 Audio RBM

One of the first steps in learning a multimodal representation is to have a unimodal RBM set up for each of the modalities. For that purpose, audio RBM is set up for which the input is audio after passing through the preprocessing stage as described in the previous chapter.

As seen in Figure 4.3, the preprocessed audio is given as input to the visible layer and, the hidden layer outputs a representation of the input data [8]. This process extracts features of audio, which are better represented as hidden states. Here the visible units

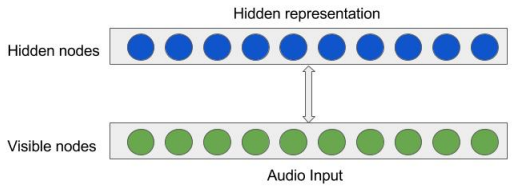

FIGURE 4.3: Audio RBM

are real-valued and the hidden units are binary units. The parameters of the model are learnt through the use of contrastive divergence. For regularization of the model for sparsity, a regularization penalty is used that determines the sparsity. Note that the hidden representations are larger than the inputs, as, this architecture uses overcomplete representations. Here, the hidden layer is 1.5 times bigger than the input layer. The number of nodes of the input layer is 1000 and the hidden layer is 1500.

In overcomplete representations, the number of hidden nodes is greater than the number of the input nodes. Overcomplete representations are known to have greater robustness in the presence of noise [73]. It is sparser, and have greater flexibility in matching structures in the data. Overcomplete codes were proposed as a model of certain response properties of neurons in primary visual cortex. Note that the neural networks discussed from now on uses overcomplete representations in all the layers.

The block diagram in figure 4.4 shows the block wise flow of the audio RBM-SVM framework [8]. This block diagram presents a block-wise flow of the audio-only system and is an abstract of the flow diagram for audio-only system in the previous chapter. Here the audio signals are segmented into digits and the spectrogram is generated on those segmented digits. Then, temporal derivatives are obtained from the spectrograms. These derivatives in combination with the actual signal are whitened to generate the input representation for the audio RBM. The hidden representation is generated using the RBM. For that, it is trained using the training data to generate a hidden layer model. This model is used by the RBM to generate a representation of any given data later on. Thus, for both the test and the training data, hidden representations are generated using the model. It is passed through a module for average pooling that results in fixed size output from the RBM section of the pipeline. This fixed size output for each digit is used in linear SVM for training and classification purposes. Linear SVM uses the training data's hidden representations to train the classifier. The test data is then given to the

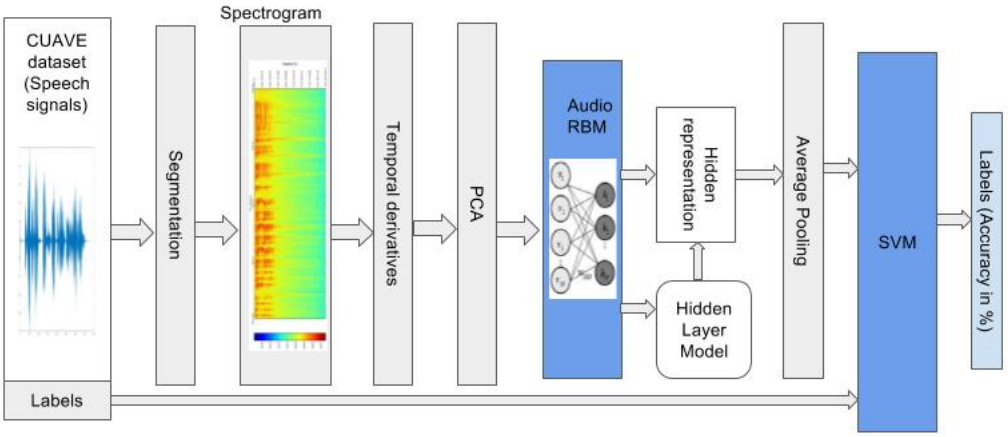

FIGURE 4.4: Block diagram for Audio RBM-SVM framework

linear SVM to get the accuracy of classification. Note that the steps explained here are discussed in detail in the previous chapter.

4.1.3 Video RBM

Video RBM is very similar to the audio RBM except that the input given to the visible nodes are video frames. The dimension of the visible and the hidden nodes are different for video and audio RBMs, as explained in the previous chapter. The figure 4.5 shows the visible and hidden nodes of a video RBM, with inputs [8]. This is also an overcomplete representation where the hidden layer is 4 times bigger than the input layer. The number of nodes for the input layer is 384 and the hidden layer is 1536.

The block diagram of the overall flow of video RBM is shown in figure 4.6 [8]. This block diagram presents a block-wise flow of the system and is an abstract of the flow diagram for video-only system in the previous chapter. It can be seen that the block level flow for video and audio RBMs are similar. For video, raw video frames of the mouth region of interest (ROI) are given as input. After the mouth segmentation is performed, it is given to the PCA module for generating whitened representations. Note that temporal derivatives are obtained on these whitened input data. These temporal derivatives along with the actual frames are given as input to the video RBM. Video RBM uses the training data to generate a hidden layer model, which is used for all future

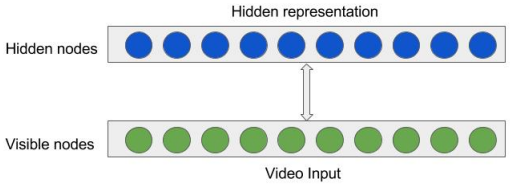

FIGURE 4.5: Video RBM

experiments. The training and the test data are then input into the RBM to generate hidden representations using the model. These hidden representations are used in the linear SVM for training and test purposes, after passing through the average pooling module. Finally, the accuracy of the classification task using linear SVM is computed and recorded for validation.

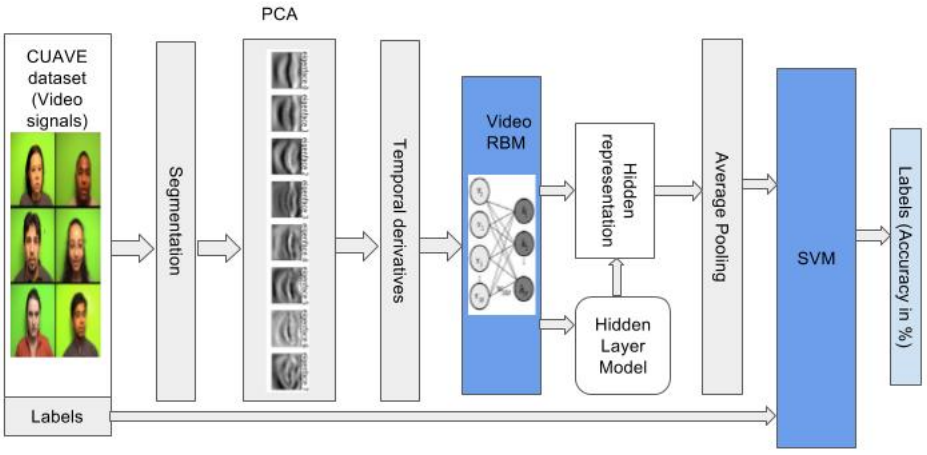

FIGURE 4.6: Block diagram for video RBM-SVM framework

4.1.4 Multimodal Deep Belief Network

The Multimodal Deep Belief Network (Multimodal DBN) is obtained by greedily training an RBM over the pretrained layer of audio and video modalities [8]. The posteriors of

the first level hidden layers are given as input to the multimodal layer. By using this method, it is possible to learn the higher order correlations between the modalities in consideration. In this context, the modalities are video and audio which has a first level hidden layer representing visemes and phonemes respectively. A new layer introduced above the audio RBM and the video RBM layers, helps in learning the correlations between visemes and phonemes that gives the multimodal representation. Figure 4.7 shows the structure of the multimodal DBN in terms of layers and nodes. This has an overcomplete representation with hidden layer 1.5 times the size of the combined first layers. The size of the combined first layer is 3036 and the hidden layer is 4554.

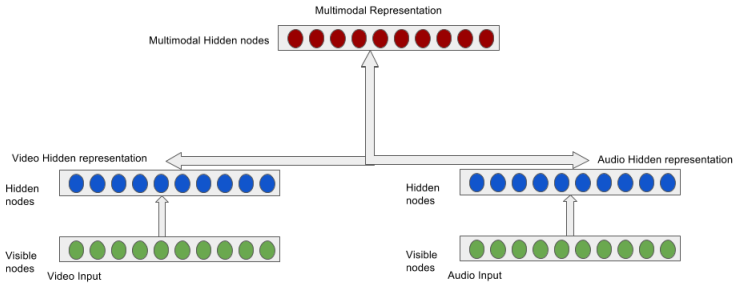

FIGURE 4.7: Multimodal DBN

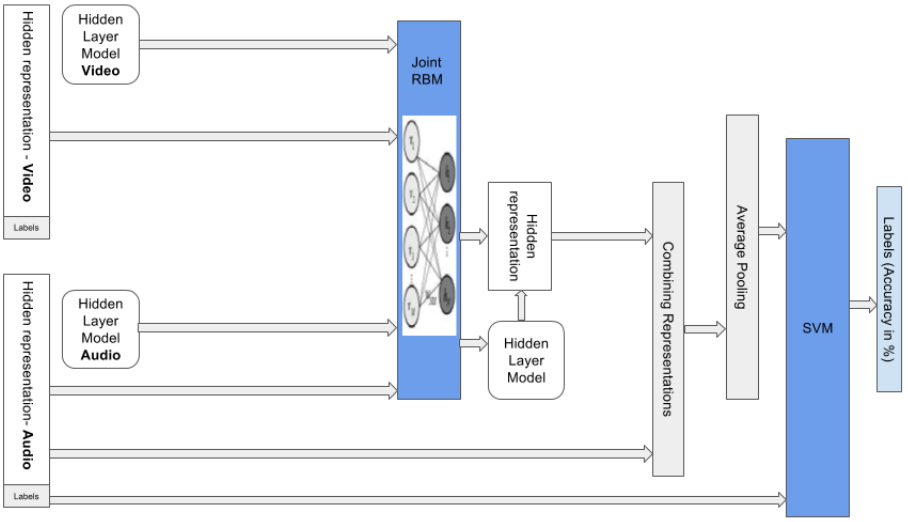

FIGURE 4.8: Block diagram for Multimodal DBN-SVM framework

The module level flow of multimodal DBN framework is shown in figure 4.8. For the multimodal DBN the inputs are given by audio RBM and video RBM [8]. The joint-RBM that uses the these inputs generates a multimodal model of audio and video. This model is used to generate multimodal representations from audio and video modalities.

For classification, as suggested by the reference architecture, audio representations are combined with the corresponding multimodal representations. These combined representations act as the inputs for the classifier. These representations are given to the linear SVM after passing through the average pooling module. The results from the classification of digits by this method is the baseline for the multimodal experiments conducted in this project.

4.1.5 Results from Reference Stack

On performing classification tasks using the above mentioned three frameworks, following results are obtained. The first table shows the results from the reference architecture given by Ngiam et al [8]. The following three tables show the results obtained by conducting experiments in this project, in an attempt to reproduce the reference results.

Feature Representation	Accuracy (%)
Video RBM	65.4 ± 0.6
Audio RBM	95.8
Multimodal DBN + Audio RBM	94.4

TABLE 4.2: Reference result for the architecture as per Ngiam et al [8]

Cross validation set index	Accuracy (%)
1	81.33
2	82.67
3	84.67
4	84.67
5	78.97
6	83.33
Average	82.61

TABLE 4.3: Audio RBM cross validation experiment results

Cross validation set index	Accuracy (%)
1	64.33
2	60.67
3	61.67
4	59
5	50.69
6	60.33
Average	59.45

TABLE 4.4: Video RBM cross validation experiment results

Cross validation set index	Accuracy (%)
1	87.67
2	89
3	88.3
4	87.67
5	86.20
6	91.33
Average	88.36

TABLE 4.5: Multimodal DBN + Audio RBM cross validation experiment results

From the results it can be seen that the Audio and Video RBM performances are close to the reference results [8]. Though the performance is less for the Audio RBM, it seems to maintain a similar relationship with the Video RBM. Configuration of RBM is obtained through grid search, that is explained in the coming section. Grid search is a way to find out better parameters and helped to achieve the performances shown in the above tables. Still, the parameters seem to be sub-optimal, as, the performance obtained in this project are not in exact match with the reference results. However, in the cases of multimodal DBN + audio RBM and Video RBM, the performances seem to be almost the same as the reference results. This implies that the multimodal DBN is able to learn reliable multimodal features from audio and video representations. Hence, this stack can be used for further studies in this project.

Grid search, that helped in achieving these performances, is explained in the next section.

4.2 Grid Search

For the purpose of fine-tuning a neural network, grid search is a reliable method. It helps in achieving configurations that may be closer to the configurations used in the reference architecture, explained in the previous chapter.

4.2.1 Types of Grid Search

The aim of any given learning algorithm A is to obtain a function f, which minimizes some given expected loss [74]. The learning algorithms will have features known as hyper parameters — λ. The actual learning algorithm is the one obtained by selecting the right λ denoted as A_λ and it can be given by the relation ,

$$f = A_\lambda(X^{train}) \tag{4.2}$$

where X^{train} is the training set.

,where τ is the learning rate and $\Delta\theta$ is increment generated in step 2.

Hyperparameter optimization is used for obtaining good values for a set of hyperparameters λ for a learning algorithm with the purpose of optimizing its performance on an independent dataset.

Algorithmic search and manual search are most widely used hyperparameter optimization techniques [74]. Manual search is helpful in choosing the promising regions of a set of configuration variables. It develops the intuition required for selecting values of variables. However, when it comes to the reproducibility of the results and ease of use of the learning algorithms by even the non-experts, manual search is not useful. This makes it less recommended for scientific research realm. On the other hand, algorithmic search gives reproducibility and ease of implementation, which makes it useful in the current context. Hyperparameter optimisations performed manually are very efficient when the number of parameters is limited. However, with more number of trials and with the development of cluster systems and GPU processors, the algorithmic search approaches became more efficient.

As can be seen, the number of trials in random search is much less compared to the grid search. Grid search explores all the possible values of all the hyperparameters.

In algorithmic search also, there are types namely — grid search and random search. [75] Grid search is an exhaustive method of searching all the possible values of the given hyperparameter variables. On the other hand, random search randomizes the search

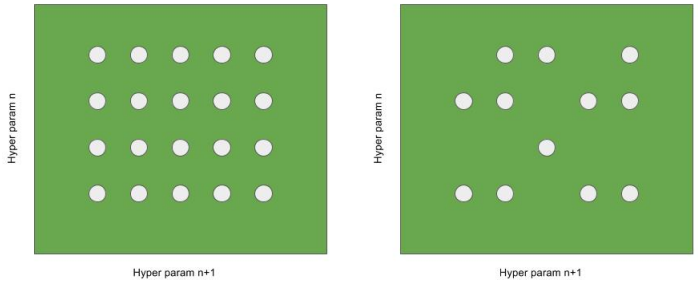

FIGURE 4.9: Possible values of 2 hyperparameters for Grid search and Random search respectively

over the parameters by which the number of trials for finding the optimum values for hyperparameters can be minimized. Random search is shown to have better performance with even one-layer neural networks of several datasets. However, it is found in [75] that when it comes to multi-layer neural networks such as DBNs, random search was unreliable. So, in the current context, it is recommended to use grid search approach with the set of possible values chosen based on the available information.

Configuration variable in neural networks can be learning rate, sparsity, number of hidden and visible variables, etc., which are introduced in the coming section.[74] If there are K such configuration variables, which constitute a set Λ, grid search finds a set of values for each of the variables,

$$(L^1, L^2, ..L^k)$$

via a set of trials of the values. Trials of grid search take into consideration of all possible combinations of possible values of the variables. Thus, the number of trials is given by:

$$S = \Pi_{k=1}^{K} |L(k)| \tag{4.3}$$

This means the curse of dimensionality is applicable to grid search as the number of trials will increase exponentially with the number of variables and their possible values.

4.2.2 Hyperparameters

The RBM used in here is trained using contrastive divergence learning procedure. A considerable amount of practical experience is required for selecting the meta-parameters for the RBM, which are called hyperparameters. The selection of such parameters can

be performed based the available literature such as from Hinton et al [75] and based on the reference models openly available. The RBM and the Deep Neural Networks used in here, namely the Deepnet library, was suggested by an author of [8], as, the code for the reference architecture is not openly available yet. The RBM and multimodal DBN are designed for dealing with multiple modalities, specifically image and text. The RBM used for image modality in Deepnet library is can be expected to be tuned to perform well with the Video, as Video again contains frames which can be considered as raw images. RBM tuned for audio modality is not available off the shelf. Trials with the default values of hyperparameters resulted in a good performance for video as expected. Still, for the audio, the performance is quite low compared to the reference result, in spite of doing several fine tunings in the preprocessing stage. This makes it necessary to tune the RBM for audio by selecting optimal values for the hyperparameters.

The reference architecture [8] has certain issues of reproducibility. The issue of reproducibility is further discussed in the appendix. Since the configurations of the neural networks are not disclosed in [8], the hyperparameters are chosen based on the available information and model in open platforms [75]. A few random trials are performed initially to see if drastically changing the selected hyperparameter has any effect on the performance. Those, which are ineffective in improving or reducing performance are discarded. This helps in reducing the curse of dimensionality and thus the time and resource taken to come up with optimal values for hyperparameters. In the following section the hyperparameters which are considered or discarded and the motivation behind choosing the range of values are discussed.

For the reference architecture, it is given by Ngiam et al [8] that the visible units of the first layer of the deep networks and visible layer of the RBM are Gaussian, whereas, the deeper layer or the hidden layer units are all binary. However, the following parameters are not provided, which makes it necessary to make a right choice of the values for those meta-parameters of the RBM.

Size of mini batch

Weights can be updated after estimating the gradient on each training case, while training the neural networks. Still, an efficient way to train is to divide the training set into smalls sets called mini-batches which can be around 100 [75]. Such a strategy is beneficial on GPU boards as GPUs allows matrix-matrix multiplies. The increment in the mini-batch size to higher values can provide a reliable gradient estimate. Still, it is not helpful in increasing the maximum stable learning rate. This results in smaller weights update per gradient evaluation. Considering the above said aspects, for all the

layers of the network, mini-batch size is set to be 128 which is the default value set for the layers in Deepnet.

Learning rate

The extent at which the weights of the neural networks are updated is controlled by a configuration parameter, namely learning rate. A high learning rate can result in reconstruction error increasing drastically and end up in a weight explosion [75] . On the other hand, reducing the learning rate below a level where it is normal, will result in slower learning in a long run. Though this can reduce the reconstruction error further, it signifies the lower noise level in updating of weights and not a better learning. The default value of the learning rate in Deepnet library, set for the layers, is 0.001. For the grid search, different learning rate ranging from 0.0001 to 0.05 are tried.

One heuristic to see if the learning rate is set properly, is to plot histograms of weights and weight updates. Upon comparison the weight updates should be around 10^{-3} times of the weights. The histograms plotted after 50,000 steps and 100,000 steps are given below, along with the histogram for weight updates per step. It can be seen in figures 4.10, 4.11, 4.12 that the values of the weights histogram are of the order of 10^{-3}, whereas the order of weight updates per step is around 10^{-3} to 10^{-7}. This assures that the learning rate is set to a good value.

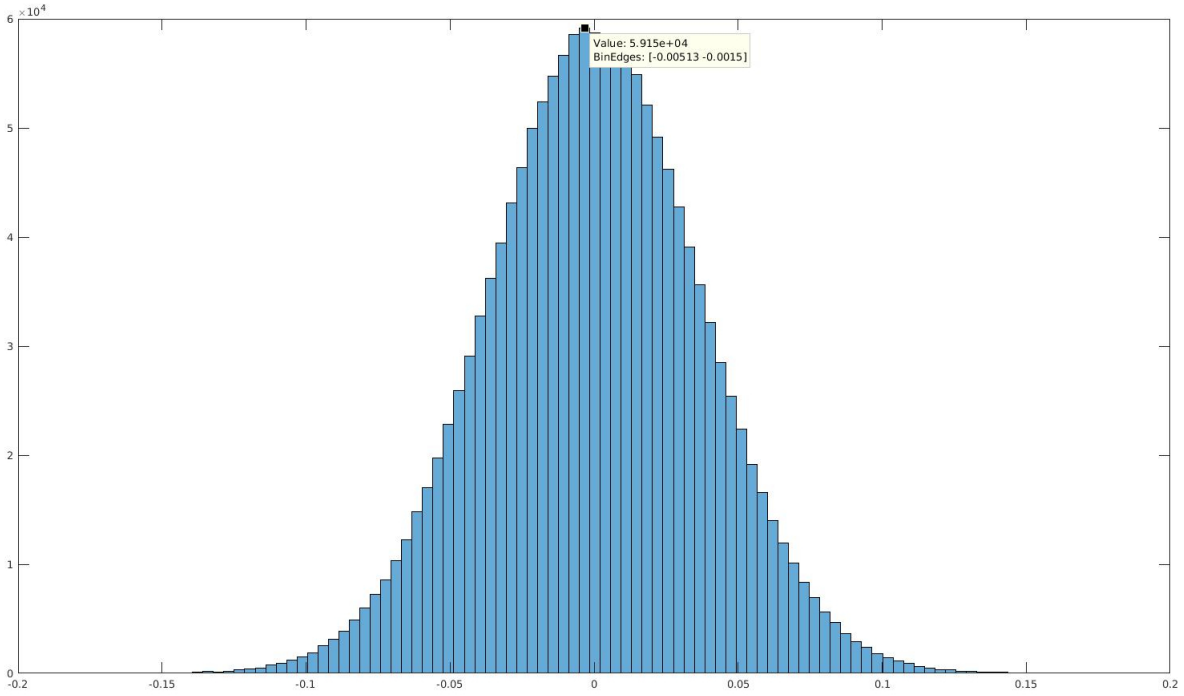

FIGURE 4.10: Histogram after 50,000 steps of the audio RBM training

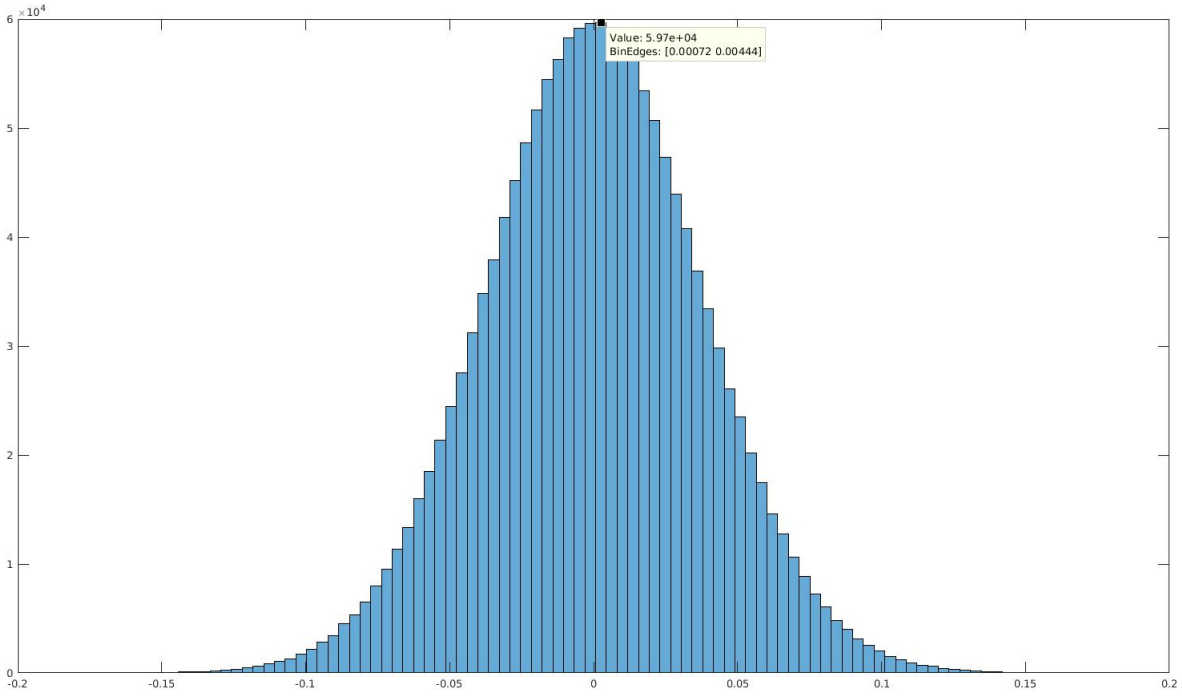

FIGURE 4.11: Histogram after 100,000 steps of the audio RBM training

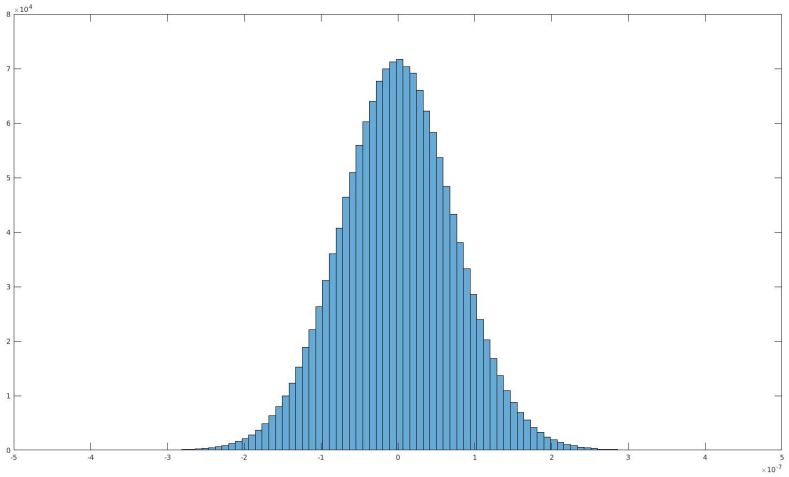

FIGURE 4.12: Histogram of weight updates per steps measure between 50,000 steps and 100,000 steps of the audio RBM training

Momentum

The objective function can be considered to have ravines having steep gradients on the sides and long, narrow and straight floor having a consistent gradient [75]. The parameter — momentum — is a simulation of a heavy ball rolling on the surfaces of a ravine. With the use of momentum, in place of values of the parameters, the velocity of the parameter is incremented by estimated-gradient times learning rate. This velocity is

then used to increment the actual value of the parameter. Velocity decays with the time and a fraction of the previous velocity will be the new velocity for the new mini-batch of training.

If the momentum is denoted by α, momentum method is increasing the learning rate by a factor of $1/(1\text{-}\alpha)$, still the effect of it is split into a series of exponentially decreasing instalments [75] . This enables the system to avoid unstable oscillations while having a larger learning rate. At the beginning, random initial parameters may result in the ball not being on the ravine floor. Thus the momentum should start from a low value such 0.5 and increase to higher values like 0.9. Such kind of a conservative momentum will damp the oscillations across the ravine. These ideal values, start momentum 0.5 and target momentum 0.9, are used for the training in the experiment. So, this parameter is a constant for the grid search experiments.

Gibbs step

Gibbs sampling is the technique used for obtaining posterior samples [76]. In Gibbs sampling the posterior samples are generated by taking samples from the conditional distribution of one or a group of variables, keeping the remaining variables fixed to their current values. In RBMs, the hidden variables are sampled given fixed visible variables and the other way around. This happens in iteration and the number of steps taken depends on the convergence, where the sample values will have the same distribution as that of the true posterior joint distribution. For any generative model of random variables, Gibbs sampling can be summarized in the following steps:

1. Obtain the full joint density and the posterior conditionals for the random variables given in the model. In the context of RBM, it is the set of all visible variables and hidden variables.

2. The posterior joint distribution, based on the posterior conditionals, is then used to simulate samples.

Each learning step in the RBM training will contain one such chain converging. But, considering that waiting for convergence will increase the expenses in terms of resources and time drastically, the number of steps is limited in the code. For the grid search, the values for the Gibbs step are taken to be 15, 20 and 25 based on the models that are openly available.

Weight decay

Weight decay is used for penalizing the large weights in training [75] . This is performed by adding an extra term to the normal gradient which is the derivative of a function that penalizes. The simplest such penalty function is called L2 and is defined as "half of the sum of the squared weight times a coefficient", called weight cost. For the current context, the L2 value is set at fixed 0.001 as small differences in it is not going to result in large differences in performance.

Sparsity

Hidden units that are rarely active are easy to interpret compared to those which are active for around half of the time of the training [75]. Using such rarely active features can also improve the discriminative performance. Such sparse activities in hidden units can be made possible by the use of sparsity target. Sparsity target signifies the amount of probability of desired active time, denoted by p $<<$ 1. The mean activation probability, q, is an exponentially decreasing average of the mean probability of activation of a unit in each step. Sparsity cost, which is the difference of mean activation probability and sparsity target, q-p, is useful in adjusting both bias and the incoming weights of hidden units. Sparsity targets of 0.01, 0.05, 0.1, 0.5 are tried for the current context of grid search. The decay rate or the sparsity damping, λ, of the mean activation probability is set to the default value 0.9. The sparsity cost is varied for a few values in a few trials. However, as it did not have any effect on the performance improvement, only sparsity target is used as a variable for the grid search.

Values of Hyperparameters for Grid Search Following are the set of values used for the grid search, resulting in total 84 different combinations for each grid search.

hyperparameter	Values
Learning rate	0.0001, 0.001, 0.005, 0.01, 0.05
Sparsity	0.01, 0.05, 0.1, 0.5
Gibbs steps	15, 20, 25

TABLE 4.6: Values of Hyperparameters for Grid search

4.2.3 Procedure for Grid Search Experiment

Here to implement the algorithm, sklearn's grid search API is used, which is one of the popular grid search methods. It automatically finds the best of the hyperparameter combination, given the possibilities of the values for each parameter. An estimator class has to be coded with fit function for fitting the training data, predict function to classify the test data and score function that returns the accuracy. Having a cross-validation with each hyperparameter combination costs a lot of time and resources. To avoid this without compromising the reliability of the grid search, the dataset for grid search is shuffled, split into training and test sets externally and is given directly to the fit and predict functions.

A discussion on how the 36 subjects of CUAVE database is divided can make this clearer. The 36 subjects are divided into 6 sets resulting in 6-fold cross validation for general experiments (not grid search). Each of the sets will thus contain 30 training subjects and 6 test subjects. The hyperparameter optimization is to be performed on the 30 training subjects of each of the cross validation set. These 30 training subjects are shuffled and split into training and test sets for grid search. Thus, for the grid search, the 30 subjects mentioned above are shuffled and split into 25 training subjects and 5 test subjects. With each hyperparameter combination, the model is created using these 25 training subjects and it is tested with these 5 test subjects. The classification task is repeated over all the hyperparameter combinations to find the best hyperparameters. The process of finding the best hyperparameters in this way, is performed on all the 6 cross validation sets resulting in 6 different grid searches. The best parameters will be used for the general experiments of the corresponding cross validation set. Figure 4.13 shows the split of subjects.

The code flow for grid search is shown in Figure 4.14. Since the cross-validation is not used in the grid search API, it is initialized to 2. The length of the dataset is 2 as it has only one training and one test set per hyperparameter. The procedure starts with the initialization of the estimator subject[1] with the default values of all the hyperparameters and cross validation settings. In the next step the initialization of hyperparameter 'variables' are performed. Initially the training subjects are used by the fit function to generate the model and the representations. The representations generated, in turn, is used by the predict function to generate classification. From this accuracy is measured and it is returned through the score function which called the predict function. This is repeated for all the hyperparameter combinations. The grid search interface returns the best parameter to the main function, once these trials are over. This is performed on

[1]Estimator object: Estimator interface of sklearn library which holds fit, predict and score functions required for grid search

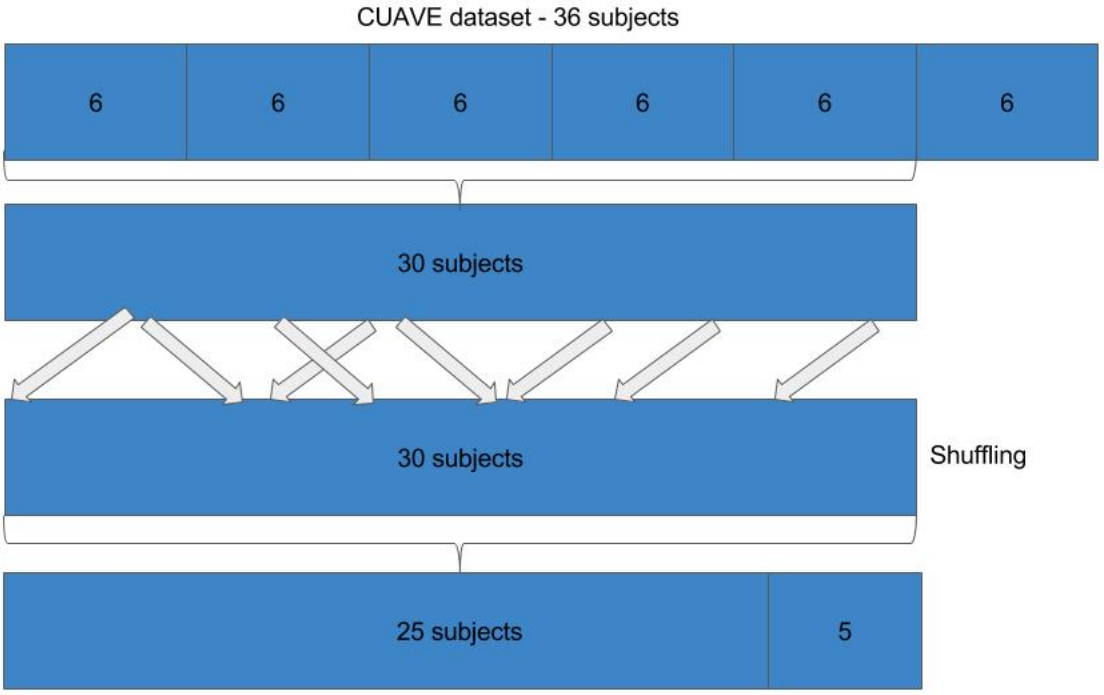

FIGURE 4.13: Training data and test data set splitting

the above mentioned 6 sets of training and test subject combination. For each of the six cross-validation sets, best hyperparameters are computed.

4.2.4 Result of Grid Search Experiment

1. Plot and discuss the results

The grid searches are executed successfully and the result, which is the best parameters for each of the 6 cross validation sets and the accuracy with best parameters in place are given in Table 4.7:

Index	Learning rate	Sparsity target	Gibbs step	Accuracy (%)
1	0.0001	0.5	20	81.6
2	0.0001	0.1	20	84.4
3	0.001	0.1	50	85.42
4	0.0001	0.1	50	82.8
5	0.001	0.5	20	74.4
6	0.0001	0.5	20	90.4

TABLE 4.7: Results of Grid search of 6 cross validation sets

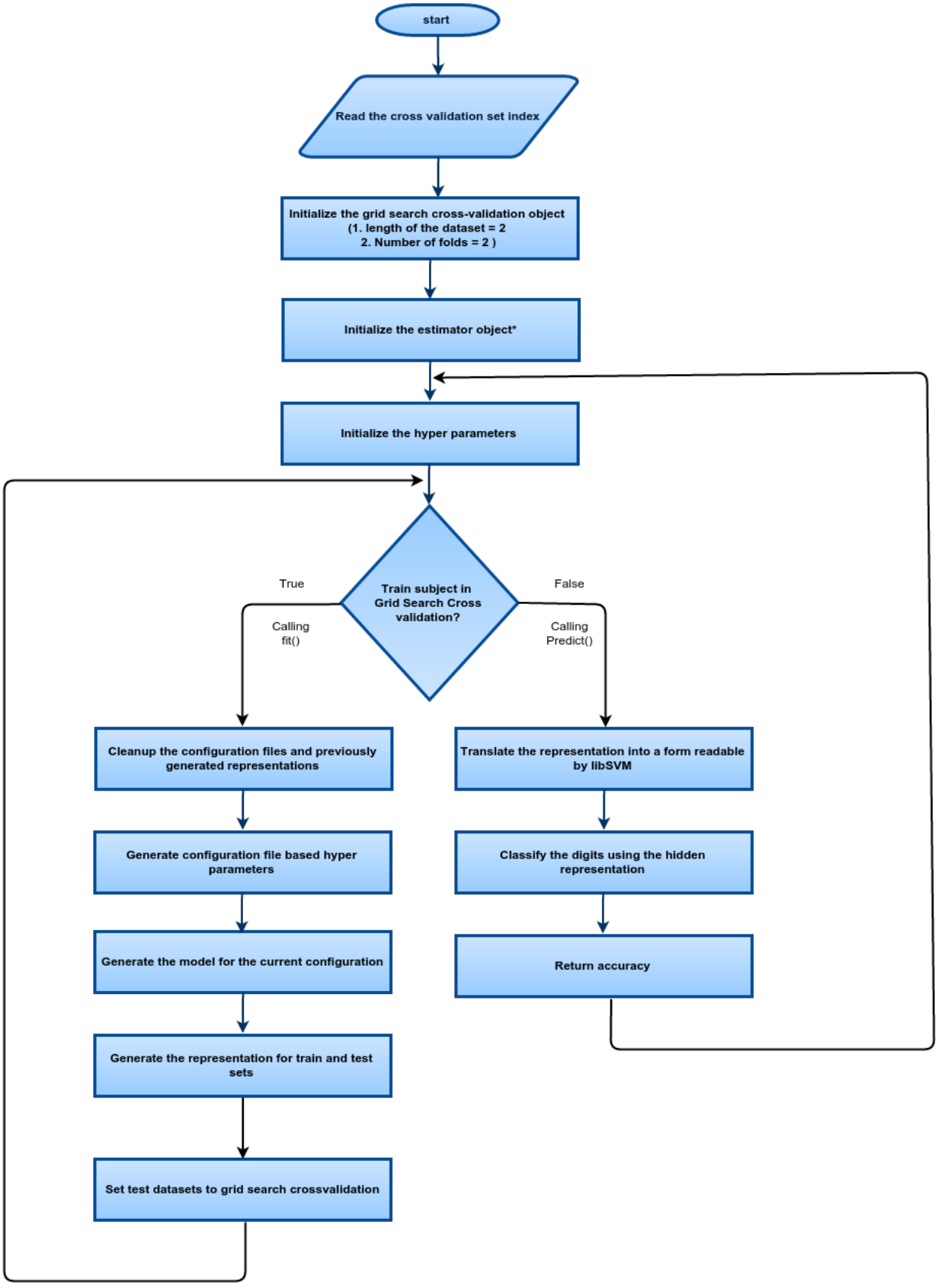

FIGURE 4.14: The procedure followed to perform the grid search (code flow)

The best values obtained for the hyperparameters fall mostly near the middle of the range of values chosen. Thus, it signifies that the range is rightly chosen. However, having more resolution with the values may help improving the performance. This is difficult within the scope of this project as it requires more time and resources. Still, the configuration of neural networks created by this grid search provided promising performances as explained in the previous section.

The following section details the experiments conducted based on the research questions.

4.3 Experiments

4.3.1 Effect of Audio Reverberation

4.3.1.1 Research Question

- As the audio signal gets less reliable, is the video signal able to help compensate?

  Method: Artificially reduce the quality of the audio signal by adding reverberation.

The following sections explain the experiment conducted on the reference stack to answer the research question 1. Thus, the effect of reverberation is studied to understand the limit of the video modality to compensate for the reverberation in the speech modality.

4.3.1.2 Detailed Description

In the coming sections, the experimental setup for studying reverberation as required by the research question is explained.

Speech in Reverberant Room

In a reverberant environment, the speech signal coming out of mouth is combined with reflections from the surroundings [5]. The reflected signals are time delayed and scaled variants of the original speech signal. This results in temporal smearing of speech signal due to the mixing up of direct and indirect energy reflected from the objects and the walls. These reflected waves fill the low energy portions in the temporal envelope of the original signal. This increases the dominance of low-energy frequencies in the speech spectrum and results in masking. Speech which has time-invariant properties, for example steady-state vowels, will be less vulnerable to distortions. However, most speech

contains varying formant patterns, which will suffer from blurring of spectral details. Speech with frequently varying spectra, for example stop consonants such as [t] and [p] will have harmful effects from the reverberation. The speech intelligibility will be affected by factors such as volume of the space, reverberation time, ambient noise level, the source's vocal output level, and the distance between source and the receiver.

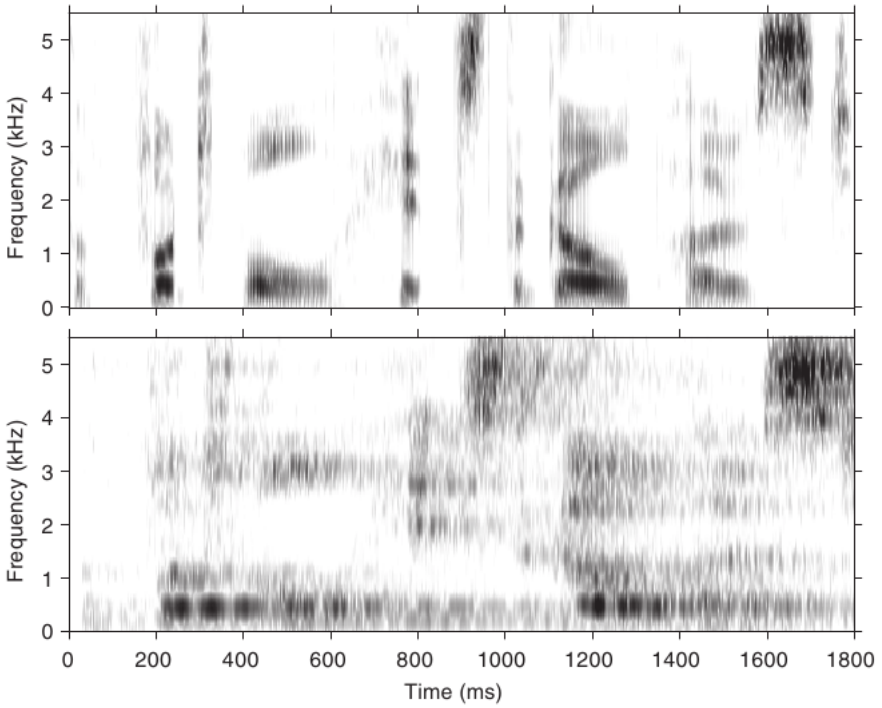

FIGURE 4.15: Shows the spectrogram representation of the sentence 'The football hit the goal post' in quiet environment (top) and reverberant environment (Reverberation Time 1.4 seconds and distance between source and receiver 2 meters) [5]

There are several differences between the speech in quiet and reverberation environment which are summarized below:

1. In stop consonants there are gaps and silent intervals due to vocal tract closure within the sound itself. Such are filled by the reverberation. Such a scenario can be seen with quick variation of noise and silence around the [t] burst in the word 'football', which gets blurred in reverberant environments.

2. Onsets and offsets of syllables gets blurred in a reverberant environment. The effect of reverberation will be more on offsets.

3. Noise bursts in consonants get expanded in duration. A very evident effect of it can be seen with [t] in the word hit.

4. The time interval between stop release and the beginning of voicing is called Voice Onset Time (VOT). Reverberation will blur the relationship between temporal

events such as VOT. This effect will make it difficult to understand the durations of speech segments, for example [U] in the word 'football'.

5. Transitions of formats are flattened, resulting in diphthongs and glides heard like monophthongs. For example [ow] in the word 'goal'

6. The fundamental frequency, denoted by f0 is the property of a speech source and is sensed by auditory system as pitch. Reverberation causes the amplitude modulations associated with f0 to be lessened, causing vertical bands in the spectrogram.

To deal a reverberant environment, the speech recognition systems have to be robust to reverberation effects, explain above. The advancements of such systems are explained in the introductory chapters.

4.3.1.3 Experimental setup

Recall that the purpose of this research question is to find out the effect of noise in audio on the system. By finding the effect of noise, it is possible to understand the issues in deploying such a system in real life scenarios. One of the important scenario of using speech recognition system is in closed rooms, where the possibility of non-linear noises such as reverberation is high. So, it is important to see how the deep neural network is performed in presence of reverberation noises and whether or not it is possible to compensate for such noises by the use of multimodal-deep neural networks.

Here the audio RBM is given with test set containing the reverberation as per the experiment design, which is followed. However, the setup of the video RBM is the same as that of the reference architecture. Room Impulse Generator is used to produce synthetic room impulse responses. Such a generator helps us decide on the reflection coefficient, number of virtual sources, the dimension of the room, the location of the source and speaker etc. Reverberation time is the most important parameter comprising the above attributes and signifying the type of room in terms of the volume and the surface area of the room, the materials used, speed of the sound in the room (changes based on the temperature). It is time taken taken by an impulsive sound, such a hand clap, to decay by 60 dB. Reverberation time is given by Sabine's reverberation formula which was empirically formed in 1890s and is still in use to get an idea of the room acoustics [43],

$$RT_{60} = (24ln(10)V)/(C_{20}S_a)$$

where V is the volume, C_{20} is the speed of the sound at 20 degree celsius, S_a is the surface area and a is the absorption coefficient. Since the room dimensions are usually not changed the materials used in the room.

Reverberation reduced to a very low level by sound absorbing material can make it an 'acoustically dead' room [43]. This makes the room acoustics dull and also requires special design of the room. This technique is mostly used in recording studios. On the other hand, if the room is not carefully designed for reverberation, reverberation can vary a lot based on the materials used for flooring, wall, roof, furnitures, the volume and the surface area of the room. It can be experienced differently based on the location of source and receiver within the room. For the experiment, carned out here, the location of the source and receiver are kept constant. Microphone type is chosen to be omnidirectional for the reason that it is mostly common with condenser microphones. Condenser microphones are mostly used with phones and laptops. Since the directivity of the microphone is omnidirectional, sound from no direction will be attenuated. It otherwise could prevent noise which are not from the source's direction. No filter is used in this virtual setup to remove the reverberation noise in the high frequencies. The room size is set to 18 m2 which is typically the size of a private office room for management in USA or Europe. The distance between source are receiver is set to be 0.3 meter which seem to be a reasonable distance for practical applications.

The plots of the room for the virtual reverberant room are given in Figures 4.16, 4.17.

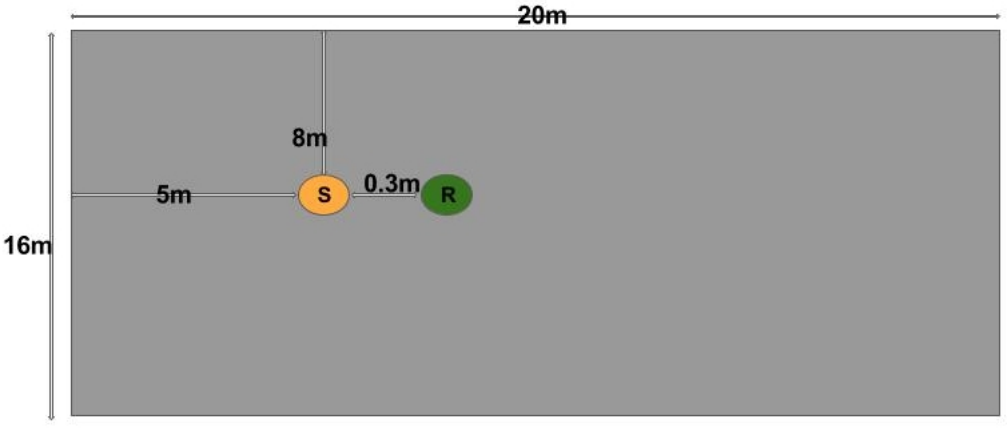

FIGURE 4.16: Shows the top view of the room with source and receiver, and dimensions.

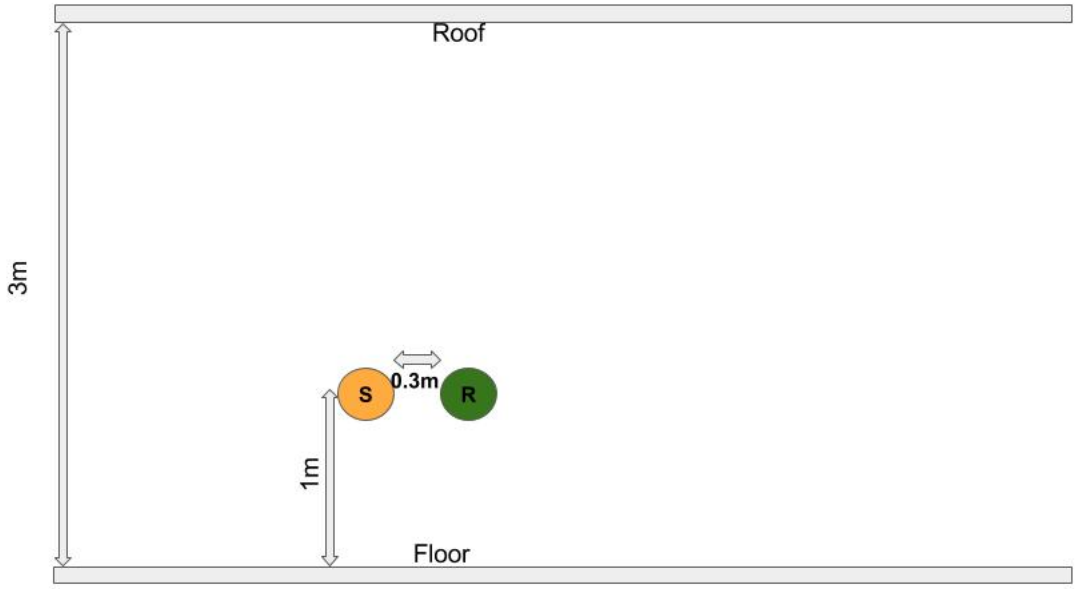

FIGURE 4.17: Shows the side view of the room with source and receiver, and dimensions

The typical values of the reverberation time changes from 0.5 to around 2 seconds. For the private office rooms the values ranges between 0.6 to 0.8 and for the open plan offices it ranges from 0.8 to 1.2. So the values under consideration is from 0.3 to 0.9 in steps of 0.2.

Plot spectrograms of clean and reverberant speeches

The following spectrograms show the spectrum of the digit 'two' with different reverberation settings

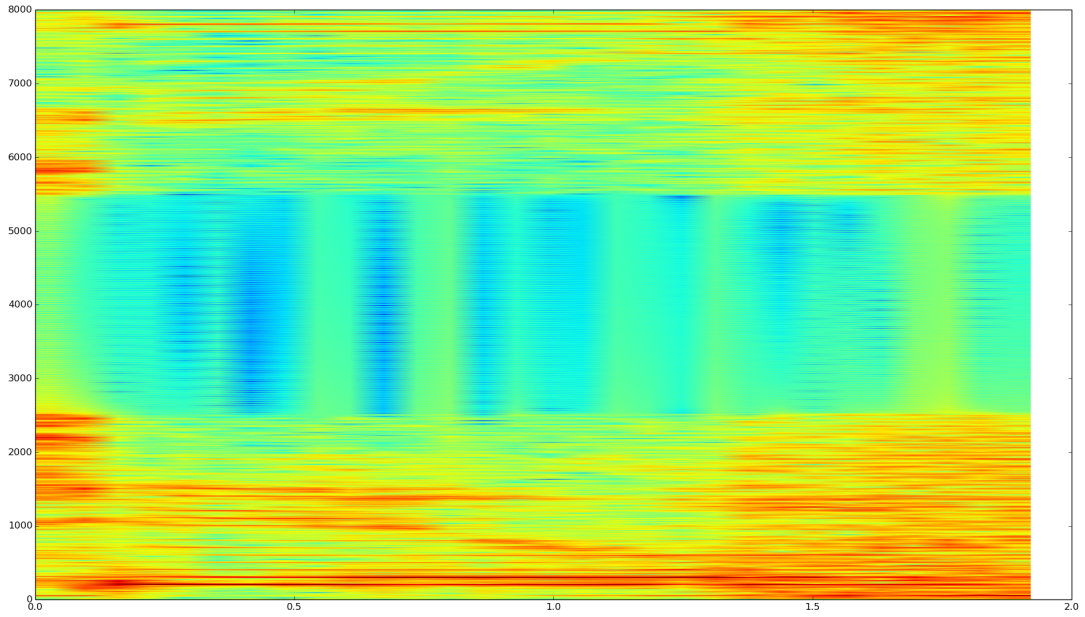

FIGURE 4.18: Shows the spectrogram of 'two' with RT60 = 0.3 seconds

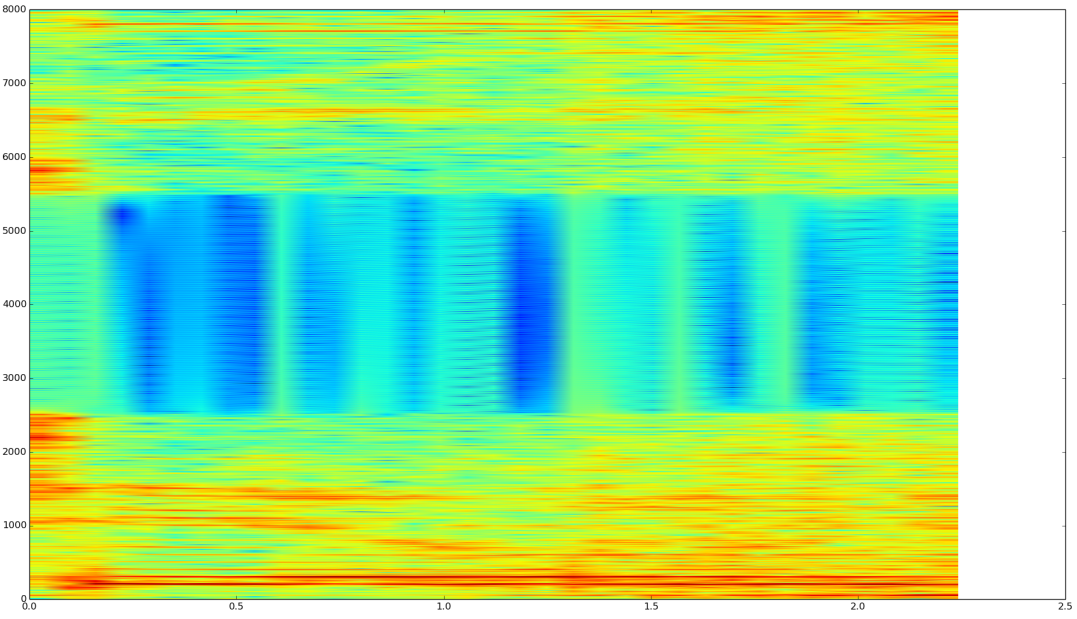

FIGURE 4.19: Shows the spectrogram of 'two' with RT60 = 0.5 seconds

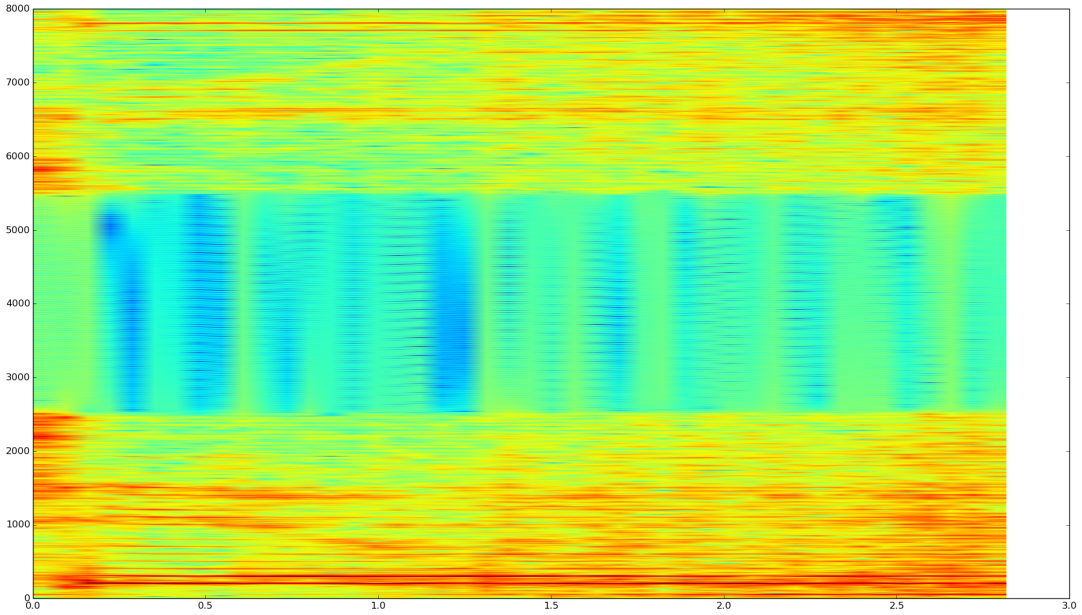

FIGURE 4.20: Shows the spectrogram of 'two' with RT60 = 0.7 seconds

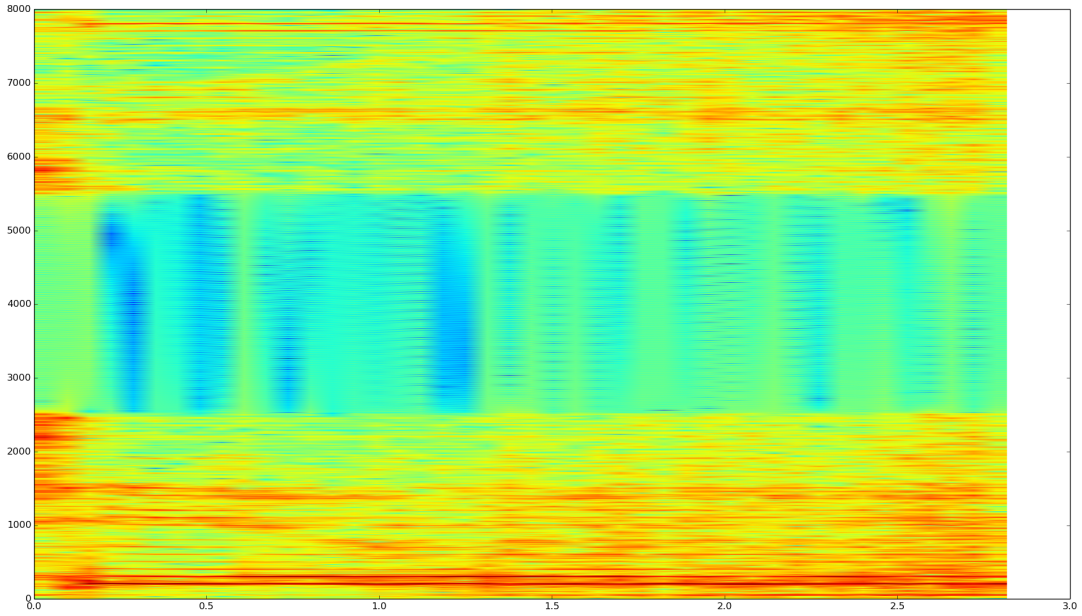

FIGURE 4.21: Shows the spectrogram of 'two' with RT60 = 0.9 seconds

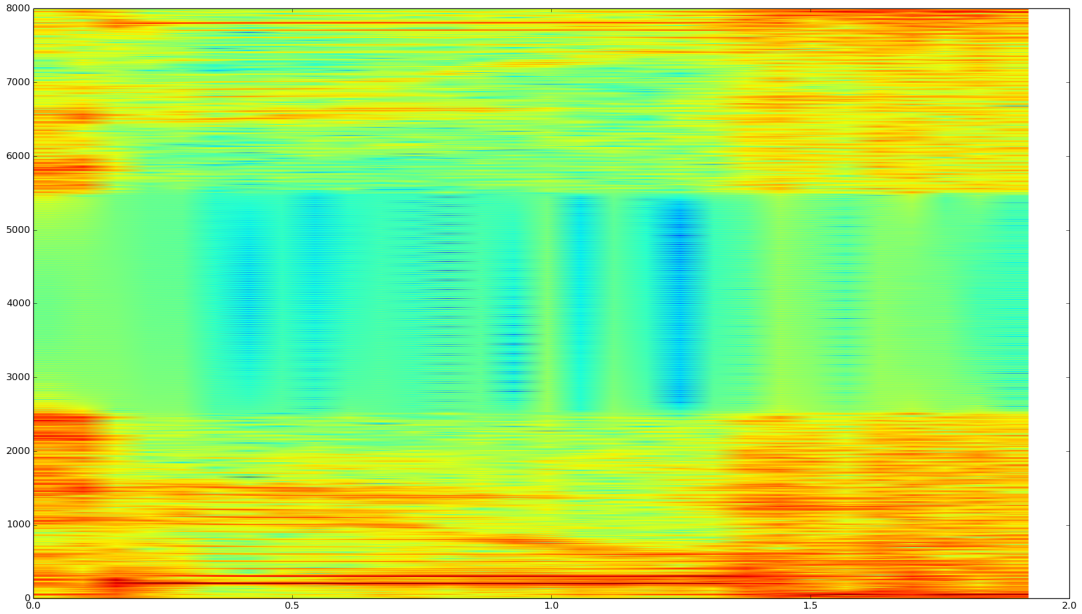

FIGURE 4.22: Shows the spectrogram of 'two' when there is no reverberation

As can be seen in the spectrograms, the burst [t] in two is getting more and more blurred with the increase in RT60, reverberation time.

4.3.1.4 Results and Conclusion

The Table 4.8 shows the result of this experiment. The accuracy obtained for each of the cases that corresponds to different reverberation time is noted. As can be seen in the results, there is a drastic reduction in accuracy with the introduction of reverberation. This considerable drop in accuracy can be mostly atrributed to the room acoustics (simulator settings), for low values of RT60.

The accuracy for audio RBM drops from 66.70% to 51.59% as the reverberation time is increased from 0.3 second to 0.9 second. Thus, the accuracy drops with the increase in reverberation time. This implies that reverberation affects the performance of the system in a destructive manner. The reverberation time is increased from 0.3 second to 0.9 second. Here, the former corresponds to a reverberation that is negligible, and latter corresponds to a level where it becomes annoying. The drop in accuracy is around 15% when reverberation is increased from 0.3 second to 0.9 second. It is seen in the case of Audio RBM + Multimodal DBN representation that the drop in accuracy is similar to that of Audio RBM. Still, the performance of the system for multimodal representation case is much better with a gain of accuracy of around 10%. This indicates that even in the presence of reverberation, the system is able to correlate the audio and video and come up with a reliable multimodal representation.

Reverberation time	Audio RBM accuracy	Multimodal + Audio RBM Accuracy
Baseline	82.61	88.36
0.3	66.70	74.36
0.5	62.89	70.26
0.7	57.38	65.86
0.9	51.59	60.18

TABLE 4.8: Results for different reverberation settings with both Audio and Multimodal DBN + Audio RBM

The results are captured in the Table 4.8. The baseline results are included for reference. The results show that the video signals are able to keep up the performance of the multimodal system in comparison with audio-only system, even when the audio signals gets less reliable due to reverberation. Though, the performance of the whole system is affected by reverberation, the multimodal approach helps in reducing the harmful effect of reverberation on the audio-only system. Interestingly, the neural network is still capable of learning a representation correlating the audio and video modalities, in spite of the considerable differences in the speech spectrographs between clean and noisy cases. Although, the network is not trained with reverberant speech, it is capable of recognizing the clean features from the noisy audio. This indicates that the features generated by the multimodal neural network are robust to reverberation, which is an important nonlinear noise. However, note that the use of multimodality is not helpful in reducing the harmful effect of reverberation. The multimodal system is affected by reverberation to an equal extent that of audio-only system.

The above observations suggest that within the practical reverberation-time limits for an office room, there is no limit for the multimodal approach to improve the performance of the audio-only system. Still, the video signal is not able to compensate for the losses due to a less reliable audio signal.

4.3.2 Information Requirement in Video

4.3.2.1 Research Question

- Does multi-modal speech recognition compensate for different levels of noise differently? (If so, what is the minimal amount of information needed in the video?)

  Method: Use occlusion to reduce the available information in the video, in steps.

The following sections explain the experiment conducted to answer the the research question 2. Thus, the effect of information in the video is studied to find out the

limits. To understand the limits of information required in visual modality, mouth occlusion is used. In steps, the information available to the video RBM is reduced and the performance of the video RBM and the multimodal classification is obtained.

The following sections explain the effect of facial occlusion on visual modality, describe experimental setup and discuss the results.

4.3.2.2 Detailed description

Following sections detail the theory related to the occlusion and mouth occlusion in specific.

Facial Occlusion

In Audio Visual systems facial occlusions can occur due to several different intentional and unintentional reasons as can be seen in Figure 4.23 [6]. In real life scenarios, facial accessories such as spectacles, scarf, facial make-up, hats and microphones are very common reasons for facial occlusions. In some cases, wearable such as veils which are used for religious reasons can even cause occlusions. In certain cases criminals wear sunglasses or scarves deliberately for to avoid getting recognized.

FIGURE 4.23: Real life scenarios of facial occlusion (use of objects that occlude face) [6]

For the face recognition systems which is the first step to use the visual information for any AV system, partial occlusions will change the actual appearance of face considerably [6]. This will drastically affect the performances of classical Audio-Visual systems. The literature so far is mostly focused on finding corruption-tolerant features or classifier to lessen the effect of partial occlusions in the representation of face or facial parts. The issue with such an approach, in general, is that the information from occluded part may

affect the performance. There are numerous algorithms which are made less sensitive to partial occlusions. Another step taken by the researchers recently is to exclude the information from such partially occluded region of face [6]. This is proved to be worth for practical systems to improve the performance. Hence explicit occlusion analysis in a preprocessing stage to have prior knowledge can be a feature of the modern Audio-Visual systems such as Audio Visual Speech Recognition systems.

FIGURE 4.24: Partially occluded face images in real life (hand and face movements) [6]

Occlusions and Speech Recognition

In human communication, non-verbal communication such as gestures and facial expressions play an important role [7]. Hand over face gestures which is one subset of the body language is a common way of expressing emotions and can vary depending on the cultural background. Many of the state of the art system deal with the facial analysis based on geometric shape or appearance of the facial features for extraction, tracking or even classification. When the facial feature such mouth region is occluded, the features that the system is looking for can be lost, corrupted or sometimes even mistakenly detected. This results in an improper analysis of the person's facial information. Sometimes in recognition systems it can result in failing to detect mouth regions. To make such systems robust either an approximation of the lost facial points is performed or the occluded face area can be excluded from the classification process. Though these body language movements convey some information, in speech recognition systems, face occlusions are noise which is to be dealt with by the system.

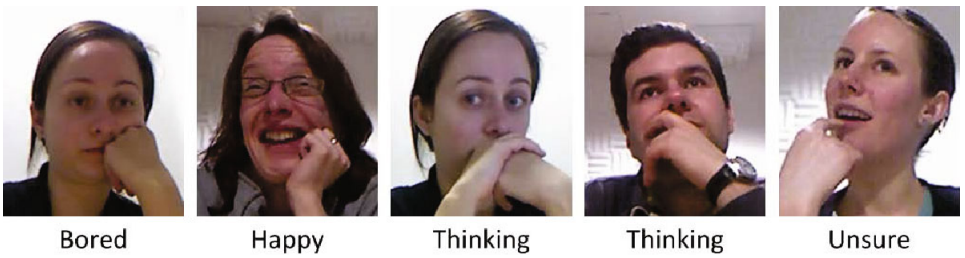

FIGURE 4.25: Examples of different hand over face gestures that occlude face [7]

Hand gestures represent an important body movement. In human to human communications, they signify the emotions that are expressed by speaker and interpreted by

FIGURE 4.26: Examples of different hand over face gestures that occlude face [7]

listener [7]. These are mostly spontaneous gestures humans make, when they want to reflect thoughts and such thoughts are not conveyed via speech. Even the gestures play an important role in shaping the thoughts. As can be seen in the figure, in thinking mental state it is common to occlude the mouth regions with one of several fingers. Though, sometimes it results in complete occlusion where, for a Audio Visual system, the only way is to discard the frame, mostly mouth regions are occluded partially. In many cases fingers cover part of the mouth regions. Still, in some cases of thinking state, there is a tendency to cover one of the lips or one lip and the mouth opening.

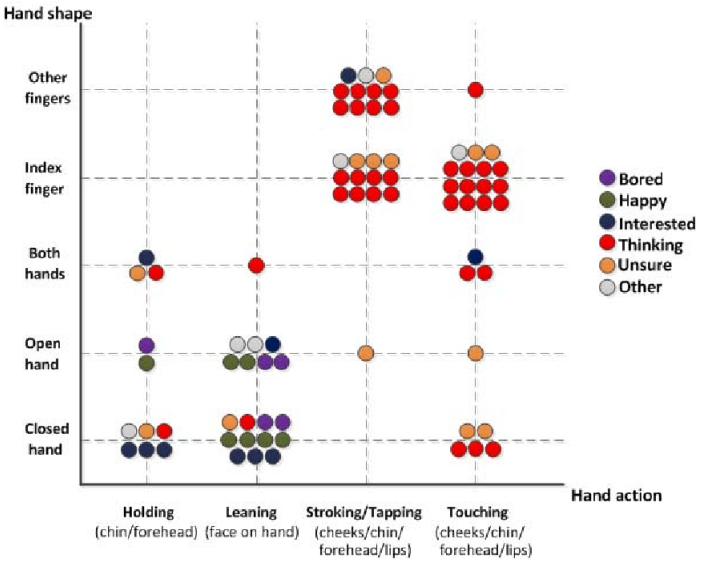

FIGURE 4.27: mental states distribution of hand over face gestures [7]

It can be seen from the mental states distribution that the hand over face gestures, such as leaning on closed hands, open hands or on both hands are rare to be seen in cognitive mental states [7]. Whereas, gestures such as stroking, tapping and touching the mouth regions especially using the index finger are very common with cognitive mental states.

Requirement for minimal information in video

To understand the robustness of the system in practical scenarios, such as occlusion, it is necessary to understand the minimal amount of information required from video. The information from the video is affected mainly due to the facial expression changes, variations in illumination, face pose changes and different kinds of occlusions [77]. Out of all the above issues, with respect to biometric techniques, occlusion is the least studied in spite of being common in real life applications. Although, the subject may have known that he or she is being captured by an AV system, many of the hand to face gestures happen inadvertently. Partial occlusions can greatly affect the original appearance of mouth ROI, possibly causing a drop in the performance of a multimodal system. Knowing the effect of partial occlusions is important to have a robust multimodal speech recognition system. If this multimodal system can deal with different types of noise in similar manner, it may be able to give reasonable performance for occluded digits also. With respect to occlusion, a good performance of the system will reflect that the feature extractor is able to generate corruption tolerant descriptors to be used in the classifier.

4.3.2.3 Experimental Setup

Occlusion in visual modality can be created for the experimental purpose by occluding the mouth region of interest (ROI) for the whole duration of each digit. It is more realistic to occlude for the whole duration of the digits, considering it is typically a few hundreds of milliseconds. This will give a partial occlusion (in space) scenario with which the classifier may be able to deal with, using the available information in non-occluded area. The minimum amount of information needed for the network is to be studied for deployment of such systems in real-life applications.

Typical types of spatial occlusions

There can be partial and complete occlusion of the facial regions. A digit's duration is so small that the occlusion can assumed to be lasting for the whole duration of one digit. Partial occlusion can happen when the subject talks on camera and unknowingly make a hand to face gesture. Such hand to face gestures in common include mouth guard[2], nose touch and eyes touch. In all these cases a portion of the mouth is occluded from the camera, resulting in less visual information.

Types of occlusion relevant to the context

[2]Fingers covering the mouth

For the current scenario, since the subject knows that he or she is being recorded for an AV speech recognition system, complete occlusion with a mouth guard is less probable. [7]. Also, in such a case, the visual information will completely be useless and system should have ways to discard the occluded visual information. With partial occlusion in nose touch or eyes touch, the system should be robust. Such kind of hand to face gestures are probable as it is reflects the mental state of the subject, such as thinking, unsure etc., which does not happen deliberately. The system should be capable to use the available information for classification so as to make it robust enough in real-life applications.

Occlusion in a simulation environment can be recreated by having skin colored stripes over the mouth ROI. In practice it is performed by adding patterns of occluders with brightness similar to that of skin that resembles skin. This stripes simulates the fingers occluding the face. This can occlude a certain portion or the whole mouth ROI. So, for the experimental purpose, 40%, 60% and 80% of the mouth ROI can be occluded with vertical stripes. This gives us 3 scenarios where the information is varied from maximum to a minimum level. Also, the hand movement can be in the vertical plane. In that plane a case where a part of upper lip is only visible and a case where a part of the lower lip only occluded are to be considered. Similar to the vertical stripes occlusion case, here the occlusion can be 40% or 70% of the mouth ROI. This gives us two more scenarios of occlusion. This experiment helps us understand the minimum level of visual information needed for such a multimodal system to be functional. The knowledge of its performance can be handy in developing more robust system for practical applications in future.

The cases of occlusion are demonstrated via figure :

FIGURE 4.28: Shows the 5 cases of occlusion. (a) occluded by 2 fingers constituting 40% of mouth ROI, (b) 3 fingers with 60% occlusion, (c) four fingers with 80% occlusion. d) and (e) shows occlusion because of up and down hand movements. (d) corresponds to 40% occlusion occluding most of lower lip, (e) corresponds to 70% occlusion, occluding a part of upper lip.

Implementation of Occlusion

The mouth ROIs are extracted and is saved in a mat file during the preprocessing stages of reference architecture. When the hand comes between the mouth and the camera, it will be seen as skin colored bright stripes comparable to typical mouth ROI. So, the value of pixels in the occluded area are to be set for higher brightness in the matrices with the same skin tone as of the subject. This needs to be carried out on test subjects of all the 6 sets of cross validation.

The algorithm in the steps involved in this experiment is explained in the appendix.

4.3.2.4 Results

A low accuracy from video RBM indicates that the occlusion caused most of the useful information to be lost, meaning that the availability of visual information for classification purpose can assumed to be less significant. The multimodal DBN + audio RBM system's classification accuracy shows the combined effect of efficacy of the non-occluded information in a multimodal context and the negative effect of occluded region on the performance.

Occlusion scenario	Accuracy (Video RBM)	Accuracy (Multimodal + Audio RBM)
Baseline	59.45	88.36
Case (a) 40% vertically occluded	53.24	83.2
Case (b) 60% vertically occluded	40.56	83.2
Case (c) 80% vertically occluded	40.02	83.05
Case (d) 40% horizontally occluded	39.94	83.44
Case (e) 70% horizontally occluded	15.01	82.66

TABLE 4.9: Results for different occlusion settings with both Audio and Multimodal DBN + Audio RBM

The results are captured in the Table 4.9. The baseline results are included for reference. As can be seen in the results for video RBM, occlusion is negligible when 40% is occluded in case a. This implies that the corners of mouth may have less influence of the classification performance. Thus, it carries less discriminative information. As the occlusion increases, the performance is reduced for 60% of occlusion. This covers

most of the mouth opening. Still, as occlusion increases further, there is no perceivable effect in case c. This can be due the lack of influence of information from the near to corner area of mouth for classification. With case d, where horizontal occlusion covers almost an entire lip, the performance is affected to some extent as in the cases b and c. However, when the occlusion is increased to 70%, covering one lip and almost the complete mouth opening, the performance drops drastically.

Still, in all the above mentioned cases, the performance of the multimodal DBN + Audio RBM features seem to be around 83%. Note that this accuracy is almost the same as that of Audio-only system. The performance of the multimodal system is not affected, as, the system may have discarded the less discriminative information from the video modality. Thus, the video modality is not able to contribute significantly with any occlusion, even when the video RBM gives a reasonable performance. With case e, that is when video RBM performance is the worst, the accuracy obtained from multimodal + audio RBM representation is equal to that of audio-only system. This implies that the video information is completely discarded by the system, as, it did not have any effect on the performance of multimodal + audio representation. So, information available in case e can be considered to be the minimum amount of information at which video modality has no effect on the system. Since this is the only case where the mouth opening is completely covered, information of opening of mouth is the minum information needed by the system. However, note that the sytem accuracy was dropped immediately to almost the audio-only accuracy as and when occlusion is introduced. This suggests that this multimodal speech recognition system is intolerant to any level of occlusion.

4.3.3 Effect of Visual Modality on Individual Phones

4.3.3.1 Research Question

- What is the effect of the visual modality on classification of individual phones?

  Method: Compare the performance of the following two scenarios:

  1. When the audio is silenced for each of the phoneme
  2. When both the modalities are silenced for each of the phoneme

4.3.3.2 Experimental Setup

Phones, Phonemes and Visemes

Phones are the unit sounds in a language which is different from phonemes [78]. For examples $[t_v]$ in tea is a different phone when compared to $[t_r]$ in trip even if they both belong to the same phoneme /t/. If we use $[t_v]$ in place of $[t_r]$, it won't make it a different word for the English language, though, it will sound a bit different. For an English speaker, the difference is so less, that it is almost indistinguishable. But, when it comes to spectrograph, the difference will clearly be seen.

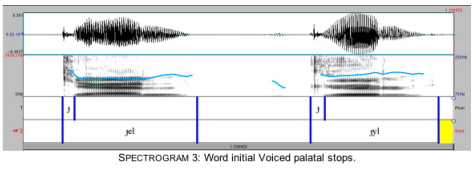

FIGURE 4.29: Shows the same phoneme 'j' which is a palatal followed by two different types of vowels

As can be seen, the spectrogram for /j/ looks different and result in two different phones for the same phoneme. English language has 44 phonemes and several phones under each of the phoneme.

Just like phonemes are the basic building blocks of sounds of a language, visemes have the same role for the visual representations of sounds [79]. Phonemes and visemes are related by a many-to-one mapping. As an example, phonemes /b/ and /p/ are acoustically distinguishable, though, those belong to the same viseme category due to similar sequence of mouth shapes. MPEG-4 multimedia standards suggests a viseme grouping strategy which constitute 14 viseme groups.

On extracting visual representation from spoken words or sentences, it is seen that the same phonetic sound results in slightly different sequence of lip shapes due to the neighbouring sounds and visemes. So, it is seen that the visemes have a polymorphic behaviour under different contexts. The utterance of a phoneme can generate different lip position and movements also based on factors such as person, situation and mood resulting in more variances [80]. In the acoustic modality, consonants such as /p/, /t/, and /k/ are quite similar[81]. Also, in many contexts, the acoustic confusion of the speech modality can be resolved by using the visual modality due to the bimodal nature of speech. For example, /p/ and /k/ are easily distinguishable with the use of visual modality due to closed mouth versus an open mouth.

Detailed explanation of research question

In the current context of digits classification, the spectrogram of the audio is used for input to Audio RBM. So, it is clear that phonemes (or in the spectrogram, phones), are

in focus to understand the effect of visual modality at the unit level of sounds of the language.

Typical use of phonemes is described in detail in the literature, such as English allophonic booklet by Lagos and Bittner, 2006. The typical duration of phonemes are studied by Hiso Kuwabara and is published in [82]. One of the method to split the isolated digits into phonemes can be by making use of frequencies present in the spectrogram. This helps in assessing the presence of phonemes at a given time as described in [83]. However, with the known best possible automatic phoneme segmentation with RNN on a given speech, the classification error is around 17% [84]. This implies a considerable amount of human involvement is needed to have a reliable phoneme segmentation, with the available systems.

The phoneme segmentation of the digits are given in the table.

Digit	ARPABET
Zero	Z-IH-R-OW
One	W-AH-N
Two	T-UW
Three	TH-R-IY
Four	F-OW-R
Five	F-AY-V
Six	S-IH-K-S
Seven	S-EH-V-AX-N
Eight	EY-T
Nine	N-AY-N

TABLE 4.10: The phoneme segmentation of the digits

As can be seen, all digits, except for two and eight, contain three or more phonemes. The phonemes contained in digits are a subset containing almost half of the phonemes in the English language.

4.3.3.3 Implementation

Speech to phonemes segmentation

One convenient way to segment phonemes can be by the use of CMU Sphinx [5]. CMU Sphinx gives a software package that can be used for accessing individual phonemes, their times and durations. This package is known to be one of the most reliable phoneme

segmentation software that is openly available. Using the phone language model, which is available with the CMU Sphinx package, phoneme segmentation was attempted on a few digits. The following description depicts one such segmentation performed using the default phonic language model - en-us-phone.lm.bin - available with the package: SIL-Z-IH-R-AA-R-OW-HH, where SIL is silence. Along with phoneme names, the duration of each of the above phoneme is also assessed by the program. The digit segmented here is zero and the phoneme segmentation is supposed to be : /z I r o/ denoted in ARPABET as Z-IH-R-OW. As can be seen, the segmentation is similar, still not robust enough to be used in the experiment. When the other digits were segmented, the accuracy was not even up to this level.

Thus, it is not completely reliable and manual intervention is required to exactly assess the duration of each phoneme. Based on the prior information available with the digits' phonemes, the corrected duration of each phoneme in digits may be assessed. Though, this reduces the complexity, the time required for phoneme validation will be more due to manual intervention with each segment. It is also possible to take a subset of the digits in the database for answering the research question. Another interesting alternative is to form 3 groups of phonemes of digits. With such a strategy, it is possible to understand the effect of the visual modality on individual phonemes. Whereas, this approach will not require dealing with the complexity phoneme segmentations . Also, it does not demand the time consumption and the expert knowledge in manual classification. As only limited time is available, the approach of dividing digits into three equal segments is used for this experiment.

Study of individual phonemes

The study of the effect of each phonemes in classification is not easy as the ground truth is not available. However, by looking at the performance of the individual digits in classification when a phoneme is absent, it is possible to assess the effect that phoneme.

The digits contain maximum 5 to minimum 2 phonemes. Out of all, five digits contain three phonemes each. Also, considering that the duration of certain phonemes is small, it is convenient to split the test cases into three parts. The three parts can be considered to be three cases that occur when first, second or third set of phonemes become absent. The phonemes can be made absent, by replacing the corresponding phoneme with silence. Since there are no readily available, reliable methods to segment phonemes, the strategy followed in here is to divide the digits into three equal segments. Each segment may contain majorly one or two of the phonemes. Thus, it gives a considerable approximation of phoneme segmenatation and in turn shows the effect of the absent phonemes.

FIGURE 4.30: Phoneme segmentation at the temporal level

As can be understood from the figure, the phoneme sets may contain one or more complete phonemes and a part of another phoneme or phonemes. Here the absence of one such segment of phoneme will show the effect of the missing phonemes in classification task.

Average duration of the phonemes contained in digits 0-9 is given in the Table 4.11. These values are obtained from the available literature on phoneme duration [82, 85–88]. It was seen that there is a variation in duration based on the accent and emphasis [89]. But, the extent of variation seem to a small percentage of the actual duration. As only the relative duration of the phoneme is important in this context, this variation is considered to be irrelevant.

APRABET	IPA	Average duration (milli seconds)
Z	/z/	63
IH	/I/	233
R	/r/	20
OW	/oʊ/	245
AH	/ʌ/	79
N	/n/	55
T	/t/	18
UW	/u/	88
TH	/θ/	15
IY	/i/	85
F	/f/	35
AY	/a_y/	270
V	/v/	11
S	/s/	102
K	/k/	37
EH	/ϵ/	136
AX	/ə/	52
EY	/e_y/	270

TABLE 4.11: Average duration of phonemes based on the available literature

To understand the effect of visual modality on phonemes, two sets of experiments are performed. One in the presence of visual modality and one in the absence of the visual modality. In both the cases the audio modality for the phonemes is absent as described in previous paragraphs. By this approach, the effect of other phonemes and visemes on the classification accuracy can clearly be perceived. In turn, by the comparison of these two experiments, the exact effect of visual modality on phoneme segments can be understood.

To understand the segmentation better the following table 4.12 shows the probable phonemes or part of phonemes present in each segment.

Digit	Segment 1	Segment 2	Segment 3
Zero	34% /z/, 66% /I/	58% /I/, 10% /r/, 32% /oʊ/	100% /oʊ/
One	89% /w/, 11% /ʌ/	100% /ʌ/	12.5% /ʌ/, 87.5% /n/
Two	51% /t/, 49% /u/	100% /u/	100% /u/
Three	37.5% /θ/ 50% /r/, 12.5% /i/	100% /i/	100% /i/
Four	35% /f/, 65% /oʊ/	100% /oʊ/	80% /oʊ/, 20% /r/
Five	33% /f/, 67% /a_y/	100% /a_y/	90% /a_y/, 10% /v/
Six	65% /s/, 35% /I/	100% /I/	12% /I/, 88% /s/
Seven	86% /s/, 14% /ϵ/	100% /ϵ/	9% /v/, 44% /ə/, 47% /n/
Eight	100% /e_y/	100% /e_y/	81% /e_y/, 19% /t/
Nine	43% /n/, 57% /a_y/	100% /a_y/	57% /a_y/, 43% /n/

TABLE 4.12: Contribution of phonemes into segments of digits

4.3.3.4 Results

The following tables show the performance in the absence of each of the three segments of phonemes. This is performed for all the ten digits. For each of thet phoneme sets, the accuracy is noted for both Audio-only system and multimodal system. As metioned in the previous section, the experiment is performed with and without the visual modality (or visemes) corresponding to the absent phonemes. Table 4.13 shows the results in the visual-only case for both Audio-only system and Multimodal system. Table 4.14 shows the comaprison of accuracies of multimodal system, with visual modality and without visual modality. Note that in all the cases the audio modality is absent for the concerned segment.

Digit	Accuracy (first segment) (Audio RBM, Multimodal)		Accuracy (second segment) (Audio RBM, Multimodal)		Accuracy (third segment) (Audio RBM, Multimodal)	
Zero	75.06	92.72	75.94	87.74	86.59	91.60
One	20.08	41.88	77.66	78.26	60.90	77.68
Two	27.85	35.15	39.00	65.79	63.00	79.29
Three	12.22	40.75	29.52	66.34	60.25	80.50
Four	12.34	31.69	45.77	77.64	50.90	76.00
Five	16.42	23.41	77.07	80.02	85.46	87.66
Six	83.16	88.18	70.96	75.94	64.18	75.86
Seven	30.00	49.12	80.44	81.51	79.31	79.41
Eight	16.32	50.12	45.23	69.21	57.49	82.11
Nine	37.33	57.47	52.45	73.16	66.55	78.83

TABLE 4.13: Performance of the audio-only system (Audio RBM) and the multimodal (DBN + Audio RBM) system in the absence of phonemes (with visual modality)

Digit	Accuracy (first segment)		Accuracy (second segment)		Accuracy (third segment)	
	No AV	Visual only	No AV	Visual only	No AV	Visual only
Zero	92.72	92.72	86.07	87.74	88.26	91.60
One	34.66	41.88	73.26	78.26	73.22	77.68
Two	32.34	35.15	62.41	65.79	79.31	79.29
Three	40.19	40.75	59.62	66.34	79.91	80.50
Four	25.58	31.69	77.64	77.64	74.91	76.00
Five	6.11	23.41	84.33	80.02	87.66	87.66
Six	89.29	88.18	76.51	75.94	77.53	75.86
Seven	47.45	49.12	79.85	81.51	77.7	79.41
Eight	49.56	50.12	69.26	69.21	82.66	82.11
Nine	53.58	57.47	69.85	73.16	79.94	78.83

TABLE 4.14: Performance of Multimodal system (DBN+Audio RBM) with visual and no visual modality

The results shows the accuracy of digit classification for each of the three cases for each of the digits. It can be seen that the multimodal system gives a better classification compared to audio-only system, in all the cases. The classification accuracy in the first case ie., when the visual modality is present for the missing segment, can also be due to the other audio segments. To make it clear, another experiment is performed with no visual as explained in the previous section. The result of the experiments with and without visuals are captured in the Table 4.13 and Table 4.14. Note that thet audio of the corresponding segment is absent in both the cases. It can be seen in these results that very few times visual are not helpful, rather deteriorates the performance in a minor way. On the otherhand, in most of the cases it can be seen that, without visal the accuracy goes down. It certain cases the accuracy drops drastically, which suggests that the effect of visual modality is positively significant.

To understand it better it is good to have a look at the confusion matrices for the above cases. The confusion matrices in the second case (where the second segment of

the phonemes is absent) are shown in the Figures 4.31 and 4.32. Here, the values in the diagonal shown in green are the number of correct classifications. The third table, Table 4.33, shows the difference of the confusion matrix for visual-only case with the confusion matrix for no-visual case. So, the values in this matrix indicate the changes in the number of classifications due to the addition of the visual modality (of the missing phoneme segment). A positive value in this difference matrix indicates an improvement due to addition of the visual modality. If the value is negative, it means the addition of the visual modality worked in a destructive manner in that case.

Digits	0	1	2	3	4	5	6	7	8	9
0	157	0	2	1	2	0	3	10	2	2
1	15	140	0	4	10	1	1	1	0	7
2	31	0	118	8	3	0	6	8	5	0
3	41	4	1	119	9	0	4	1	0	0
4	18	12	1	2	139	3	2	2	0	0
5	2	9	0	0	0	154	2	4	1	7
6	19	0	1	4	3	0	136	10	6	0
7	12	7	2	2	0	0	3	146	4	3
8	15	0	6	5	0	0	15	13	124	1
9	22	11	0	4	0	5	1	4	1	131

FIGURE 4.31: Confusion matrix for visual with no audio (second segment)

Digits	0	1	2	3	4	5	6	7	8	9
0	152	0	5	2	1	0	5	9	3	2
1	11	152	0	3	11	2	0	1	0	11
2	32	0	102	12	2	0	8	12	11	0
3	39	1	1	116	16	0	4	2	0	0
4	14	10	0	2	141	3	3	6	0	0
5	0	7	0	0	0	158	3	2	2	7
6	18	0	0	6	4	0	132	12	7	0
7	7	7	4	2	0	0	6	141	6	6
8	17	0	7	7	0	0	14	12	121	1
9	17	14	0	3	0	6	3	6	1	129

FIGURE 4.32: Confusion matrix for no visual and no audio (second segment)

The effect of visemes in the classification tasks can be understood by analysing the difference matrix. Since the second segment majorly contains vowels, the effect of visemes on classification of vowels can be understood. For performing the analysis, each of the digits is considered separately and is analyzed. Such an analysis will be helpful in understanding the effect of the visual modality on a subset of phonemes.

Note that green indicates the changes in the correct classification, in the following tables. Red indicates the changes in important misclassifications of the corresponding digit.

Zero:

Digits	0	1	2	3	4	5	6	7	8	9
0	5	0	-3	-1	1	0	-2	1	-1	0
1	4	0	0	1	-1	-1	1	0	0	-4
2	-1	0	16	-4	1	0	-2	-4	-6	0
3	2	3	0	3	-7	0	0	-1	0	0
4	4	2	1	0	-2	0	-1	-4	0	0
5	2	2	0	0	0	-4	-1	2	-1	0
6	1	0	1	-2	-1	0	4	-2	-1	0
7	5	0	-2	0	0	0	-3	5	-2	-3
8	-2	0	-1	-2	0	0	1	1	3	0
9	5	-3	0	1	0	-1	-2	-2	0	2

FIGURE 4.33: Difference of individual elements in the confusion matrices)

Table 4.15 shows the difference vector for the digit zero. It can be seen that the number of correct classification is increased by 5. The misclassifications to two and six is reduced by 3 and 2 respectively.

5	0	-3	-1	1	0	-2	1	-1	0

TABLE 4.15: Difference vector for zero

Phonemes in the second segment of zero are as follows.

Zero: 58% /I/, 10% /r/, 32% /oʊ/

Zero was misclassified majorly to digits 2 and 6 with phonemes as follows.

Two : /u/ Six: /I/

The effect of visual modality is evident, as there is a drop in accuracy between the cases where only visual is present and no visual/audio is present. Though, the corresponding segments of zero and six contain the same phoneme, zero has other phonemes present in the segment. Also, zero has a transition from unrounded and stretched lip shape of /I/ to rounded lip shape for /oʊ/. Whereas, two has rounded lip shape of /u/ alone and six has unrounded and stretched lip shape of /I/ alone. This information is carried in the visemes and that is evident from the improvement of the classification accuracy in the presence of the visual modality.

<u>One:</u>

4	**0**	0	1	-1	-1	1	0	0	**-4**

Phonemes in the second segment of one are as follows.

One: /ʌ/

Misclassified to:

Zero: 58% /I/, 10% /r/, 32% /oʊ/ Nine: /a_y/

It can be seen that the accuracy of one is not changed by the presence of the visemes. But, digits, which are misclassified as zero in the presence of visemes were misclassified as nine in the absence of visemes. This should be because visemes of /I/ and /ʌ/ are similar, unlike /a_y/, that has a transition from wide open mouth to stretch lips. Thus, the effect of the visual modality on /ʌ/ is neutral.

Two:

-1	0	**16**	**-4**	1	0	-2	**-4**	**-6**	0

The visemes of the second segment of two has a significant effect on the classification. It can be seen that in the absence of visemes, 16 instances of two were misclassified as three, seven and eight.

Phonemes in the second segment of two are as follows.

Two: /u/

Misclassified to:

Three: /i/ Seven: /ϵ/ Eight: /e_y/

The lip takes a very round shape for /u/. Whereas, the vowels of three, seven and eight have stretched and unrounded lip shapes. Thus, the visemes have a large and constructive effect on the classification in the case of /u/.

Three:

2	**3**	0	**3**	**-7**	0	0	-1	0	0

Phonemes in the second segment of three are as follows.

Three: /i/

Misclassified to:

One: /ʌ/ Four: /oʊ/

As can be seen, there is a minor positive effect of visemes on the classification of three. Visemes help in discriminating /i/ and /oʊ/. Still, visemes are not useful in segregation /i/ and /ʌ/ as both have unrounded lips though the stretching of the lips are slightly different.

Four:

4	**2**	1	0	**-2**	0	-1	**-4**	0	0

Phonemes in the second segment of four are as follows.

Four: /oʊ/

Misclassified to:

Zero: 58% /I/, 10% /r/, 32% /oʊ/ One: /ʌ/ Seven: /ɛ/

Use of visemes of second segment has a negative effect on the classification of four. Though, the misclassification of four as seven is reduced, it is further misclassified as zero and one. This is not surprising as /ɛ/ has unrounded lips, whereas, /oʊ/ has rounded lip shape. Also, /oʊ/ is present in both four and zero. One is preferred over seven, as, /ʌ/ is closer to rounded than /ɛ/.

Five:

2	**2**	0	0	0	**-4**	-1	**2**	-1	0

Phonemes in the second segment of five are as follows.

Five: /a_y/

Misclassified to:

Zero: 58% /I/, 10% /r/, 32% /oʊ/ One: /ʌ/ Seven: /ɛ/

Five has similar results that of four. /a_y/ has a transition from one unrounded lip shape to another unrounded lip shape. /ʌ/ and /I/ are acoustically different from /a_y/. Still, the difference is not enough useful in discrimination of these phonemes. From the results, it is clear that the addition of visemes further creates a confusion in the classifier resulting in more misclassification.

Six:

1	0	1	**-2**	-1	0	**4**	**-2**	-1	0

Phonemes in the second segment of six are as follows.

Six: /I/

Misclassified to:

Three: /i/ Seven: /ɛ/

In the above three digits, it can be seen that the audio is very similar to each other. Still, mouth shapes look different for those phonemes. /i/ has stretched lips almost like a smiling face, for example /i/ in 'cheese'. This is the reason for saying cheese when posing for pictures. /i/ has more stretched lips than /I/ and it is clearly distinguished in visuals. Lip's shape for /ɛ/ is close to /I/, still, not as stretched as /I/. Thus, the visuals of these phonemes add on to the discriminating capability of the features.

Seven:

5	0	**-2**	0	0 0	**-3**	**5**	**-2**	**-3**

Phonemes in the second segment of seven are as follows.

Seven: /ɛ/

Misclassified to:

Zero: 58% /I/, 10% /r/, 32% /oʊ/

Two: /u/ Six: /I/ Eight: /e_y/ Nine: /a_y/

For seven, it is seen that the addition of visuals of the second segment improved the performance. Still, half of the digits, previously misclassified as two, six, eight and nine are misclassified as zero. The remaining half is classified as seven. Zero, in the second segment, has a transition from unrounded and stretched lip's shape to a rounded shape. It seems like this transition makes the lip's shape in this segment looks similar to that of /ϵ/, which is unrounded and a little stretched. Still, the visuals are useful in distinguishing lip's shape of /ϵ/ from /u/ which is rounded, /I/ and /e_y/ which are more stretched and /a_y/ which is more wide.

Eight:

-2	0	-1	**-2**	0	0	1	1	**3**	0

Phonemes in the second segment of eight are as follows.

Eight: /e_y/

Misclassified to:

Zero: 58% /I/, 10% /r/, 32% /oʊ/ Three: /i/

/e_y/ of eight achieved a better performance with the addition of visuals. /e_y/, that has an unrounded and a little stretched lip shape is distinguished from zero, that has a transition from unrounded /I/ to rounded /oʊ/ and /i/, that has an unrounded and more stretched lip shape.

Nine:

5	**-3**	0	1	0	-1	**-2**	**-2**	0	**2**

Phonemes in the second segment of nine are as follows.

Nine: /a_y/

Misclassified to:

Zero: 58% /I/, 10% /r/, 32% /oʊ/ One: /ʌ/ Six: /I/ Seven: /ϵ/

Adding visual for the segment gives a similar result for both seven and nine. Similar to seven, adding visuals improves the performance. Still, the improvement is less as the digits are further misclassified to zero. /ay/ has a transition from an unrounded lip shape to a rounded lip shape. This is similar to the second segment of zero, which has a similar transition from /I/ to /oʊ/. Still, the visuals help in distinguishing /I/, /Λ/ and /ϵ/ - those having unrounded lip shapes - in comparison with /ay/ that has rounded lip shape.

Since zero has multiple phonemes in the second segment, the misclassification to zero can be overlooked. Whereas, the addition of visuals always helps in disambiguating other vowels from /a_y/ and that is promising.

From the analysis of the second segments of digits (which are assumed to correspond to vowels) using the difference matrix, it is clear that the visemes help in improving the discriminating capability of the multimodal features. From the results, it seems that the system is able to make use of the information carried by the lip shape in phonemes and the transition of lip shapes within each phoneme. Further, it is seen that the phonemes with similar lip shapes are even distinguished by the system. Though, in one case, the addition of the visual modality reduced the performance in a minor way, majorly, it helps in improving the classification. Thus, the effect of the visual modality on the classification of individual phonemes is mostly constructive. This effect depends on the dissimilarity of the visemes in comparison with the visemes of other sounds under consideration. The dissimilarity can be in terms of roundness, stretch, wideness and/or transition.

Chapter 5

Conclusion and Future work

5.1 Conclusion

This thesis is an analysis of an Automatic Speech Recognition system which makes use of the multimodal nature of speech. Similar to a human speech recognition system, the system reimplemented here has an ability to associate visual and audio modalities in understanding the speech. It is proved in literature that the visual modality helps in disambiguating speech in many cases such as in the case of McGurk effect.

In the early period of 21st century, the speech recognition system were mostly based on HMMs and GMMs. These models were based on the known models of human speech recognition system. However, secrets of the articulatory human speech dynamics and the deep control mechanisms are not completely known to us. For efficient computation, even the known model was minimized and a requirement of statistically independent features was used. The use of visual modality was missing in such systems. These limitations were addressed by the modern speech recognition systems. In the modern speech recognition systems, several neural networks were made use of. This includes several deep neural networks such as Convolutional Deep Neural Networks (CDNNs), Deep Belief Networks (DBNs), Denoising Auto Encoders and hybrid models combining CDNNs and HMMs. Nigam et al used multimodal nature of speech for the first time in ASRs that was identical human speech perception in terms of many attributes including McGurk effect. It was further studied in the literature and many systems were proposed based on the same concept.

This thesis implements the system proposed by Ngiam et al to have a multimodal speech recognition system, using DNNs. The goal of the thesis was to analyze the system in different real life scenarios and find the limits on modeling compensation of such a

system. Analysis includes scenarios such as performance of the system in reverberant rooms and occluded visuals and also, a study of the effect of multimodality on individual phonemes.

The reference system was reimplemented based on the information available in the available reference literature. Assumptions were made based on the available knowledge, as, some of the information was missing in the reference literature for the system suggested by Ngiam et al. The reference system was implemented using Deepnet library as suggested by one of the authors of the reference system. The preprocessing stages were implemented mostly in python and C++ using publicly available libraries such as sklearn and object detector of Matthias Dantone et al [53]. LibSVM was used as the linear SVM classifier for the reference implementation. For the neural networks, the hyper parameters were not completely disclosed in the literature. Thus, a sequential grid search was used to fine tune the hyperparameters of the neural networks. This made the accuracy of the system improve to an acceptable level.

Grid search was performed to find the best suitable values for the hyper parameters of the neural networks. This is an exhaustive search of all the possible values of the given hyperparameter variables. In this thesis project, based on the available literature, a set of values were chosen to be used in grid search, for each of the hyperparameters. Grid search was performed on this set of values and the result obtained was promising. This helped in achieving a reasonably good performance for all the three systems — audio-only, video-only and multimodal systems — and replicating the reference architecture.

The reference results for this experiment was obtained by replicating the architecture of Ngiam et al. The experiment has not shown a complete correlation with the original results. Still, the results obtained were promising as the multimodal system yielded a good accuracy. It gave an accuracy of 88.36%, in spite of all the possible dissimilarities in the experimental setup. The audio-only system yielded an accuracy of 82.61% which is considerably less than that obtained in the original experiment of Ngiam et al. However, since the study conducted here is focused on the multimodal nature of speech, the system is considered to be reliable for further experiments.

The first experiment was to understand the limits of the system in the presence of reverberation. To setup a reverberant room, a Room Impulse Response (RIR) generator was used. This way, it is possible to set the room for a specific reverberation time. The reverberation time (RT60) signifies the extent of reverberation in a room, as, RT60 is the duration of time required for sound to decay 60 decibels from its initial level. For this experiment, the value of RT60 varied from 0.3 seconds to 0.9 seconds. It is seen that the system's performance deteriorates with the increase in RT60, both for audio only system and multimodal system. Still, the use of multimodality constantly gives

an improvement over the performance of the audio-only system. The improvement that multimodal system offers is around 10%. This indicates that multimodality helps the system in reverberant conditions to achieve a better performance. However, the use of multimodality is not able to compensate for the losses due to reverberation.

In the second experiment, the effect of occlusion on multimodal systems was studied. To understand the effect, occlusions with hand movement around mouth is considered. Cases considered for the experiment simulated presence of multiple number of fingers blocking the view of mouth. The extent of occlusion was varied between 40% to 80%. Occlusion in vertical and horizontal dimensions were considered. It is observed that the performance of video-only system reduced as the occlusion increased. Whereas, the multimodal system was able to give a reasonable performance in spite of the presence of occlusion. As such, the system gave a performance that is feebly better than of audio-only system. Such a performance implies that the multimodal system is stabilized against missing information in video due to occlusion.

To understand the effect of visual modality on phonemes, a third experiment was conducted. This experiment segmented the digits into three equal slices. Accuracy was measured in three cases corresponding to each of the segments. Also, to understand the effect of the visual modality, the experiment was conducted with and without visual modality in all the cases. The second segment of digits that corresponds to vowels was considered for detailed analysis. From the analysis, it is observed that the system makes use of the information carried by the lip shape in phonemes and the transition of lip shapes within each phoneme. It is also seen that even the similar lip shapes of closely sounded vowels were distinguished by the system in many cases. In general, presence of visual modality of the phonemes (without audio) proved to be useful for classification. The experiment proves that effect of visual modality on classification of individual phonemes is mostly constructive. The dissimilarity of the visemes in comparison with the visemes of other sounds plays an important role here.

In general, it can be seen that, by using multimodal system in place of audio-only/video-only systems, an improvement in performance is achieved, even in the presence of noise in individual modalities. It is observed that the visual modality is used by the system in a similar manner to that of a human being. Also, the multimodal system uses certain visual cues which are overlooked by the human perception system. An example for such a scenario is, the multimodal system making use of the visual modality in distinguishing similar visemes, which a trained human can only understand. However, the multimodal system studied in here, is far from achieving human-like performance in a noisy environment. On observing the results of the experiments on phonemes and also the drop in performance with the introduction of occlusion/reverberation, the above-mentioned

statement is justified. These observations imply that there need to more research in the area of robustness of the multimodal systems in real-life scenarios.

5.2 Future work

One of the achievement of the studies conducted in this thesis is that it gives new insights for future research initiatives. Since, the studies conducted in thesis cover a wide area under speech recognition, the recommendations cover several disciplines under speech recognition. The recommendations are based on the research questions, the understanding of the systems that we came across and the knowledge acquired by study of the related literature. A few of the future recommendations are based on the inference from the experiments.

In reverberation experiments performed in this thesis project, we make use of a Room Impulse Response generator to have a specific reverberation configuration enabled. Another way to do this is to use the impulse responses recorded from different real-life rooms. This approach still has issues as it is again a simulation of the reverberant environment. Having an AV database, where the recordings are made in real-life reverberant environment can provide better analysis of the performance of the multimodal systems.

Occlusion is just one type of disturbance in visual modality. There are different types of real-life scenarios to be studied such as variation in lighting conditions, distortion in shape of the region of interest due to the angle of camera, missing frames due to the issues in the transmission channel and so on. Also, the simulation of occlusion was performed using skin colored strips on mouth ROI. This experiment can be conducted using real-life data, if recording of several subjects can be performed with different types and extent of occlusions, which are naturally occurring.

With phonemes, the experiment was performed using approximate segmentation of the AV data. Further research can be conducted using precise segmentation performed by experts or using expert knowledge. Also, the effect of coarticulation, effect of accents and non-native pronunciations can be studied using a similar experimental setup. Such a study can be used to obtain insights of the behavior of multimodal systems in real-life scenarios.

Apart from several analysis, which were suggested above, there are a few changes in the system that can be tried out to improve the performance of a multimodal system. The current system uses an ROI extractor which does not compensate for any sort of shape distortions or orientations. Though, the subject is requested to be in a natural stand still, there will be enough movements to cause a distortion in the shape of the ROI.

This can result in performance degradation which is due to an inefficient preprocessing. Usage of Active Appearance Model for mouth recognition and extraction may be studied to see the effect of such distortions in the multimodal system. This can give a better understanding of multimodal system in dealing with shape distortions in visual modality.

The system used in here has an average pooling module, which is essential due to the presence of linear SVM requiring fixed size inputs. Though, pooling helps in making the system translation invariant, it removes the information carried in features generated by the neural networks. It will be interesting to see the effect of having a classifier, that can handle variable size input. This can help in understanding the effect of variance in the modalities on the multimodal features generated by the system. Also, it can give a better understanding of the effect of losing information via techniques such as pooling.

Finally, the multimodal system can be further extended by addition of modalities such as the depth information. Since this is one of the inputs to the human perception system, it will be interesting to see the effect of depth information on multimodal features. Addition of regions of face apart from the mouth, such as jaws may add more discriminative information for the classification task. Since this is another information available through visual modality, the effect of depth information will be interesting to study.

Appendix A

Appendix: Implementation Details

A.1 Preprocessing

Digit segmentation

A python code was created for segmentation of digits automatically. The interface goes as follows:

digitSegmentation.py -i <inputfile> -o <out file> -d <duration >

The total duration of frontal face is given as the duration. The input file can be a wav or mpg file and the output file is a label file containing the durations of digits. This label files are generated in compliance with label format of Audacity. Having this feature enables manual editing or verification of labels when necessary. Algorithm at a high level is given in 1

Algorithm 1 Segmentation of Digits

procedure DIGITSEGMENTATION
    $file \leftarrow$ name of input wav/mpg file
    $duration \leftarrow$ duration of frontal face
    $prefix \leftarrow$ prefix of output file
    $audio \leftarrow file$
    $frontalAudio \leftarrow audio(0 : duration)$
    $minimumSilenceDuration \leftarrow$ 200 ms
    $silenceThreshold \leftarrow$ -45dB
    $nonSilenceRegions \leftarrow detectNonsilent(frontalAudio, minimumSilenceDuration, silenceThreshold)$
    $i \leftarrow 1$
loop:
    if $numberNonSilentRegions \geq i$ **then**
        $i \leftarrow i + 1$.
        digitDuration(i) $\leftarrow formatAndSplit(nonSilenceRegions(i))$
        goto *loop*.

Spectrogram and mat file creation

A python function is created for the spectrogram generation and the interface is given as below:

python ./spectrogram.py -i <input file> –start=<start time> –stop=<stop time> -o <output file>

Here, the spectrogram takes an input (mpg/wav file) reads the audio for the duration of start time to stop time, which is in milliseconds. It outputs a mat file containing spectrogram for the whole duration. For each of the frames, an array containing 161 x 3 elements is created. The output mat files is a linearization of the array into 483 xN elements, for each audio frame duration. Here, N specifies the number of audio frames of duration 20 milliseconds. The high level algorithm is given in 2

Algorithm 2 Spectrogram Generation

1: **procedure** SPECTROGRAM
2: $file \leftarrow$ name of input wav/mpg file
3: $startTime \leftarrow$ start time of the speech for spectrogram
4: $endTime \leftarrow$ end time of the speech for spectrogram
5: $fs \leftarrow$ 16kHz
6: $numberOfBins \leftarrow 320$
7: $overlap \leftarrow 160$
8: $window \leftarrow$ Hanning
9: $audio \leftarrow$ data from file
10: $formattedAudio \leftarrow formatToArray(audio)$
11: $i \leftarrow startTime$
12: *loop*:
13: **if** $endTime \geq i$ **then**
14: $scaledAudio \leftarrow scale(formattedAudio)$
15: $spectrograph(i) \leftarrow spctrogram(scaledAudio, window, numberOfBins, fs)$
16: $reshapedSpectrograph \leftarrow reshape(spectrograph)$
17: **goto** *loop*.
18: $fullSpctrograph \leftarrow matrixConversion(reshapedSpectrograph)$

Mouth Recognition

For the mouth recognition, the library[1] openly available from Matthias Dantone et al is used as mentioned before. This library is implemented completely in C++ and comes with a pretrained facial feature and head pose recognizer. Though this library contains the code for facial feature recognition, the code had to be edited to make it suitable for the project.

The following are the overview of the modification made in this library:

1. The library has annotation files which gives a predefined user box for the facial feature search and the name of image on which the search has to be done. This was modified to read video and images given through the command line.
2. The input images were converted to a compatible format through array operations.

[1]To download the library: http://www.dantone.me/projects-2/facial-feature-detection

3. The code for usage of predefined bounding box is modified to have no bounding boxes for face detection. This is fine for CUAVE visuals as there is only one face in a frame and the noise due to background is negligible.

4. One key feature required by the code was to integrate it with python. For that purpose, initially, python/C APIs and CPython were tried. But, it was seen that as the code is not robust enough, on a long run the program crashes. Thus, it was decided to use it as a separate executable to avoid memory and synchronization issues.

5. Another solution tried was, to use video directly with the library as input, which increased the execution time. This is because seeking through the video files when long, takes a lot of time.

6. To avoid the above issues, the program is changed to output only the mouth coordinates to console. This was done by channelling the debug output of the program to a dummy buffer. Only the mouth coordinates were allowed to be redirected to standard output by temporarily disabling the buffer.

7. The function of the library is edited to avoid the use of annotations.

8. The facial feature recognizer was edited to retrieve only the mouth coordinates and avoid rest of the facial features including the head pose.

9. Disable the above mentioned unused features to save execution time for the ROI extractor.

10. New debug functions were added to display the detected mouth and the mouth feature points.

In spite of the above mentioned modifications, the program was not able to recognize the face occasionally. For smooth run of the program, the previous successful face ROIs were saved and reused on failure of face detection. In addition to that, a provision to pass a face ROI was added which can be useful when face detection fails for initial frames.

By the above said major modifications and a few minor modifications including changes in the build, the library and the program was made perfectly suitable for the project purposes.

The executable of mouth ROI extractor can be run by using the command:

mouthRecog -i <input file> x y w h

Here, the input file is the image on which mouth recognition is to be done. x and y coordinates for the top left corner, w is the width and h is the height of the ROI of

previous frame which was successful in recognition. x, y, w and h are optional and is used when the output of the ROI extractor is corrupted in certain error scenarios.

Apart from the above mentioned detector, another facial feature detector was used in the first place. It is an opencv haar cascade mouth detector, create by Castrillon et al. Though it was possible to use this from the python codes, the detector was not stable enough for many subjects or, part of most of the subjects. The workarounds tried to make it better could not increase the performance to a reliable level. So, this implementation was eventually discarded and the detector from Matthias Dantone et al is used for the project purpose. The details of the original open source implementation can be found here: http://mozart.dis.ulpgc.es/Gias/modesto_eng.html

The algorithm follows in 3. It is described as a high level flow for understanding.

Algorithm 3 Mouth Recognition

```
1:  procedure MOUTHRECOG
2:      buffer ← save(currentBuffer)
3:      manualRoi ← read(manuallysetROI).
4:      imageFile ← read(imagefilename).
5:      readAndLoad(configuration file for Facial Feature Detection)
6:      readAndLoad(configuration file for head pose)
7:      readAndLoad(configuration file for face cascade)
8:      mouthRoi ← evaluateForest(imageFile, manualRoi, configFiles)
9:      restore(buffer)
10:     returnValue ← outputToConsole(mouthRoi)
11:     return(returnValue)
12: procedure EVALUATEFOREST
13:     gray ← convert(imageFile)
14:     faceRoi ← detectFace(gray)
15:     if size(faceRoi) = 0 then
16:         faceRoi ← manualRoi
17:     mouthRoi ← analyzeFace(gray, faceRoi)
18:     return(mouthRoi)
19: procedure DETECTFACE
20:     boundingBox ← detectMultiscale(imageFile)
21:     offsets ← findOffsets(boundingBox)
22:     faceRoi ← adjustFaceRoi(boundingBox, offsets)
23:     if size(faceRoi) ≠ 0 then
24:         faceRoi ← save(boundingBox)
25:         saveToDisk(faceRoi)
26:     else
27:         if isAvailableInDisk(faceRoi) then
28:             faceRoi ← readFromDisk(faceRoi)
29:     return(faceRoi)
30: procedure ANALYZEFACE
31:     gray ← convert(imageFile)
32:     faceRoi ← detectFace(gray)
33:     mouthRoi ← analyzeFace(gray, faceRoi)
34:     return(mouthRoi)
```

Highlighted lines indicate the modifications to the high level flow of the original program.

A cascade classifier[2] is a classifier consisting of several simple classifier stages. It will be applied to different regions of the image to detect an object of interest. It has the capability to be resized so as to be able to find the objects of interest of various sizes. It also has the capability to detect objects with different orientations. As can be seen in the algorithm, face detection is performed using a cascade classifier. From practice, it was seen that the face cascade classifier fails at times. To stabilize the program for mouth detection, the previous region of interest of face is used when face detection fails.

Median Filter

For this project the median filter is implemented in python and the algorithm is given in 4. This module is the first stage after the retrieval of the mouth ROI. It is further stabilized for aspect ratio before extracting mouth image.

Algorithm 4 Median Filter

procedure MEDIANFILTER
    $filterSize \leftarrow$ read window size
loop:
    $x, y, w, h \leftarrow$ current mouth ROI information
    if $initialized =$ False **then**
        $sequenceX = repeat(x, filterSize)$
        $sequenceY = repeat(y, filterSize)$
        $sequenceW = repeat(y, filterSize)$
        $sequenceH = repeat(y, filterSize)$
    else
        $addToSequence(x, y, w, h)$
    $x_m \leftarrow median(sequenceX)$
    $y_m \leftarrow median(sequenceY)$
    $w_m \leftarrow median(sequenceW)$
    $h_m \leftarrow median(sequenceH)$
    $return(x_m, y_m, w_m, h_m)$

Temporal Derivatives

Temporal derivatives are taken on the audio and video frames to model dynamic speech information. In case of audio, the temporal derivatives are taken just after the spectrogram stage. On the other hand, in case of video, spectrogram is taken after a PCA on

[2] http://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html

the mouth ROI. In either cases, to get the temporal derivative for a particular frame, the adjacent frames are also considered. The element-wise differences are computed for the frame with respect to the frame before and the frame after. On either sides, zeroed frames are appended to facilitate temporal derivatives for the first and the last frame. The algorithm for temporal derivatives is in 5for which the implementation is done in python.

Algorithm 5 Temporal Derivatives

procedure TEMPDERIVATIVE
  $frames \leftarrow read$(frames for a segment)
  $combine(zeroedFrame, frames, zeroedFrame)$
  $numberOfFrames \leftarrow length(frames)$
  $index \leftarrow 1$
loop:
  $frame_{current} \leftarrow readFrame(current)$
  $frame_{previous} \leftarrow readFrame(current - 1)$
  $frame_{next} \leftarrow readFrame(current + 1)$
  $temporalDerivative_1 = elementWiseDifference(frame_{current}, frame_{previous})$
  $temporalDerivative_2 = elementWiseDifference(frame_{current}, frame_{next})$
  $combinedFrames(index - 1) \leftarrow combine(temporalDerivative_1, frame_{current}, temporalDerivative_2)$
  if index $\neq$ numberOfFrames **then**
    $index \leftarrow index + 1$.
    goto *loop*.

PCA

PCA implementation is done in python using scikit-learn[3] library functions. PCA algorithm has two sections. One, is for finding the principal components and one, for the generation of reduced-dimension representation using the retrieved PCA dimensions using PCA whitening. Whitening makes sure that the processed data will have a diagonal covariance matrix and the covariance matrix is identity matrix. In scikit-learn library,*fit* function is used for generation of principal components. Since the data is large, this step is done here in batches using *IncrementalPCA* function. The number of principal component for video frames is 32 and audio frames is 100. Once principal components are generated, using *transform* function, the reduced-dimension representation for the given data is generated. In case of audio, original frames and the temporal derivatives are used as input PCA module. Whereas, with video, only the original frames which

[3] http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

are mouth ROIs are used as the input. The output of this stage is given to temporal derivative stage for video. For audio, the output goes to feature mean normalization stage.

The algorithm for principal component generation is given in 6. This is executed first after which whitening is done on the input data following algorithm 7.

Algorithm 6 Principal Components Generation

1: **procedure** PCA
2: $dataFile \leftarrow$ name of input matrix file
3: $frames \leftarrow loadMatrix(dataFile)$
4: $numberOfCompnents \leftarrow$ 23 for video, 100 for audio
5: $batchSize \leftarrow$ 10000
6: $whitening \leftarrow$ True
7: $pca \leftarrow initializePCA(numberOfCompnents, batchSize)$
8: $fit(pca, frames)$
9: $return(pca)$

Algorithm 7 Generation of whitened representation

1: **procedure** WHITENING
2: file $\leftarrow$ name of pca file
3: $pca \leftarrow loadPCA$(file)
4: $dataFile \leftarrow$ name of input matrix file for audio/video segment
5: $frames \leftarrow loadMatrix(dataFile)$
6: $whitenedFrames \leftarrow transform(pca, frames)$
7: $return(whitenedFrames)$

Average Pooling

Average pooling bridges between the output of neural networks to the linear SVM for classification. Since linear SVM can handle only fixed size inputs and the digit's representations are not fixed size, this module is mandatory. The implementation for this module is done in python and the pseudo code is given in 8.

At the output of neural networks, the representations of digits are generated as numpy [4] files which is compatible to use in python code. Average pooling is also coded in python and the program has the following syntax for execution:

[4] http://www.numpy.org/

It reads all the numpy files for train or test in one call and creates a file combining the digits, along with average pooling at the output. The output can be directed to a particular directory and the files are prefixed with the given prefix at the input. These files can directly be used with the classifier program.

Algorithm 8 Average Pooling

procedure AVERAGEPOOLING
    iDirectory ← input directory
    $files \leftarrow readNumpyFiles(iDirectory)$
    $labels \leftarrow readLabelFiles(iDirectory)$
    $index \leftarrow 1$
loop:
    $data \leftarrow load(files[index])$
    $(split1, split2, split3) \leftarrow splitData(data)$
    $averagedData \leftarrow averageFrames(data)$
    $averagedSplit1 \leftarrow averageFrames(split1)$
    $averagedSplit2 \leftarrow averageFrames(split2)$
    $averagedSplit3 \leftarrow averageFrames(split3)$

    $combinedData = combineData(averagedData, averagedSplit1, averagedSplit2, averagedSplit$
    $linearizedVector \leftarrow linearize(combinedData)$
    $featureArray \leftarrow mergeArray(featureArray, linearizedVector)$
    $featureLabelsVector \leftarrow mergeVector(featureLabelsVector, labels[index])$
    if length(files) ≠ index **then**
        $index \leftarrow index + 1.$
        goto *loop*.

A.2 Neural Networks

A.2.1 Reproducibility

Reproducibility in the current context refers to the ability to reproduce the experimental results published in a paper. Inability to reproduce the experimental results in a paper is a common issue among the machine learning researchers [90]. It is known that a few researchers have even failed to reproduce their own results. The experimental results are important as they play a 'central role' in the research community and strong reliance is placed on them.

When the configuration of neural networks is not made available publicly, the researchers use hyperparameter optimization methods such as grid search or random search. Independently of the type of hyperparameter optimization used, tuning the model cause the already published results in deep neural networks difficult to reproduce and extend [74]. Thus, the investigation becomes more of an art than a science. This is considered to be a hindrance to the progress of technology when it comes to large and multilayer networks. Such an art is the key for state-of-the-art techniques' performances, especially in cases such as image classification, and not strategies or innovative modelling in machine learning.

Many researchers came up with solutions to resolve the problems with reproducibility [90]. One of the solution is to make the source open to support reproducibility. Another solution proposed is to make the artifacts necessary in regenerating the results available with the submitted papers. Either way, there is a widespread agreement that the experimental results being reproducible is necessary for the advancement of technology. We can hope that the issue of reproducibility will be addressed aptly by the research community in the near future.

A.2.2 Contrastive Divergence

The probability of a state (**v,h**) for an RBM, where **v** is is a column vector containing the states of visible units, v_i and **h** is a vector containing the hidden states, h_i, is given by,

$$\mathrm{P}(\mathbf{v},\mathbf{h}|\theta) = \frac{1}{Z(\theta)} \exp[-\mathrm{E}(\mathbf{v},\mathbf{h}|\theta)] \ldots (1)$$

Where $\mathrm{Z}(\theta)$ is a normalizing constant, E is the energy function and θ is a parameter set that includes weight $\mathbf{W} = [w_{ij}]$, biases $\mathbf{b} = [b_i]$ and $\mathbf{c} = [c_j]$. Here, w_{ij} is the weight of the connection between v_i and h_j.

The parameters of RBM is estimated using Maximum Likelihood Estimation with the following gradient update rules,

$$\bigtriangledown w_{ij} = < v_i h_j >_d - < v_i h_j >_m \ldots (2)$$
$$\bigtriangledown b_i = < v_i >_d - < v_i >_m \ldots (3)$$
$$\bigtriangledown c_j = < h_j >_d - < h_j >_m \ldots (4)$$

$< . >_d$ is the expectation over data which is the distribution $P(h|v^(t), \theta)$. $< . >_m$ is the expectation over the model distribution, (1). Computing gradients exactly is computationally infeasible. To avoid computing exactly, a sampling method called Markov

Chain Monte Carlo (MCMC) method is used, which computes the gradients approximately. Since, in any RBM layer the neurons of the same layer are mutually independent, it is possible to sample from the whole layer at once. This restricted structure of RBM ensures that the Gibbs sampling is efficient in drawing samples from (1). Contrastive Divergence (CD) is a method to approximate the true gradients. CD computes the expectation $< . >_m$, using the samples obtained after n steps of Gibbs sampling. The CD gradient is approximated,

$$\bigtriangledown w_{ij} \approx < v_i h_j >_d - < v_i h_j >_{P_n}$$

Although, this method is biased, it works well in practice. So, it is commonly used for neural network training.

A.3 Experiments

A.3.1 Mouth Occlusion

The algorithm for mouth occlusion is explained below. Here, the mouth ROI that is pre-saved in the execution of reference stack is reused. The input to this module is an array containing mouth ROI and the output is an array containing occluded mouth ROI.

Algorithm 9 Occlusion of mouth ROI

```
1:  procedure OCCLUDEMOUTH
2:      file ← name of input matrix file for video
3:      skinPixels ← read(file(4columns))
4:      skinTone = average(SkinPixels)
5:      type ← type of occlusion

6:      if type = case a then
7:          storeInColumn(file, skinTone, (1, 16))

8:      if type = case b then
9:          storeInColumn(file, skinTone, (1, 16), (32, 48))

10:     if type = case c then
11:         storeInColumn(file, skinTone, (1, 16), (24, 40), (48, 64))

12:     if type = case d then
13:         storeInRow(file, skinTone, (1, 24))

14:     if type = case e then
15:         storeInRow(file, skinTone, (1, 42))
```

Here the *storeInColumn* is a function that stores the skin tone value into range of columns. The range of columns is passed by (start, end) pairs, where each pair indicates the start and end of one occluded area. The start and end used in here are computed based on the percentage of occlusion required by the experiment. The percentages are calculated for 80x60 pixel mouth ROI.

Bibliography

[1] Keith Waters and Thomas M Levergood. Decface,"an automatic lip-synchronization algorithm for synthetic faces," digital equipment corporation, cambridge research lab. 1993.

[2] Sarah Hawkins. Situational influences on rhythmicity in speech, music, and their interaction. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 369(1658):20130398, December 2014. ISSN 0962-8436. doi: 10.1098/rstb.2013.0398. URL http://europepmc.org/articles/PMC4240965.

[3] M. K. Jensen and S. S. Nielsen. Speech reconstruction from binary masked spectrograms using vector quantized speaker models. 2006. URL http://www2.imm.dtu.dk/pubdb/p.php?4725. Supervised Lars Kai Hansen, IMM.

[4] Thomas Nodes and Neal Gallagher. Median filters: Some modifications and their properties. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 30(5): 739–746, 1982.

[5] James Henry. Springer handbook of auditory research. *International Journal of Audiology*, 52(3):207, 2013. doi: 10.3109/14992027.2012.754109. URL http://dx.doi.org/10.3109/14992027.2012.754109.

[6] Sohee Park, Hansung Lee, Jang-Hee Yoo, Geonwoo Kim, and Soonja Kim. Partially occluded facial image retrieval based on a similarity measurement. *Mathematical Problems in Engineering*, 2015:11, 2015. doi: 10.1155/2015/217568.

[7] Marwa Mahmoud and Peter Robinson. Interpreting hand-over-face gestures. pages 248–255, 2011.

[8] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal Deep Learning. *Proceedings of The 28th International Conference on Machine Learning (ICML)*, pages 689–696, 2011.

[9] Steve Stemler. An overview of content analysis. *Practical assessment, research & evaluation*, 7(17):137–146, 2001.

[10] Alan Hanjalic. Content-based analysis of digital video. pages I–XII, 1–193, 2004.

[11] Yao Wang, Zhu Liu, and Jin-Cheng Huang. Multimedia content analysis-using both audio and visual clues. *IEEE signal processing magazine*, 17(6):12–36, 2000.

[12] Jean-Claude Junqua and Jean-Paul Haton. *Robustness in Automatic Speech Recognition: Fundamentals and Applications*. Kluwer Academic Publishers, Norwell, MA, USA, 1995. ISBN 0792396464.

[13] MA Anusuya and Shriniwas K Katti. Speech recognition by machine, a review. *arXiv preprint arXiv:1001.2267*, 2010.

[14] Dong Yu and Li Deng. Hidden markov models and the variants. *Automatic Speech Recognition*, pages 23–54, 2015.

[15] Li Deng and Xiao Li. Machine learning paradigms for speech recognition: An overview. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(5): 1060–1089, 2013.

[16] Li Deng and Navdeep Jaitly. Deep discriminative and generative models for pattern recognition. November 2015. URL https://www.microsoft.com/en-us/research/publication/deep-discriminative-and-generative-models-for-pattern-recognition/.

[17] Harry McGurk and John MacDonald. Hearing lips and seeing voices. *Nature*, 264: 746–748, 1976.

[18] Alejandro Jaimes and Nicu Sebe. Multimodal human–computer interaction: A survey. *Computer vision and image understanding*, 108(1):116–134, 2007.

[19] Dongge Li, Nevenka Dimitrova, Mingkun Li, and Ishwar K Sethi. Multimedia content processing through cross-modal association. pages 604–611, 2003.

[20] Gerasimos Potamianos, Chalapathy Neti, Juergen Luettin, and Iain Matthews. Audio-Visual Automatic Speech Recognition: An Overview. *Issues in Visual and Audio Visual Speech Processing*, pages 1–30, 2004. URL http://researchweb.watson.ibm.com/AVSTG/MITBOOK04_REVIEW.pdf.

[21] William H Sumby and Irwin Pollack. Visual contribution to speech intelligibility in noise. *The Journal of the Acoustical Society of America*, 26(2):212–215, 1954.

[22] Hervé Bredin and Gérard Chollet. Audiovisual speech synchrony measure: application to biometrics. *EURASIP Journal on Applied Signal Processing*, 2007(1): 179–179, 2007.

[23] Sharon Oviatt and Philip Cohen. Perceptual user interfaces: multimodal interfaces that process what comes naturally. *Communications of the ACM*, 43(3):45–53, 2000.

[24] George Papandreou, Athanassios Katsamanis, Vassilis Pitsikalis, and Petros Maragos. Adaptive multimodal fusion by uncertainty compensation with application to audiovisual speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(3):423–435, 2009.

[25] Pradeep K Atrey, M Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia systems*, 16(6):345–379, 2010.

[26] Li Deng and Dong Yu. Deep Learning: Methods and Applications. *Foundations and Trends in Signal Processing*, 7(3-4):197—-387, 2013. ISSN 09598138. doi: 10.1136/bmj.319.7209.0a.

[27] Li Deng. Three Classes of Deep Learning Architectures and Their Applications: A Tutorial Survey. *Research.Microsoft.Com*, 2013. URL http://research.microsoft.com/pubs/192937/Transactions-APSIPA.pdf.

[28] Nitish Srivastava and Ruslan Salakhutdinov. Learning representations for multimodal data with deep belief nets. *International Conference on Machine Learning Workshop*, 2012.

[29] Z G a O C Xie Y Du. Restricted Boltzmann Machines with Svm for Object Recognition. 21:9199–9206, 2014. doi: 10.12733/jcis12276.

[30] Kuniaki Noda, Yuki Yamaguchi, Kazuhiro Nakadai, Hiroshi G Okuno, and Tetsuya Ogata. Audio-visual speech recognition using deep learning. *Applied Intelligence*, 42(4):722–737, 2015.

[31] Youssef Mroueh, Etienne Marcheret, and Vaibhava Goel. Deep multimodal learning for audio-visual speech recognition. pages 2130–2134, 2015.

[32] Steven Greenberg, William A Ainsworth, Arthur N Popper, and Richard R Fay. Speech processing in the auditory system. 18, 2004. doi: 10.1007/b97399.

[33] Vikramjit Mitra, Julien Van Hout, Wen Wang, Martin Graciarena, Mitchell McLaren, Horacio Franco, and Dimitra Vergyri. Improving robustness against reverberation for automatic speech recognition. pages 525–532, 2015.

[34] Ritwik Giri, Michael L Seltzer, Jasha Droppo, and Dong Yu. Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning. pages 5014–5018, 2015.

[35] Takaaki Ishii, Hiroki Komiyama, Takahiro Shinozaki, Yasuo Horiuchi, and Shingo Kuroiwa. Reverberant speech recognition based on denoising autoencoder. pages 3512–3516, 2013.

[36] Andreas Schwarz, Christian Huemmer, Roland Maas, and Walter Kellermann. Spatial diffuseness features for dnn-based speech recognition in noisy and reverberant environments. pages 4380–4384, 2015.

[37] Xue Feng, Yaodong Zhang, and James Glass. Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition. pages 1759–1763, 2014.

[38] Ahmad Basheer Hassanat. Visual words for automatic lip-reading. *arXiv preprint arXiv:1409.6689*, 2014.

[39] Xingjie Wei, Chang-Tsun Li, and Yongjian Hu. Face recognition with occlusion using dynamic image-to-class warping (dicw). pages 1–6, 2013.

[40] Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey Hinton. Robust boltzmann machines for recognition and denoising. pages 2264–2271, 2012.

[41] Lele Cheng, Jinjun Wang, Yihong Gong, and Qiqi Hou. Robust deep auto-encoder for occluded face recognition. pages 1099–1102, 2015.

[42] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deeply learned face representations are sparse, selective, and robust. pages 2892–2900, 2015.

[43] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2012.

[44] Brian Hutchinson, Li Deng, and Dong Yu. A deep architecture with bilinear modeling of hidden representations: Applications to phonetic recognition. pages 4805–4808, 2012.

[45] Ossama Abdel-Hamid, Abdel-Rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10):1533–1545, 2014.

[46] Honglak Lee, Chaitanya Ekanadham, and Andrew Y Ng. Sparse deep belief net model for visual area v2. pages 873–880, 2008.

[47] Eric K Patterson, Sabri Gurbuz, Zekeriya Tufekci, and John N Gowdy. Cuave: A new audio-visual database for multimodal human-computer interface research. 2: II–2017, 2002.

[48] Christopher T A T T Wadsworth. The editor's toolkit : a hands-on guide to the craft of film and TV editing. 2016. URL SafariBooksOnlinehttp://proquest.safaribooksonline.com/?fpi=9781317367758Volltexthttp://proquest.tech.safaribooksonline.de/9781317367758.

[49] Goran Ferenc. Dynamic properties of musical signals in genre ethnic electronica. *INFOTEH JAHORINA*, 11, 2012.

[50] Tontechnik Rechner. Decibel Table, SPL, Loudness Comparison Chart. URL http://www.sengpielaudio.com/TableOfSoundPressureLevels.htm.

[51] Hilmi R Dajani, Willy Wong, and Hans Kunov. Fine structure spectrography and its application in speech. *The Journal of the Acoustical Society of America*, 117(6): 3902–3918, 2005.

[52] C. Berry and N. Harte. Region of interest extraction using colour based methods on the cuave database. *IET Irish Signals and Systems Conference (ISSC 2009)*, pages 1–6, June 2009. doi: 10.1049/cp.2009.1715.

[53] Matthias Dantone, Juergen Gall, Gabriele Fanelli, and Luc Van Gool. Real-time facial feature detection using conditional regression forests. pages 2578–2585, 2012.

[54] Mihaela Gordan, Constantine Kotropoulos, and Ioannis Pitas. A support vector machine-based dynamic network for visual speech recognition applications. *EURASIP Journal on Applied Signal Processing*, 2002(1):1248–1259, 2002.

[55] Mihai Gurban and Jean-Philippe Thiran. Audio-visual speech recognition with a hybrid svm-hmm system. pages 1–4, 2005.

[56] Hynek Hermansky. Speech recognition from spectral dynamics. *Sadhana*, 36(5): 729–744, 2011.

[57] Elana M Zion Golumbic, David Poeppel, and Charles E Schroeder. Temporal context in speech processing and attentional stream selection: a behavioral and neural perspective. *Brain and language*, 122(3):151–161, 2012.

[58] Andrew Hunt and Richard Favero. Using principal component analysis with wavelets in speech recognition. pages 296–301, 1994.

[59] Tetsuya Takiguchi and Yasuo Ariki. Pca-based speech enhancement for distorted speech recognition. *Journal of multimedia*, 2(5):13–18, 2007.

[60] Stephen A. Zahorian and Hongbing Hu. Nonlinear dimensionality reduction methods for use with automatic speech recognition. *Speech Technologies, Prof. Ivo Ipsic (Ed.), InTech*, 2011. doi: 10.5772/16863.

URL https://www.intechopen.com/books/speech-technologies/nonlinear-dimensionality-reduction-methods-for-use-with-automatic-speech-recognit

[61] M.a. Turk and a.P. Pentland. Face Recognition Using Eigenfaces. 3(1):72 – 86, 1991. ISSN 1063-6919. doi: 10.1109/CVPR.1991.139758.

[62] Paul Duchnowski, Uwe Meier, and Alex Waibel. See me, hear me: integrating automatic speech recognition and lip-reading. 94:547–550, 1994.

[63] Gerasimos Potamianos, Chalapathy Neti, Giridharan Iyengar, Andrew W Senior, and Ashish Verma. A cascade visual front end for speaker independent automatic speechreading. *International Journal of Speech Technology*, 4(3-4):193–208, 2001.

[64] Gerasimos Potamianos, Hans Peter Graf, and Eric Cosatto. An image transform approach for hmm based automatic lipreading. pages 173–177, 1998.

[65] Gerasimos Potamianos, Chalapathy Neti, Juergen Luettin, and Iain Matthews. Audio-visual automatic speech recognition: An overview. *Issues in visual and audio-visual speech processing*, 22:23, 2004.

[66] Guoping Li, Mark E Lutman, Shouyan Wang, and Stefan Bleeck. Relationship between speech recognition in noise and sparseness. *International journal of audiology*, 51(2):75–82, 2012.

[67] Nitish Srivastava. Deepnet library. 2012. URL http://www.cs.toronto.edu/~nitish/deepnet/.

[68] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. pages 111–118, 2010.

[69] Syed Zubair, Fei Yan, and Wenwu Wang. Dictionary learning based sparse coefficients for audio classification with max and average pooling. *Digital Signal Processing*, 23(3):960–970, 2013.

[70] Yichuan Tang. Deep learning using linear support vector machines. *In ICML*, 2013.

[71] Yuxuan Wang and DeLiang Wang. Towards scaling up classification-based speech separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(7): 1381–1390, 2013.

[72] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425, 2002.

[73] Michael S. Lewicki and Terrence J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000. doi: 10.1162/089976600300015826. URL http://dx.doi.org/10.1162/089976600300015826.

[74] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[75] Geoffrey Hinton. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1):926, 2010.

[76] Ilker Yildirim. Bayesian inference: Gibbs sampling. *Technical Note, University of Rochester*, 2012.

[77] Rui Min, Abdenour Hadid, and Jean-Luc Dugelay. Efficient detection of occlusion prior to robust face recognition. *The Scientific World Journal*, 2014, 2014.

[78] P Coxhead. Natural language processing ans applications phones and phonemes. *Phonemes*, pages 1–13, 2006.

[79] Say Wei Foo, Yong Lian, and Liang Dong. Recognition of visual speech elements using adaptively boosted hidden markov models. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5):693–705, 2004.

[80] Michiel Visser, Mannes Poel, and Anton Nijholt. Classifying visemes for automatic lipreading. pages 349–352, 1999.

[81] Tsuhan Chen. Audiovisual speech processing. *IEEE Signal Processing Magazine*, 18(1):9–21, 2001.

[82] Hisao Kuwabara. Acoustic properties of phonemes in continuous speech for different speaking rate. 4:2435–2438, 1996.

[83] JC Rutledge. Fundamentals of speech recognition, by lawrence rabiner and biing-hwang juang. *ANNALS OF BIOMEDICAL ENGINEERING*, 23:526–526, 1995.

[84] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. pages 6645–6649, 2013.

[85] Bridget Smith. An acoustic analysis of voicing in american english dental fricatives. *OSUWPL, Volume 60*, pages 117–128, Spring 2013.

[86] William Labov and Maciej Baranowski. An unsolved problem in chain shifting. 2004.

[87] David Deterding. Measurements of the/ei/and/eu/vowels of young english speakers in singapore. *The English Language in Singapore, Research on Pronunciation*, pages 93–99, 2000.

[88] Renáta Gregová. Quantity in slovak and in british english. *SKASE Journal of Theoretical Linguistics;2008, Vol. 5 Issue 1*, page 17, June 2008.

[89] Don C Utulu. Monophthongisation and vowel lengthening processes in educated urhobo english: A moraic account. 2014.

[90] Chris Drummond. Replicability is not reproducibility: Nor is it good science. *Proc. of the Evaluation Methods for Machine Learning Workshop at the 26 ICML, Montreal, Canada*, 2009.