London South Bank University
School of Engineering
Division of Computer Science and Informatics

Viseme-based Lip-Reading using Deep Learning

Souheil Fenghour

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computer Science and Informatics at
London South Bank University, November 2021

Abstract

Research in Automated Lip Reading is an incredibly rich discipline with so many facets that have been the subject of investigation including audio-visual data, feature extraction, classification networks and classification schemas. The most advanced and up-to-date lip-reading systems can predict entire sentences with thousands of different words and the majority of them use ASCII characters as the classification schema. The classification performance of such systems however has been insufficient and the need to cover an ever expanding range of vocabulary using as few classes as possible is challenge.

The work in this thesis contributes to the area concerning classification schemas by proposing an automated lip reading model that predicts sentences using visemes as a classification schema. This is an alternative schema to using ASCII characters, which is the conventional class system used to predict sentences. This thesis provides a review of the current trends in deep learning-based automated lip reading and analyses a gap in the research endeavours of automated lip-reading by contributing towards work done in the region of classification schema. A whole new line of research is opened up whereby an alternative way to do lip-reading is explored and in doing so, lip-reading performance results for predicting sentences from a benchmark dataset are attained which improve upon the current state-of-the-art.

In this thesis, a neural network-based lip reading system is proposed. The system is lexicon-free and uses purely visual cues. With only a limited number of visemes as classes to recognise, the system is designed to lip read sentences covering a wide range of vocabulary and to recognise words that may not be included in system training. The lip-reading system predicts sentences as a two-stage procedure with visemes being recognised as the first stage and words being classified as the second stage. This is such that the second-stage has to both overcome the one-to-many mapping problem posed in lip-reading where one set of visemes can map to several words, and the problem of visemes being confused or misclassified to begin with.

To develop the proposed lip-reading system, a number of tasks have been performed in this thesis. These include the classification of continuous sequences of visemes; and the proposal of viseme-to-word conversion models that are both effective in their conversion performance of predicting words, and robust to the possibility of viseme confusion or misclassification. The initial system reported has been testified on the challenging BBC Lip Reading Sentences 2

(LRS2) benchmark dataset attaining a word accuracy rate of 64.6%. Compared with the state-of-the-art works in lip reading sentences reported at the time, the system had achieved a significantly improved performance.

The lip reading system is further improved upon by using a language model that has been demonstrated to be effective at discriminating between homopheme words and being robust to incorrectly classified visemes. An improved performance in predicting spoken sentences from the LRS2 dataset is yielded with an attained word accuracy rate of 79.6% which is still better than another lip-reading system trained and evaluated on the the same dataset that attained a word accuracy rate 77.4% and it is to the best of our knowledge the next best observed result attained on LRS2.

Acknowledgements

There are a handful of people that I would like to give special mention to:

- I would like to express my gratitude to my supervisor Dr Daqing Chen for his consistent support and guidance throughout the duration of the research project, my second supervisor Dr Perry Xiao and Dr Kun Guo who has given me a lot of advice and assistance.

- I also acknowledge the support of other researchers whom I have been closely associated with including Laureta Hajderanj and Isakh Weheliye.

- I would like to thank my parents and my sisters for their support support throughout the four years of my PhD studies.

- Finally I would like to thank the sponsors of this research Chinasoft International and London South Bank University for providing the funds to allow me to undertake this research

Contents

List of Tables

List of Figures

List of Abbreviations

Abbrevation	Meaning
2D CNN	Two-Dimensional Convolutional Neural Network
3D CNN	Three-Dimensional Convolutional Neural Network
AAM	Active Appearance Model
ASM	Active Shape Model
Bi-Conv-LSTM	Bidirectional Convolutional Long Short Tem Memory
Bi-GRU	Bidirectional Gated Recurrent Unit
Bi-LSTM	Bidirectional Long Short Tem Memory
BN	Batch Normalisation
CER	Character Error Rate
CFI	Concatenated Frame Image
CNN	Convolutional Neural Network
CTC	Connectionist Temporal Classification
DBN	Deep Belief Network
DBNF	Deep Bottleneck Features
DCT	Direct Cosine Transformation
DNN	Deep Neural Network
GMM	Gaussian Mixture Model
GRU	Gated Recurrent Unit
HiLDA	Hierarchical Linear Discriminant Analysis
HMM	Hidden Markov Model
LDA	Linear Discriminant Analysis
LSTM	Long Short Tem Memory
MD-ATT-MC	Modality Attention Mechanism
MFCC	Mel-Frequency Cepstral Coefficient
MLLT	Maximum Likelihood Linear Transform
N/A	Not Applicable
NIN	Network in Network
PCA	Principle Component Analysis
RBM	Restricted Boltzmann machine
Res-Bi-Conv-LSTM	Residual Bidirectional Convolutional Long Short Tem Memory
RNN	Reccurrent Neural Network
ROI	Region of Interest
SAR	Sentence Accuracy Rate
SAT	Speaker Adaptive Training
SER	Sentence Error Rate
ST-PCA	Sparse Tensor Principle Component Analysis
SVM	Support Vector Machine
TCN	Temporal Convolution Network
TM-CTC	Transformer Connectionist Temporal Classification
TM-Seq2seq	Transformer Sequence-to-Sequence
VER	Viseme Error Rate
WER	Word Error Rates
WFST	Weighted Finite-State Transducer

Chapter 1

Introduction

1.1 Research Motivation, Aims and Objectives

Visual Speech Recognition, or Lip Reading, plays an important role in human communication - especially in noisy environments where audio speech recognition may be difficult. It is extremely useful for people whose hearing is impaired, for those who are autistic and for those suffering from language impairment - especially given that many people with hearing problems are unable to sign [1].

For almost forty years, people have implemented many approaches to automate the task of lip reading using machine learning and the potential areas [2] [3] [4] that would benefit from automated lip reading software is numerous. Automated lip reading would be beneficial for the police if ever they needed to decipher CCTV footage of people speaking when there is no audio available [5]. There is an entire branch of forensic speech reading devoted to the purposes of gathering forensic evidence where professional lip readers are employed to interpret inaudible speech [6] [7]. Other beneficiaries of automated lip reading include autistic people with reduced lip-reading abilities, as well as people with conditions like Williams Syndrome [8] or Specific Language Impairment [9].

Sumby and Pollack [10] proposed the basic theory of lip-reading in 1954 where it was first

suggested that the features of lip motion could be used to identify a speaker's speech content. One of the first automated speech recognition systems to be constructed was by Petajan in 1984 who used a geometric feature based extraction method [11] with height, width, area and perimeter of a speaking person's mouth all being extracted. These visual features were combined with audio features that had been extracted from audio to classify speech.

Significant breakthroughs in the performance of automated lip reading systems have been made in the last fourteen years thanks to developments of deep neural networks and the emergence of large-scale databases covering vocabularies with thousands of different word. Lip-reading systems have evolved from recognising isolated speech units in the form of digits and letters to decoding entire sentences.

In 2011, deep learning-based feature extraction was introduced into visual speech recognition for the first time as Ngiam et al. [12] proposed an audio-visual speech recognition system based on Restricted Boltzmann Machines(RBMs) [13]. In 2014, Noda et al. [14] used Convolutional Neural Networks(CNNs) as a feature extraction method for the lip-reading of people speaking in videos that had been sampled into image frames. The experimental results indicated that the visual features obtained through the use of a CNN were significantly better than traditional methods like Principal Component Analysis. In 2016, Wand et al. [15] used a Long Short-Term Memory(LSTM) for lip-reading and a word recognition rate of 79.6% was achieved on the GRID audio-visual corpus. In 2017, Assael et al. [16] proposed the LipNet model consisting of a spatial-temporal convolution network and Recurrent Neural Network(RNN) with a CTC used as the network loss function. Also in 2017 [17], Chung et al, proposed the WLAS network which is composed of a CNN and RNNs and word error rates of 50.2%, 23.8% and 3.0% were obtained on the BBC-LRS2, BBC-LRW and GRID databases.

Lip-reading systems typically follow a framework where there is a frontend for feature extraction, a backend for classification and some pre-processing at the start. Stages of automated lip-reading are outlined in Figure 1.1 and include the following steps:

- Visual Input - Videos of people speaking are sampled into image frames representing speech to be decoded.

- Pre-processing - This is where the region of interest (ROI), i.e., the lips are located and extracted from the raw image data. This involves detecting the face, locating the lips and extracting the lip region from the video image. Some basic transformations are applied to the ROI such as cropping to reduce the number of overall operations needed for training and validation.

- Feature Extraction (Frontend) - This involves extracting effective and relevant features from redundant features and the mapping of high dimensional image data into a lower dimensional representation.

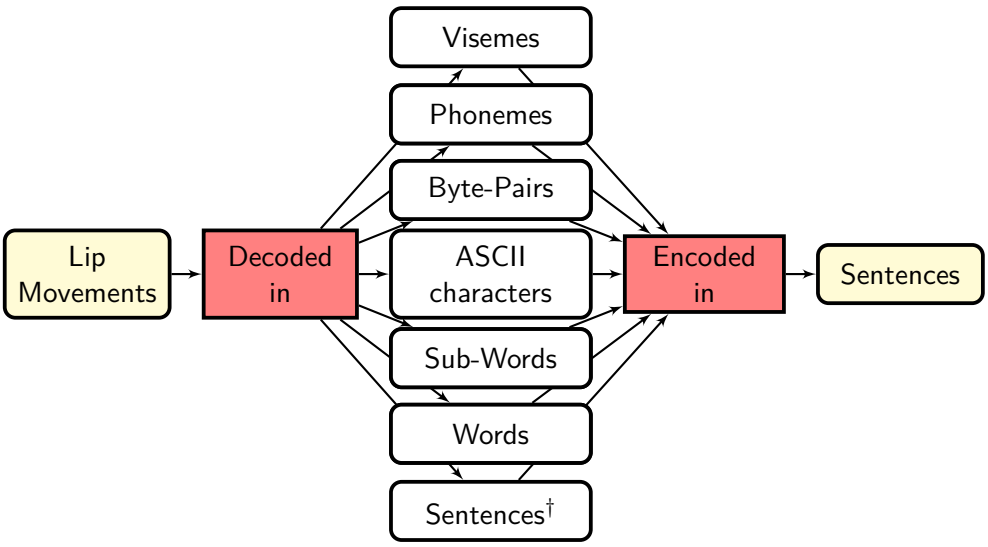
- Classification (Backend) - This involves ascribing speech to facial movements that have been transformed into a lower dimensional feature vector.

- Decoded Speech - Speech is decoded in classes or units and eventually encoded as spoken words or sentences.

Figure 1.1: General framework for automated lip-reading.

When it comes to the decoding of speech in automated lip reading, the lip movements can be interpreted in different ways and so different classification schemas have been introduced into the domain. Figure 1.2 illustrates the various interpretations of lip movements and classification schemas used for lip-reading.

Lip-reading systems that are designed to decode digits and letters have used individual words as classes and there are lip reading systems designed to decode a limited number of phrases that have encoded each phrase as a class. The emergence of large-scale audio-visual datasets covering thousands of words has meant that some of the recent state-of-the-art lip-reading systems can decode speech from people uttering thousands of different possible words. To do this, such system use individual ASCII characters as classes where words are predicted by learning the conditional probability relationship between characters as encoding every individual word as a class would be impractical.



† Some lip-reading systems only decode a limited number of phrases

Figure 1.2: Different classification schemas.

The subject of this thesis is about the use of visemes as a classification schema in automated lip reading whereby a neural network-based lip reading system has been constructed that decodes videos of people speaking entire sentences by predicting the spoken visemes. The proposed lip-reading system must be able to decode natural sentences from a benchmark dataset covering a vocabulary range with thousands of words and that features both profile and frontal videos. The proposed lip-reading system must attain a good accuracy not only for the classification of individual visemes but for also for the correct prediction of words. The prediction of words is a bottleneck in itself that needs to be overcome because many words share identical visemes.

Using visemes for lip reading sentences has some unique advantages. The use of visemes as classes in comparison to the use of either words or ASCII characters as classes requires an overall smaller number of classes which alleviates bottleneck in the computation. In addition, using visemes does not require pre-trained lexicons, meaning that a viseme-based lip reading system can be used to classify words that have not presented in the training phase, and they can be generalised to different languages because many different languages share the same visemes.

On the other hand, there are some specific issues to be considered when designing a viseme-based lip reading system for sentences. The general classification performance for individual segmented visemes has been less satisfactory in comparison to the classification of words due to

the fact that visemes tend to have a shorter duration than words. This results in there being less temporal information available to distinguish between different classes, as well as there being more visual ambiguity when it comes to class recognition [18]. One possible way to address this problem is to significantly increase the training data available to enhance the system's ability to distinguish between classes, and this is why a high volume of training videos have been utilised. Moreover, there is a direct conversion of recognised ASCII characters to possible words in a one-to-one mapping relationship, whereas this one-to-one mapping relationship does not exist when using visemes, because one set of visemes can map to multiple different sounds or phonemes. This also means that once visemes have been classified, there is still the need to perform a viseme-to-word conversion. This approach also helps to distinguish between homopheme words or words that look the same when spoken but sound different [19], a phenomenon that exists because of the one-to-many mapping relationship between visemes and phonemes.

One of the main reasons that visemes have not been widely deployed for use as classes in neural network-based lip-reading systems is that there is no universally agreed upon convention for defining visemes. Many practical techniques have been used to formally define visemes whether it is by grouping lip movements together through the use of articulatory gestures, such as lips closing together, jaw movement and teeth exposure; or by grouping together phonemes that produce lip movements that are visually similar. However, attempts to map phonemes to viseme across multiple speakers who have varying appearances of lips has resulted in different phoneme-to-viseme mappings being generated and thus different viseme conventions having been proposed [20] [21].

The biomechanics of lip-movements are complex and beyond the scope of this thesis but it was Alexander Graham Bell [22] [23] who first hypothesised that multiple phonemes may be visually identical on a given speaker. This was later verified and it gave rise to the concept of a viseme [24] [23]. The visually apparent features of lip movements for visual speech are controlled by the same articulatory organs that control audible speech i.e. the lips, teeth, tongue, jaw, velum, larynx, and lungs [25]. However unlike audible speech, only the lips, teeth, jaw and tongue are directly visible for lip-reading.

The English language consists of phonemes that are non-existent in other languages and one can likewise find phonemes in other languages that are not present in English. However, phonemes that are present in English will often share identical lip movements and thus map to the same visemes so it is possible to deploy a viseme convention that can be used across many different languages whereby every possible acoustic sound will correspond to viseme belonging to a fixed set of visemes [26].

1.2 Research Questions

Lip reading systems have evolved from recognising small isolated speech segments to predicting entire sentences that consists of thousands of different words. This means that a system designed to predict words from a vocabulary set will need to have been trained to predict those specific words during the training phase. Lip reading systems are generally confined to a fixed-lexicon of vocabulary. Whilst lip-reading systems have recorded good results for individual word classification, the endeavour to attain good performances for decoding entire sentences has been more of a challenge.

Lip reading systems that classify individual words can simply encode each individual word as class. This would however be impractical to implement when predicting entire sentences as every possible word would need to be encoded as a class and some of the most up-to-date lip reading datasets consist of vocabularies covering thousands of different words.

One possible classification scheme that is often used for systems to decode sentences is ASCII characters. Words and sentences can be treated as sequences of ASCII characters and lip-reading systems that predict words as a series of ASCII characters are reliant on learning the conditional probability relationship that exists between combinations of ASCII characters. Even the use of ASCII characters to predict words itself does rely on the system having been trained on a set of vocabulary to decode words from that vocabulary set. This is because ASCII-based lip-reading systems predict words as sequences of ASCII characters, so it is necessary for the sequence to have been observed during training for it to be detected during validation.

The issue of possible classification schemas to be used in lip-reading is one that topic that does deserve more attention. Visemes are one of many alternative classification schemas that can be used for automated lip-reading and there does not appear to be many previous works devoted to examining the application of visemes as classes in lip-reading despite there being many advantages to the use of visemes(discussed in Section 1.1).

The use of visemes to predict sentences does come with a bottleneck in that because of the one-to-many mapping relationship between visemes and phonemes, one set of visemes can correspond to multiple words and Figure **??** shows a distribution of visemes and phonemes for the LRS2 corpus illustrating the ambiguity where several sounds all look the same. So when visemes have been classified, there is still the need to determine the words that were spoken and because roughly half of words share identical visemes [27], a language model is needed to disambiguate between homopheme words.

For written speech, it may be sufficient in some languages to decode the identity of a character by simply knowing the identity of the character before like when performing Optical Character Recognition for Arabic characters whereby the appearance of a character is governed only by the identity of a character before it, though this is not the case for spoken speech. This is why language models need as much context as possible.

To accurately predict spoken words from visemes, a language model is required that is both:

1. Effective at disambiguating between homopheme words
2. Robust to the possibility of misclassified visemes as an input

A lip-reading system that predicts entire spoken sentences using visemes as an intermediate classification schema has many challenges. One of these hurdles is the fact that half of words in the English language have are homopheme words and they share identical visemes to other words [27]. This means that even with a 100% classification accuracy for visemes, the identities of one half of words can in be known while the identity of the other half will still be uncertain because of the one-to-many mapping relationship between visemes and words.

The disambiguation of homopheme words must be performed using context and probabilistic

language models because when matching sets of visemes to words, it is expected that the most likely set of words to have been uttered are the combination of words that make the most sense grammatically and this is the output that would be expected for a lip-reading systems decoding words using visemes. For example the words "time" and "type" share visemes and to determine the most likely words to have been uttered, it is necessary to know the possible identity of surrounding words as the word combination "for a brief time" has greater likelihood of being spoken than "for a brief type".

Of course, the prediction of spoken words in real time by simply knowing which visemes were spoken relies on visemes having been decoded with 100% precision. We can try to maximum performance accuracy for viseme classification to achieve the ideal case as much as possible but in reality, this is not always possible. A lip-reading system that predicts words by classifying visemes has to be robust to the possibility of visemes not having been classified correctly.

A lip-reading system that classifies speech in two stages with visemes predicted in the first stage and words in the second has to be prone to the possibility of errancy in the first stage. Visemes being incorrectly decoded not only causes words matching to that set of visemes to be predicted in correctly but can also cause successive words to be incorrectly predicted as language models doing the viseme-to-word conversion often predict words in combination. One word being predicted incorrectly can cause result in other successive words being predicted incorrectly too.

In this thesis, the question of "can we attain a good accuracy for lip reading sentences using solely visual cues?" is posed. There are many relevant points to address:

(**Q1**): What are the benefits of using visemes to lip-read compared with using other units of speech and classification schemas?

(**Q2**): Can a good classification performance of individual visemes be attained such that following conditions are fulfilled?

- Visemes are uttered from profile and frontal views

- For a video sampled at 25 fps, individual visemes can be of different durations. This also means that because the start time and stop time of every individual viseme is unknown, the viseme classification model must be able to perform temporal alignment
- The viseme classifier must be somewhat robust to lighting variations present within videos but it also must have good generalisation capabilities

(**Q3):** What are the different language models available used to predict words from images of lip movements?

(**Q4):** Can a language model be implemented that is effective at converting visemes to correct words? It must fulfil the following criteria:

- For words that have a unique set of visemes(approximately half of words in the English language), the classification of these words must be sufficient
- The conversion model must be effective at disambiguating between homopheme words(for example "able" and "epoch")

(**Q5):** Can a language model be implemented that is robust to misclassified visemes?

- The language model used for predicting spoken words given the recognised visemes must be prone to the possibility of visemes not being classified correctly at the current time step
- The words at any point in a sequence for a particular time step being predicted must prone to the possibility of earlier words in the sequence not being predicted correctly for previous time steps

(**Q6):** Can a good overall performance be attained for word classification when predicting sentences from some of the most challenging audio-visual datasets without audio?

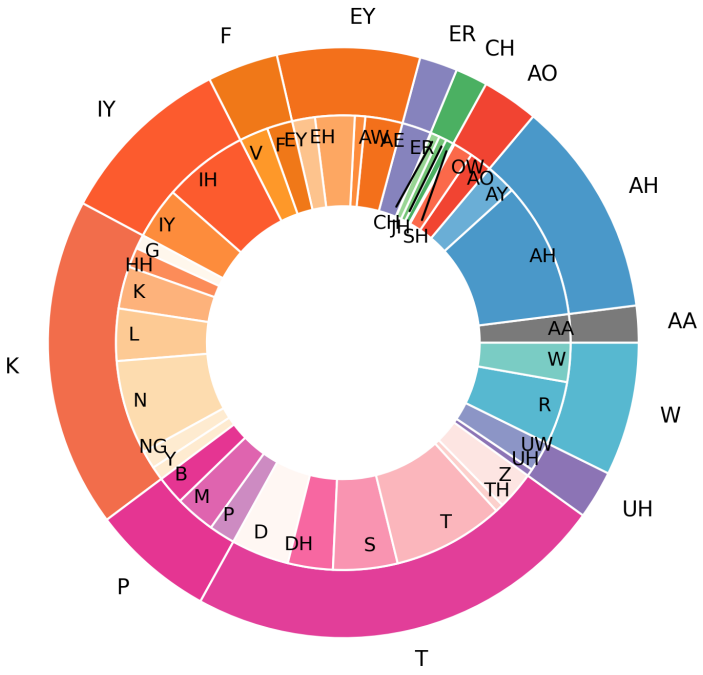


Figure 1.3: Distribution of phonemes(inner circle) and visemes(outer circle) in the LRS2 corpus.

1.3 Contributions

The novel contributions of this thesis to the area of automated lip reading is a lip-reading system that has been developed with unique features which include:

- The classification of sentences in continuous speech from videos of people speaking from both profile and frontal viewpoints with good precision including individual word classification accuracy compared with state-of-the-art approaches
- A classification system based purely on visual cues that predicts sentences by classifying individual visemes
- A system that does not require a pre-trained lexicon and can be applied to people speaking in different languages
- The classification of visemes in continuous speech with good precision using a specially designed transformer with a unique topology

- It addresses the one-to-many problem in lip reading where visemes are converted to words using a language model that is effective at disambiguating between homopheme words by using perplexity and by learning semantic and syntactic information where words are correctly predicted with good accuracy

- It uses a language model for converting visemes to words with a good level of robustness to the possibility of incorrectly classified visemes

1.4 Publications

Below are a list of my publications:

- Peer-reviewed journals:

    1. S. Fenghour, D. Chen, K. Guo, B. Li and P. Xiao. (2021). An Effective Conversion of Visemes to Words for High-Performance Automatic Lipreading. Sensors, 21, 7890. https://doi.org/10.3390/s21237890

    2. S. Fenghour, D. Chen, K. Guo, B. Li and P. Xiao. (2021). Deep Learning-Based Automated Lip-Reading: A Survey," in IEEE Access, vol. 9, pp. 121184-121205, doi: 10.1109/ACCESS.2021.3107946.

    3. S. Fenghour, D. Chen, K. Guo and P. Xiao. (2020). Lip Reading Sentences Using Deep Learning With Only Visual Cues," in IEEE Access, vol. 8, pp. 215516-215530, doi: 10.1109/ACCESS.2020.3040906.

- Conference papers:

    1. S. Fenghour, D. Chen, L. Hajderanj, I. Weheliye and P. Xiao. (2021). "A Novel Supervised t-SNE Based Approach of Viseme Classification for Automated Lip Reading," 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), 2021, pp. 1-7, doi: 10.1109/ICECET52533.2021.9698534.

2. S. Fenghour, D. Chen, and P. Xiao. (2019). Decoder-Encoder LSTM for Lip Reading. In Proceedings of the 2019 8th International Conference on Software and Information Engineering (ICSIE '19). Association for Computing Machinery, New York, NY, USA, 162–166. DOI:https://doi.org/10.1145/3328833.3328845.†
3. S. Fenghour, D. Chen, and Perry Xiao. (2018). Contour mapping for speaker-independent lip reading system. Proc. SPIE 11041, Eleventh International Conference on Machine Vision (ICMV), 1104114; https://doi.org/10.1117/12.2522936.

† The oral presentation for this paper won "Best Presentation Award" for the Computer Vision and Deep Learning session at the 2019 International Conference on Software and Information Engineering in Cairo.

1.5 Thesis Structure

Chapter 2, *Technical Background* gives an overview of background information pertaining to lip reading and visual aspects of speech, definitions of the different nomenclature used in this thesis; and all of the different components that make up automated lip-reading systems including the audio-visual databases, feature extraction and classification networks. Some of the features discussed are critical for proposed lip-reading system in that it must not only be effective at extracting lip image features but must also be able to deal with visemes of different durations and for the learning of temporal alignment in order to decoded sequences of visemes in real-time.

Chapter 3, *Literature Review* reviews many of the latest trends in the automated lip-reading and the evolution of lip-reading systems over the last fourteen years based on the information presented in Chapter 2. A gap in the literature review is identified in that alternative classifications schemas for predicting sentence to the use of ASCII characters is not something that has widely been investigated and visemes are one of many possible classification schemas that could be implemented in a lip-reading system. Also discussed is the importance of language model in a speech recognition system for not only being able to distinguish between words that share identical lip movements but in that it can enhance the performance accuracy of lip-reading

systems using all kinds of classification schema. The final section explains the overall rational behind the proposed lip-reading system and how it addresses the research questions of this thesis.

Chapter 4, *Sentence Prediction using Visual Cues* presents a neural network-based lip reading system that is lexicon-free, uses solely visual cues and that uses visemes as a classification schema. The overall system consists of two components: an attention-based transformer for classifying visemes in continuous speech, and viseme-to-word converter that uses a pre-trained language model to predict sentences using perplexity analysis of word combinations. The overall architecture outperforms some the previous state-of-the-art approaches for decoding sentences from the benchmark sentence-based dataset BBC-LRS2. The proposed lip-reading system also addresses some of the criteria for one of research questions raised in Chapter 1 in that the system decodes words with a unique set of visemes with near perfect precision but is also still somewhat effective at discriminating between homopheme words.

Chapter 5, *Viseme-to-Word Conversion with Robustness* a presents modified approach to viseme-to-word conversion that address the limitations of the approach in Chapter 4 in that it is robustness to incorrectly classified visemes. It also addresses the limitations of the viseme-to-word converter of Chapter 4 in that it uses few parameters and therefore has less overhead and uses less time to train and execute. The viseme-to-word converter also yields improved performance in predicting spoken sentences from the BBC LRS2 dataset when integrated in the lip reading system presented in Chapter 4.

Chapter 6, *Conclusion* summarises and discusses the results and achievements of this thesis. Also addressed are the limitations of this work with a discussion about further potential research to be explored in automated lip reading that uses visual cues.

Chapter 2

Technical Background

This chapter gives an overview of the background information pertaining to lip reading and visual aspects of speech; as well as the the different components of lip reading systems that are pertinent to the lip-reading system proposed in this thesis. This includes the definitions of phonemes and visemes, the definition of a language model, metrics used to evaluate speech recognition performance in this thesis; and all of the various automated lip-reading systems including the audio-visual databases, feature extraction, classification networks and classification schemas. Several comparisons are given including a comparison of Convolutional Neural Networks with other neural network architectures for feature extraction; a review on the advantages of Attention-Transformers and Temporal Convolutional Networks to Recurrent Neural Networks for classification;

This Chapter is organized as follows: Section 2.1 is the chapter Introduction, followed by Section 2.2 which gives definitions of visemes and phonemes along with an outline of the different viseme conventions that exist, then in Section 2.3 a definition of language model is given, while Section 2.4 lists the different metrics used for evaluation the performance of lip-reading systems. The rest of the Chapter includes Section 2.5 which lists the different audio-visual databases used to train and test lip-reading systems for decoding at the character, word and sentence levels are described; Section 2.6 gives an overview of the different pre-processing aspects that make up lip-reading systems, followed by a comparison of the different frontend network architectures

used for feature extraction in Section 2.7 and a comparison of the different backend classification systems in Section 2.8.

2.1 Introduction

Most speech recognition relies on both audial and visual features and takes the form of what is known as audio-visual speech recognition. When audio is either unavailable or corrupted due to circumstances like background noise, this is when the most data that is available from the presence of visual aids [28]must be extrapolated.

Lip reading is the decoding of speech by analysing the movement of lips or visual information generated by the speaker moving their lips. Automated lip reading is also known as visual speech recognition because the automation of lip reading is principally about trying to read a person's lip speech without the assistance of audio [29].

Speech data can be decomposed into sentences, words and characters. Sentences themselves are made up of words while words are made up of characters and those characters will be in the form of either phonemes (for audio speech), visemes (for visual speech) or ASCII characters (natural language). Deep learning approaches to automated lip reading have focused on classifying words and sentences but viseme and phoneme classification can be incorporated too.

2.2 Phonemes and Visemes

The term **viseme** is generally used in machine-based lip reading to a represent a distinct lip shape that is required to generate a spoken character of the **phoneme**. The phoneme itself can be represented by an acoustic signal, however, one viseme can generate multiple phonemes which is why the mapping of visemes to phonemes represent a one-to-many relationship. According to to Hazen [30], the English language consists of roughly 40 phonemes with around a dozen distinct visemes.

The one-to-many mapping relationship between visemes and phonemes is a situation that exists not only for English speakers but is also present amongst people speaking other languages too. In fact, some languages consist of more acoustic sounds than the English language and thus contain more phonemes; meaning that there is an even greater one-to-mapping relationship between visemes and phonemes and more ambiguity for lip-readers where different sounds produce identical lip movements.

The English language has identical visemes and lip movements to other languages, but because there are phonetic sounds in other languages that are not present in English and likewise some phonetic sounds in English are not present in other languages, some other languages can contain either fewer or more visemes. However common phonemes that exist between English and other languages will map to identical visemes depending on the convention used.

Table 2.1: Phonemes and Visemes in other languages.

Language	**Phonemes**	**Visemes**	**Phoneme-to-Viseme Ratio**
Arabic	47	12	3.92
Catalan	39	16	2.44
Chinese(Madarin)	36	13	2.77
Danish	46	13	3.54
Dutch	40	13	3.08
English	47	13	3.62
French	36	13	2.77
German	48	13	3.69
Icelandic	50	14	3.57
Italian	32	13	2.46
Japanese	37	14	2.64
Korean	30	11	2.73
Norwegian	43	13	3.31
Polish	36	13	2.77
Portuguese	33	12	2.75
Romanian	31	13	2.38
Russian	45	14	3.21
Spanish	31	14	2.21
Swedish	43	16	2.69
Turkish	43	13	3.31
Welsh	49	13	3.77

Just like in English, there is no consensus on the precise number of visemes that exists in other languages. Table 2.1 gives a list of the number of phonemes and visemes that exist in

different languages and the number of visemes or phonemes listed is in accordance with Amazon Polly [26], which is a cloud service that converts text into speech. Figure 2.1 shows a bar plot of the number of phonemes used by Amazon Polly for speech in different languages.

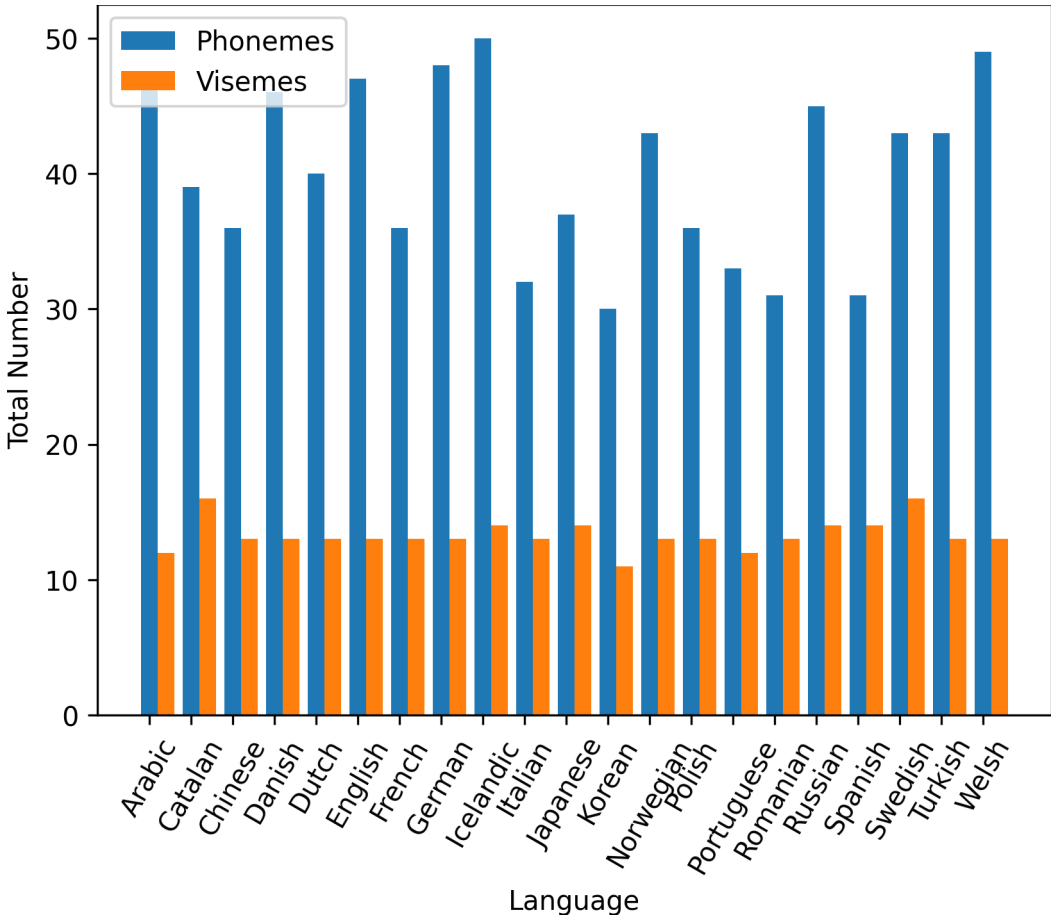


Figure 2.1: Phonemes and Visemes for different languages

There is no official standard convention for defining precise phonemes and visemes or even the number and different approaches (some of which are shown in Table 2.2) to either phoneme or visemes classification have used varying numbers of phonemes and visemes as part of their conventions with different phoneme-to-viseme mappings. They all have consonant visemes, vowel visemes and one silent viseme.

Visemes have multiple interpretations in lip-reading literature and there is no consensus on a way to define them. Two practical techniques have been used to outline visemes:

- The grouping of lip movements through the use of articulatory gestures, such as lips closing together, jaw movement and teeth exposure

- The grouping of phonemes that have the same visual appearance

The second of the two mentioned techniques, has been the most widely used for the outlining of visemes and this appears to the very reason why there are different conventions for defining visemes. Different groupings of phonemes exist and there is disagreement in the decision for choosing optimum phoneme-to-viseme mappings.

A lot of research has been carried out to compare and reconcile the rationale behind the differences such as [20] and citeTheobald. Several reasons for why such discrepancies exist, which include the difference in lip appearances across difference speakers which causes difficulty in grouping phonemes when it has to be completed across more than one speaker. This is in addition to the variation in lip-reading ability which varies across different individuals, and those with more experience are better able to identify visemes.

Most attempts at grouping together phonemes will be data-driven whereby phonemes are clustered together using statistical models. The criteria for grouping together phonemes adds another layer of discrepancy because the grouping methodologies vary between different research groups e.g. different thresholds are used.

Lee's convention appears to be the most favoured for speech classification and it is the convention that Amazon Polly themselves use [26]. In one HMM-based word classification study [31] which compares different conventions including the 5 listed in Table 2.2, Lee's convention achieves the greatest accuracy for word recognition suggesting that both the best class definitions and mappings are used. Images of lip movements corresponding to Lee's viseme convention are given in Figure 2.2 with a breakdown of the phoneme-to viseme mappings shown in Table 2.2.

Figure 2.4 gives examples of words decomposed into visemes and by analysing the distinct visemes that make up those words, they can be grouped into clusters - each representing a distinct sequence of visemes. The words "red" and "white" fall into different clusters because they have different viseme combinations however, the words "red" and "wrath" are homopheme words, that both fall into the same cluster because they have equivalent combinations of visemes according to Table 2.3.

Table 2.2: Different phoneme and viseme conventions used.

Convention	Phonemes	Total Visemes	Consonant Visemes	Vowel Visemes
Jeffers [32]	44	12	7	4
Neti [33]	43	13	8	4
Hazen [30]	53	15	9	5
Bozkurt [34]	46	16	8	7
Lee [35]	40	14	6	7

Table 2.3: Lee and Yook's viseme convention with vowels and consonants [35]

Viseme Class	Viseme Type	Phonemes Set
p	consonant	b, p, m
t	consonant	d, t, s, z, th, dh
k	consonant	g, k, n, ng, l, y, hh
ch	consonant	jh, ch, sh, zh
f	consonant	f, v
w	consonant	r, w
iy	vowel	iy, ih
ey	vowel	eh, ey, ae
aa	vowel	aa, aw, ay, ah
ah	vowel	ah
ao	vowel	ao, oy, ow
uh	vowel	uh, uw
er	vowel	er
s	silent character	sil

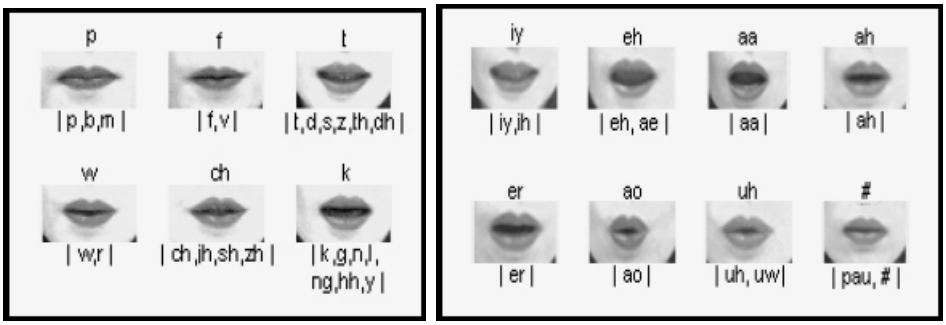


Figure 2.2: Six consonant visemes on the **left** and 7 vowel visemes plus silent viseme on the **right**. [35]

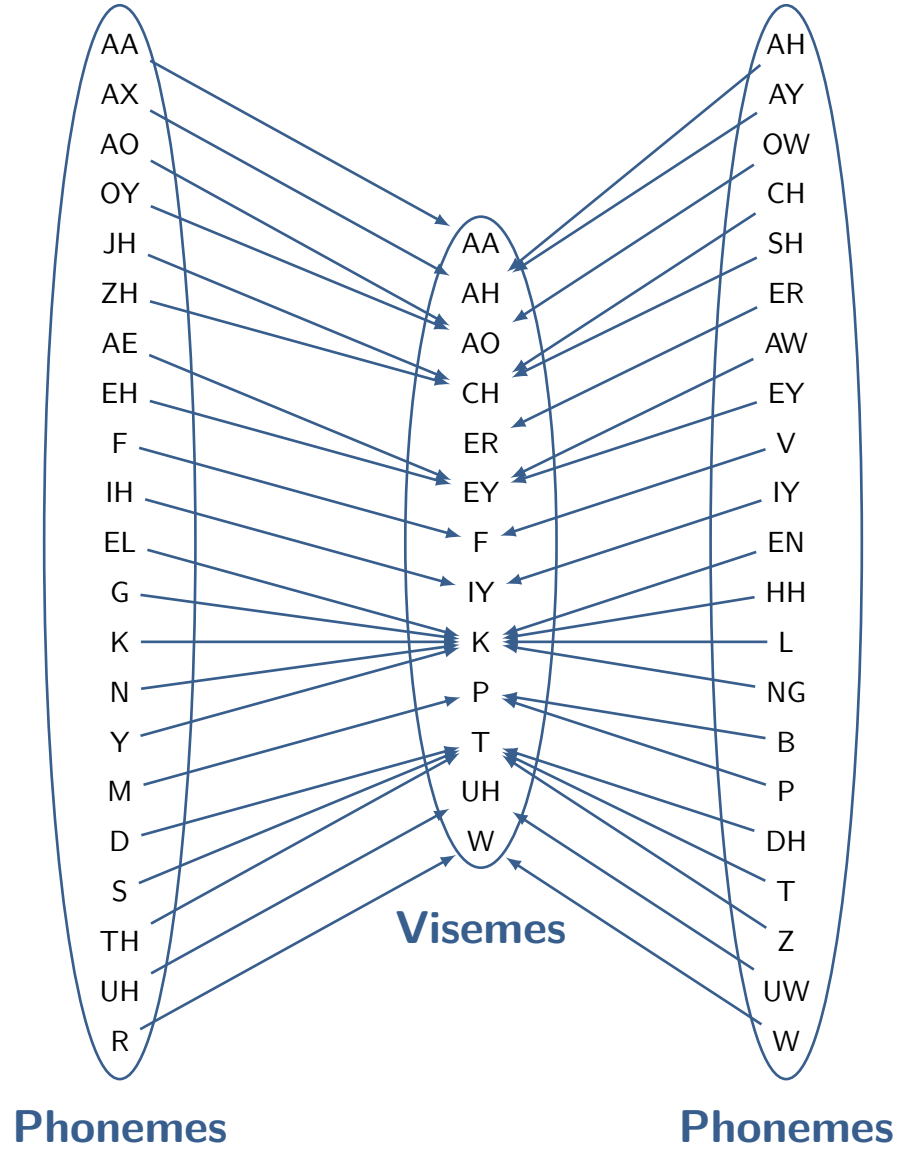


Figure 2.3: One-to-many mapping between visemes and phonemes

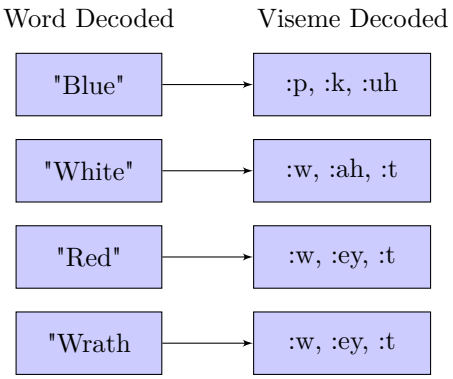


Figure 2.4: Examples of word decomposed into visemes showing the sequence of lip movements that are witnessed when uttered.

2.3 Language models

A language model is a probability distribution over sequences of words and it can be measured on the basis of the entropy of its output from the field of information theory [36].

Definition: A language model consists of a finite set of all possible words in the language V, a set of possible sentences $V^\dagger$ that could be composed with words from vocabulary V which is infinite because sentences can be of any length; and a probability distribution $p(w_1, w_2, ..., w_n)$ over sentences in $V^\dagger$ where $w_i \epsilon V$ such that [37]:

1. For any $(w_1, w_2, ..., w_n) \epsilon V^\dagger$, $p(w_1, w_2, ..., w_n) \geq 0$

2. In addition, Eq. 2.1 is satisfied.

$$\sum_{(w_1, w_2, ..., w_n) \epsilon V^\dagger} p(w_1, w_2, ..., w_n) = 1 \tag{2.1}$$

A language model provides context to distinguish between words and phrases that look similar when spoken; for example, the phrases "recognize speech" and "wreck a nice beach" both look the similar with identical lip movements when spoken. The context of a word in a language model can be deduced by its surrounding words and according to the linguist J. R. Firth: "you shall know a word by the company it keeps" [38].

2.4 Metrics for Performance Evaluation

The measures that have been used to evaluate the lip reading sentence system are edit distance-based metrics [39]. Edit distance is defined as the minimum number of character-level operations required to correct a decoded sentence to the ground truth and edit distance-based metrics are computed by calculating the normalized edit distance between the ground truth and a predicted sentence. Metrics reported in this thesis include VER, CER, WER and SAR.

Error rate metrics used for evaluating accuracy are given by calculating the overall edit distance. In determining misclassifications, one has to compare the decoded speech to the actual speech.

The equation for calculating Error Rate (ER) is given in Eq. 2.2 with N being the total number of characters in the ground truth, S being the number of characters substituted for wrong classifications, I being the number of characters inserted for those not picked up and D being the number of deletions being made for decoded characters that should not be present. CER, WER and VER are all calculated this way with the expressions given in Eqs. 2.3, 2.4 and 2.5 where C, W and V correspond to characters, words and visemes.

$$ER = \frac{S + D + I}{N} \tag{2.2}$$

$$CER = \frac{C_S + C_D + C_I}{C_N} \tag{2.3}$$

$$WER = \frac{W_S + W_D + W_I}{W_N} \tag{2.4}$$

$$VER = \frac{V_S + V_D + V_I}{V_N} \tag{2.5}$$

SAR is a binary metric as expressed in Eq. 2.6, where the value is 1 if the predicted sentence P_P is equal to the ground truth P_T, otherwise it would take the value of 0:

$$SAR = \begin{cases} 1, & P_P = P_T \\ 0, & P_P \neq P_T \end{cases} \tag{2.6}$$

Tables 2.4 and 2.5 give examples of how the character and word accuracies can be calculated. If we take the first pair of phrases in Table 2.4, 3 character substitutions are required to modify the phrase in Case 1 to make it identical to Case 2 whereby we would literally change "in o" to "at l". Meanwhile, "bin blue a x e again" requires a total of 6 changes including 1 substitution and 5 deletions for "a x e" to be modified to "at s three" and "lay white at e zero please" requires 7 changes including 5 substitutions and 2 deletions for "white at" to be modified to become "red in". Table 2.5 shows how word error rates would be calculated where all of phrases listed would involve direct word substitutions.

Table 2.4: Character error rates calculations for different phrases.

Case 1	Case 2	S	D	I	N	CAR(%)
bin blue in o six now	bin blue at l six now	3	0	0	3	85.8
bin blue in x one soon	bin blue at s one soon	3	0	0	3	86.4
bin blue a x e again	bin blue at s three again	1	0	5	6	76.0
lay white at e zero please	lay red in e zero please	5	2	0	7	70.8

Table 2.5: Word error rates calculations for different phrases.

Case 1	Case 2	S	D	I	N	WAR(%)
bin blue in o six now	bin blue at l six now	2	0	0	6	66.7
bin blue in x one soon	bin blue at s one soon	2	0	0	6	66.7
bin blue a x e again	bin blue at s three again	3	0	0	6	50.0
lay white at e zero please	lay red in e zero please	2	0	0	6	66.7

2.5 Datasets

As a data-driven process, the design and development of lip-reading systems has been inevitably affected by available data. Ideally, the data should be vocabulary rich, with variations in pose and illumination. Large data corpuses such as BBC-LRS2 [17], LRS3-TED [40], LSVSR [41] have been compiled from hours of programmes that have been streamed on the BBC, TED-X and YouTube. These corpuses consist of thousands of videos of people uttering sentences with thousands of different words. These datasets also consist of people speaking at different angles with varying levels of illumination.

Table 2.6 lists some of the main audio-visual datasets that have been utilized for lip-reading over the last thirty years. This is a table that has been put together as part of this research to highlight how lip-reading corpuses have matured. The first lip-reading datasets to be constructed were designed for classifying isolated speech segments in the form of digits and letters, with more recent datasets consisting of videos designed to classify longer segments in the form of words. Moreover, the most up-to-date lip-reading datasets consist not only of longer speech segments, but segments in continuous speech as opposed to isolated speech to better model visual speech in real time.

A further development of lip-reading data corpuses in addition to the nature of speech segments themselves is the ability to train lip-reading systems to classify speech from people speaking

at various different angles(profile views), as opposed to frontally facing the cameras(frontal views). Additionally datasets such as LRW [42], LRS2 and LRS3 have moved on to gathering videos from multiple speakers as opposed to individual speakers, as one of the challenges facing the success of automated lip-reading systems is the inability to generalize to different people - especially unseen speakers who have not appeared in the training phase.

Other trends in the evolution of audio-visual corpuses include varying resolutions to accommodate for the fact that in real time, a person will often be speaking at varying distances from a video camera. There have also been varying frame rates to accommodate for videos that are sampled at different frequencies as well having to contend with the possibility of there not being enough temporal information available due to the nature of videos having a low sampling frequency. The majority of corpuses uses the English language due to English being the world's lingua franca, though there are datasets that utilize other languages.

Table 2.6: Available audio-visual datasets. **I** stands for Isolated (one speech segment per recording) and **C** stands for Continuous recording.

Dataset	**Language**	**Year**	**I/C**	**Segment**	**Speakers**	**Classes**	**Utterances**	**Resolution**	**Frame rate (fps)**	**Pose°**
AGH AV [43]	Polish	2012	I	Digits	20	10	N/A	1920×1080	50	Frontal
AusTalk [44]	English	2014	I	Digits	1000	10	24000	640×480	-	Frontal
AusTalk [44]	English	2014	I	Words	1000	996	996000	640×480	-	Frontal
AusTalk [44]	English	2014	I	Sentences	1000	59	59000	640×480	-	Frontal
AV Digits [45]	English	2018	I	Digits	53	10	795	1280×780	30	0,45,90
AV Digits [45]	English	2018	I	Phrases	39	10	5850	1280×780	30	0,45,90
AV@CAR [45]	Spanish	2004	I	Alphabet	20	26	800	768×576	25	Frontal
AV@CAR [46]	Spanish	2004	I	Digits	20	10	600	768×576	25	Frontal
AV@CAR [46]	Spanish	2004	I	Sentences	20	250	6000	768×576	25	Frontal
AVAS [47]	Arabic	2013	I	Digits	50	10		640×480	30	-90,-45,0,45,90
AVAS [47]	Arabic	2013	I	Words		24	13850	640×480	30	-90,-45,0,45,90
AVAS [47]	Arabic	2013	I	Phrases		13		640×480	30	-90,-45,0,45,90
AVICAR [48]	English	2004	C	Alphabet	86	26	59000	720×480	30	4 views
AVICAR [48]	English	2005	C	Digits	86	10		720×480	30	4 views
AVICAR [48]	English	2006	I	Sentences	86	20		720×480	30	4 views

Continued on next page

Table 2.6 – continued from previous page

Dataset	Language	Year	I/C	Segment	Speakers	Classes	Utterances	Resolution	Frame rate (fps)	Pose°
AVLetters [49]	English	1998	I	Alphabet	10	26	780	376×288	25	Frontal
AVLetters2 [50]	English	2008	I	Alphabet	5	29	910	1920×1080	50	Frontal
AVSD [51]	Arabic	2019	I	Phrases	22	10	1100	1920×1080	30	Frontal
AV-TIMIT [30]	English	2004	I	Sentences	233	510	4660	720×480	30	Frontal
BANCA [52]	Multiple	2003	I	Digits	208	10	29952	720×576	25	Frontal
BL [53]	French	2011	I	Sentences	17	238	4046	640×480	30	0,90
CAVSR1.0 [54]	Chinese	2000	I	Words	20	78	3120	352×228	25	Frontal
CENSREC-1-AV [55]	Japanese	2010	C	Digits	42	10	3234	720×480	30	Frontal
CMU AVPFV [56]	English	2007	I	Words	10	150	15000	640×480	30	0,90
CUAVE [57]	English	2004	I	Digits	36	10	7000	720×480	30	-90,0,90
DAVID [58]	English	1996	I	Words	123			640×480	30	Frontal
GRID [59]	English	2006	I	Phrases	34	34000	34000	720×576	25	Frontal
GRID-Lombard [60]	English	2018	I	Phrases	54	5400	5400	720×480(face) 864×480(side)	24	0,90
HAVRUS [61]	Russian	2016	I	Sentences	20	1530	4000	640×460	200	Frontal
HIT-AVDB-II [62]	Multiple	2008	I	Sentences	30	11	1980	720×576	25	0,30,60,90
IBM AV-ASR [63]	English	2015	I	Sentences	262	10400	N/A	704×480	30	Frontal
IBMIH [64]	English	2004	C	Digits	79	10	16197	720×480	30	Frontal
IBMSR [65]	English	2008	C	Digits	38	10	1661	368×240	30	-90,0,90
IBMViaVoice [33]	English	2000	I	Sentences	290	10500	24325	704×480	30	Frontal
IV2 [66]	French	2008	I	Sentences	300	15	4500	780×576	25	0,90
LiLiR [67]	English	2010	I	Sentences	12	200	2400	720×576	25	0,30,45,60,90
LRS2 [17]	English	2017	I	Sentences	>1000	17428	118116	160×160	25	-30 ~30
LRS3 [40]	English	2018	I	Sentences	>1000	70000	165000	224×224	25	-90 ~90
LRW [42]	English	2016	C	Words	>1000	500	400000	256×256	25	-30 ~30
LRW-1000 [68]	English	2018	C	Words	>2000	1000	718018	Distributed	25	-90 ~90
LSVSR [41]	English	2014	I	Sentences	>1000	127055	2934899	128×128	23-30	-30 ~30
LTS5 [69]	French	2011	I	Digits	20	10	180	1920x1080	25	0,30,60,90
M2VTS [70]	English	1997	C	Digits	37	10	2920	286×350	25	Frontal

Continued on next page

Table 2.6 – continued from previous page

Dataset	Language	Year	I/C	Segment	Speakers	Classes	Utterances	Resolution	Frame rate (fps)	Pose°
MIRACL-VC [71]	English	2011	I	Words	15	10	1500	640×480	15	Frontal
MIRACL-VC [71]	English	2012	I	Phrases	15	10	1500	640×480	15	Frontal
MOBIO [72]	English	2017	I	Sentences	150	N/A	N/A	640×480	16	Frontal
MODALITY [73]	English	2015	I	Words	35	182	231	1920×1080	100	Frontal
MV-LRS [74]	English	2009	I	Sentences	>1000	14960	74564	160×160	25	0 ~90
NDUTAVSC [75]	German	2010	I	Digits	66	10	6907	640×480	100	Frontal
NDUTAVSC [75]	German	2010	I	Words		6907	6907	640×480	100	Frontal
NDUTAVSC [75]	German	2010	I	Sentences				640×480	100	Frontal
OuluVS [76]	English	2015	I	Sentences	20	10	1000	720×576	25	Frontal
OuluVS2 [77]	English	2010	C	Digits	53	10	159	1920×1080	30	0,30,45,60,90
OuluVS2 [77]	English	2010	I	Phrases	53	10	1590	1920×1080	30	0,30,45,60,90
OuluVS2 [77]	English	2010	I	Sentences	53	540	2120	1920×1080	30	0,30,45,60,90
QuLips [78]	English	2015	I	Digits	2	10	3600	720×576	25	-90 ~90
RM-3000 [79]	English	2015	I	Sentences	1	1000	3000	360×640	60	Frontal
TCD-TIMIT [80]	English	2011	I	Sentences	20	62	5954	1920×1080	30	0,30
TULIPS1 [81]	English	1995	I	Digits	12	4	96	100×75	30	Frontal
UNMC-VIER [82]	English	2002	I	Sentences	123	12	2460	708×640	29	0, 90
UWB-05-HSAVC [83]	Czech	2005	I	Sentences	100	200	20000	720 × 576	25	Frontal
UWB-07-ICAV [84]	Czech	2008	I	Sentences	50	7550	10000	720 × 576	<50	Frontal
VALID [85]	English	2005	I	Digits	106	10	590	576×720	25	Frontal
VIDTIMIT [86]	English	2010	I	Sentences	34	346	430	512×384	25	Frontal
VLRF [87]	Spanish	2017	I	Sentences	24	1374	10200	1280×720	50	Frontal
WAPUSK20 [88]	English	1999	I	Sentences	20	52	2000	640×480	32	Frontal
XM2VTS [89]	English	1999	C	Digits	295	10	1064	720×576	25	Frontal
AV Digits [45]	English	2020	C	Digits	6	10	6000	1920×1080	25	Frontal

Continued on next page

Table 2.6 – continued from previous page

Dataset	Language	Year	I/C	Segment	Speakers	Classes	Utterances	Resolution	Frame rate (fps)	Pose°
NSTDB [90]	Chinese	2020	C	Words	N/A	349	N/A	64×64	25	-90 ~90

2.5.1 Letter and Digit Recognition

Because research in automated lip-reading started with simplest cases possible before gradually evolving to be suited to lip-reading natural spoken language in real time, the first databases that were available for lip-reading were designed for the task of recognizing English letters and digits.

The AVLetters [49] dataset consists of 10 speakers (5 males and 5 females) uttering isolated letters from A to Z. Each letter was repeated three times by the speaker, and videos were recorded at a rate of 25 frames per second(fps) at an audio sampling rate of 22.5 kHz. A higher definition edition of the AVLetters database named AVLetters2 [50] was later compiled; and it includes 5 speakers uttering 26 isolated letters seven times with videos sampled at 50 fps, with an audio sampling rate of 48 kHz.

The AVICAR [48] dataset was recorded in a moving car with four cameras deployed on the dashboard for recording videos. The dataset consists of 100 speakers (50 males and 50 females) with 86 of them available for downloading. Each speaker was asked to first speak isolated digits and then letters twice, followed by 20 phone numbers with 10 digits each. Videos have a visual frame rate of 30 fps and an audio sampling rate of 16 kHz.

Tulips [81] which was released in 1995 is one of the oldest databases constructed for digit recognition. It consists of 96 grayscale image sequences pertaining to 12 speakers (9 males and 3 females) each uttering the first four English digits twice. Videos were sampled at 30 fps with resolution 100×75 pixels and the images contain only the mouth region of the speakers.

The M2VTS database [70] contains videos of 37 people (25 men and 12 women) uttering consecutive French numerals from 0-9, which were repeated five times by each person. The XM2VTSDB database [89] is an extension of the M2VTS database, and was constructed by getting 295 people to utter digits 0-9 in different orders. The VALID [85] database was designed to test a lip-reading system's robustness to light and noise conditions which is why the videos contain illumination, background and noise variations. Altogether, it contains 530 videos with 106 speakers speaking in five different environments.

AVDigits [91] is one of the largest datasets available for digit classification. It contains videos recorded with normal, whispered and silent speech and in it; participants read out 10 digits, from 0 to 9 in a random order five times in the three different modes of speech. They spoke at normal volume for the mode of normal speech, whispered for the whispering mode and remained silent in silent speech mode. 53 participants were recorded in total.

The CUAVE [57] (Clemson University Audio-Visual Experiments) database includes speaker movement and simultaneous speech from multiple speakers. It is split into two major sections: the first consists of individual speakers and the second consists of pairs of speakers. For the first section, 36 speakers (17 males and 19 females) were recorded with each speaker uttering 50 isolated digits while facing the front; another 30 isolated digits while moving the head and after that, the speaker was recorded from both profile views while speaking 20 isolated digits. Each individual then uttered 60 connected digits while facing the camera again. Videos were recorded at 30 fps with an audio sampling rate of 16 kHz.

Other corpuses constructed for digit recognition in speech recognition include AV@CAR [46] for Spanish digits, CENSREC-1-AV [55] for Japanese, NDUTAVSC [75] for German; LTS5 [69] databases for French, AGH AV [43] for Polish as well as other English datasets like IBMIH [64], IBMSR [65] and QuLips [78].

2.5.2 Word and Sentence Recognition

The focus of compiling datasets for letter and digit recognition initially was not motivated solely by starting with simplest cases possible, but also due to the simplicity in the gathering of such data. Later, researchers focused more on the the task of predicting words, phrases and sentences in continuous speech whereby they had to overcome the problem of trying to identify different words that look or sound identical when spoken.

The MIRACL-VC1 [71] database was released in 2014. It consists of videos from 15 participants who each uttered one of 10 possible words ten times, resulting in the availability of 1500 word videos. Videos were recorded using an RGBD camera with resolution 640×480 pixels and a frame rate of 15 fps. The videos were sampled into image frames with the images being divided into colour pictures and depth pictures - the latter of which contained more depth information.

Other isolated word datasets for the English language include MODALITY [73], AusTalk [44], CMU AVPFV [56] and DAVID [58]. Corpuses for other languages include AVAS [47] for Arabic, CAVSR1.0 [54] for Chinese and NDUTAVSC [75] for German.

Meanwhile, possibly the one of largest English word datasets we have available to us today, LRW [1] contains 1000 utterances of 500 different words, spoken by over 1000 different speakers. Videos were extracted from a number of BBC television programmes streamed between 2010 and 2016, and they are 1.16s long with a frame rate of 50 fps without any audio.

LRW-1000 [68] is possibly one of the largest continuous audio-visual datasets for words altogether consisting of over 700,000 samples of 1000 Chinese words spoken by over 2000 different speakers from Chinese CCTV programs. This dataset is unique in that it consists of videos with varying resolutions which makes it useful for the natural variability of people speaking in real-time where you will either have people speaking at varying distances from a video camera or videos that have been recorded with varying spatial dimensions.

The XM2VTSDB [89] corpus which consists of 295 speakers uttering digits, also consists of videos with the 295 speakers pronouncing the sentence "Joe too parents green shoe bench out". This makes it one of the oldest sentence-based corpuses. The MIRACL-VC1 [71] dataset in

addition to having compiled word video data, also consists of sentence videos whereby each of the 10 speakers uttered one of ten phrases ten times to generate 1500 phrase videos.

IBMViaVoice is one of the largest datasets available for lip-reading sentences and it contains videos with 290 speakers speaking a total of 24325 sentences with 10500 different words being spoken. It is however unavailable to the public.

The OuluVS1 [76] database consists of 10 phrases spoken by 20 speakers(17 males and 3 females), with each utterance repeated by the speaker up to nine times. Videos were recorded at 25 fps with an audio sampling rate of 48kHz. The OuluVS2 [77] database is an extension of OuluVS1 which also contains videos of these 10 phrases but spoken by with 52 different speakers.

The GRID [59] corpus consists of 34 speakers(18 males and 16 females) who each utter 1000 sentences [59] that follow a standard pattern of verbs, colours, prepositions, alphabet, digits, and adverbs [59]. "Set white with p two soon" is an example of one spoken sentence and each video has a duration of 3 seconds with a sampling rate of 25 fps and audio 25kHz.

The GRID-Lombard [60] database is an extension of the GRID corpus and consists of 54 speakers(30 females and 24 males) who altogether pronounce 5400 sentences that follow the GRID convention and take the form of "<verb>, <colour>, <preposition>, <letter>, <number>, <adverb>" using combinations that do not appear in the GRID corpus. It should be noted that the emphasis of this corpus is to not only include profile views of people speaking in addition to frontal views but to also provide videos of people speaking according to Lombard speech so that the Lombard effect can be modelled. The Lombard effect is the spontaneous habit of a speaker to increase their vocal effort when speaking in loud noise to enhance the audibility of their voice [92].

The TIMIT corpus is a dataset with audio recordings of 630 speakers each speaking 10 different sentences to give a total of 6300 sentences [93]. Several datasets with people uttering sentences following the TIMIT structure have been constructed.

The AV-TIMIT [30] database was constructed for performing speaker-independent audio-visual

speech recognition and the corpus contains videos of 233 speakers (117 males and 106 females) uttering TIMIT sentences [93]. Each speaker was asked to utter 20 sentences, and each sentence was spoken by at least 9 different speakers with one sentence that was uttered by all the speakers. Videos were recorded at 30 fps with a resolution of 720×480 pixels and an audio sampling rate of 16 kHz.

Similarly, the Vid-TIMIT [86] database is comprised of videos of 43 speakers (19 females and 24 males), each pronouncing 10 different TIMIT sentences. The videos were recorded at 25 fps with resolution 512×384 pixels and an audio-sampling rate of 32kHz. Meanwhile, the TCD-TIMIT [80] database consists of videos of resolution 1920×1080 pixels from 62 female speakers of whom 3 are professional lip readers and the other 59 are volunteers. The three professionals say 377 sentences each while the remaining speakers speak 98 sentences each.

In recent years, more challenging datasets consisting of spoken sentences that are more random and less structured have been constructed which consist of thousands of sentences spoken by limitless people, with extensive vocabularies covering thousands of different possible words so that lip-reading systems can be generalised to natural spoken language. The LRS2 [1] dataset is a sentence-based dataset of videos without audio which was compiled by extracting videos from BBC television programmes much like the LRW corpus. Altogether the corpus covers 17,428 different words with a total of 118,116 samples.

MV-LRS [74] is also a sentence-based dataset constructed from videos from BBC programs with a total of 74,564 samples covering 14,960 words. However, unlike the LRS2 corpus which only includes frontal shots, MV-LRS includes both profile and frontal shots.

The LRS3-TED [94] dataset is another sentence-based dataset compiled in a similar fashion by extracting videos from Ted-X videos where 150,000 sentences were extracted from TED programs. LSVSR [95] was built using YouTube videos with 140,000 hours of audio, approximately 3,000,000 speech utterances and over 127,000 words making it the largest database to date.

Lip-reading datasets with people pronouncing sentences in other languages have also been created too. Examples include AV@CAR [46] and VLRF for Spanish, AVAS [47] and AVSD [51]

for Arabic, BL [53] and IV2 [66] for French, UWB-05-HSAVC [83], and UWB-07-ICAV [84] for Czech, the German NDUTAVSC [75] dataset, the Russian HAVRUS [61] corpus and the HIT-AVDB-II [62] database that covers Chinese and English.

2.5.3 Multiview Databases

In an ideal situation, an automated lip-reading would only need videos of people speaking from frontal poses. However, in practice it is impossible to always guarantee that the input images will be exclusively from frontal shots.

Another challenge with pose is when a video with a talking person consists of that very person speaking at different angles. When there is a static camera, a speaker may rotate their face while speaking which results in the data that is present consisting of a person speaking at multiple angles in the very same video. Some datasets provide image data recorded at various angles whilst a speaker is speaking, though this is not always the case.

Many researchers argue that the frontal shots are not necessarily the best angles to use for lip-reading. One reason for this is that a slight angle deviation can be beneficial because lip-protrusion and the rounding of the lip can be better observed [96] [97].

2.6 Preprocessing

One of the stages of automated lip-reading is to extract the region of interest and in the case of automated lip-reading, the ROI that needs to be extracted is the person's lips. The lip movements will be given a speech class label according to the hierarchy of speech data explained in the Introductory Chapter.

There are different feature representation methods that can be used to represent lip movements and they can typically be divided into four categories as summarized by Dupont and Luettin [98]: geometric-based, image-based, model-based and motion-based. A more detailed comparison of feature representation can be found in the following works [99] [98].

The overwhelming majority of deep learning classification methods use image-based feature representation and the input will either be an image with channels of red, green and blue pixel intensities or an input with grayscale images. A general advantage of being able to use raw pixel data as a neural network input is that there is less pre-processing involved as there is no need to device hand-crafted models for extracting facial contours or the representations of lip motion.

For a recorded video of a person speaking, an automated lip-reading system will first need to sample the video into image frames. Once the video has been sampled, the face must be detected as part of a face localization step which involves facial landmarks needing to be located in order to extract just the speaking person's lips as the ROI and feature input to the visual frontend. Figure 2.5 outlines the process of extracting the ROI of an individual speaking in a video, while Figure 2.6 shows an example of an image frame and its corresponding ROI.

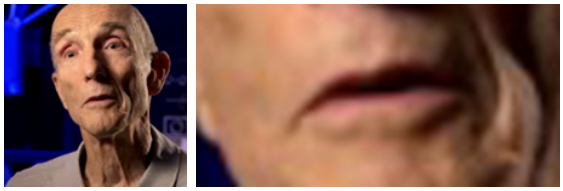


Figure 2.5: A person's face on the **left** with the extracted ROI shown on the **right**.

A variety of face localization methods can be used for extracting facial landmarks from people's faces and such approaches include Naive Bayes classifiers [100], neural networks [101], HMMs [102] and Principal Component Analysis [103] to name a few. A more detailed review of face localization procedures can be found in [102], though they all typically use the standard iBug landmark convention where 68 landmarks are detected for the face. The procedure for locating facial landmarks and to extract the ROI is shown in Figure 2.7.

For the first deep learning-based lip reading systems, the ROI extraction was often performed as part of preprocessing, but modern end-to-end lip reading systems now perform ROI extraction during the feature extraction stages whereby a frontend will have been trained to locate the ROI and this means that video frames do not need cropping beforehand [104] [105].

After locating and extracting the ROI, a series of pre-processing steps will typically be applied to the image and this is done to not only improve the efficiency of training and validation by

reducing the number of overall operations but also to limit variation as much as possible. Pre-processing will often consists of processes such as grayscale conversion, z-score normalization and some augmentation techniques; though augmentation is implemented during the training phase.

Images naturally consist of three pixel channels in the red-green-blue(RGB) format with red, green and blue pixel components. The challenge with images having multiple colour channels is that there will be huge volumes of data to work with, making the process computationally intensive. So as a result, lip-reading systems will often consist of a grayscale conversion stage where RGB pixels are converted to a grayscale format beforehand.

Another pre-processing step is the Normalization process. Normalizing helps to ensure consistency of scale when processing images, which can improve a model's ability to learn if the scales for different features are very different. Z-score normalization is the simplest of such techniques where a correction is applied to all of the pixels by subtracting from every pixel x the mean pixel value $\bar{x}$ and then dividing by the standard deviation σ to give a corrected pixel value x' with zero-mean and unit-variance according to Eq. 2.7.

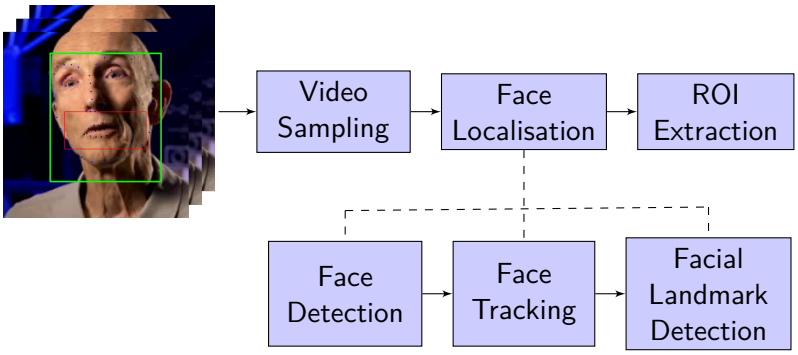


Figure 2.6: Procedure for video processing.

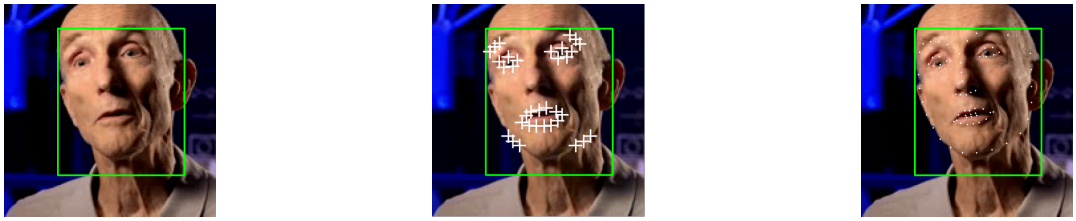


Figure 2.7: Stages of facial landmark extraction including face detection(**left**), face tracking(**middle**) and facial landmark detection(**right**).

$$x' = \frac{x - \bar{x}}{\sigma} \tag{2.7}$$

In summary, the training of a good classification model for speech recognition requires a lot of data and the lack of the labelled training data leads to poor generalization. A greater availability of training data will invariably lead to a better classification model. However, when there is an insufficient supply of data available to begin with, augmentation can be a useful strategy which is where existing training data is extended by adding modified or augmented samples. New training samples can be created by applying various transformations to existing labelled samples. Examples of image-based augmentation techniques include rotation, scaling, flipping, cropping, spatial or temporal pixel translation and even the addition of Gaussian noise.

2.7 Feature Extraction

Feature extraction for visual speech recognition has two main purposes. The first is to separate redundant features in the images from relevant features and the second is to convert images from high-dimensional space into low-dimensional space. A variety of techniques such as Active Appearance Models, Active Shape Models, Discrete Cosine Transformation, Linear Discriminant Analysis, Principal Component Analysis and Locality Discriminant Graphs have been deployed for feature extraction in lip-reading and more detailed information about such approaches can be found in Zhou's work [99]. Non-deep learning methods of feature extraction will not be discussed in this Section. For most of the up-to-date state-of-the-art lip-reading systems, deep learning methods are preferred to traditional methods because feature extraction can be automated.

Convolutional Neural Networks are one family of neural networks that have been deployed for feature extraction in neural network architectures for automated lip-reading. They are a supervised learning method and they account for majority of networks used for feature extraction. The other family of architectures used for feature extraction include Autoencoders, Restricted Boltzmann Machines and Deep Belief Networks which are all unsupervised methods mainly

used in dimensionality reduction tasks.

2.7.1 Multilayer Perceptrons

A multilayer perceptron or a multilayer feed-forward neural network is the most basic neural network that can be used for feature extraction. Wand et al. used a multi-layer feed-forward network as part of a frontend for three of their approaches where 51 different possible variants of words from the GRID corpus were decoded with an LSTM configuration used in the backend. A 40×40 pixel window containing the lips was extracted from each video frame before being converted to grayscale and flattened into a 1D vector. This was performed for every frame that made up the video and so videos were inputted into the frontend in the form of 2D matrices.

Multilayer perceptrons are limited in comparison to other architectures that can be used for feature extraction including Autoencoders and CNNs because image frame pixels from videos have to be stacked together. This means that feed-forward neural networks simply compress image data without being able to learn the spatial and temporal features needed for processing sequential inputs.

2.7.2 Autoencoders and RBMs

An Autoencoder is a network used for learning compressed distributions of data. Autoencoders consist of an encoder and decoder. The encoder converts data in higher-dimensional space to lower-dimensional space, while the decoder transforms the lower-dimensional data into higher-dimensional data. For input data x, the autoencoder tries learning identity relationship $x_{out} = x$ by tuning the network weights and biases when the network is being trained. The loss function is simply a normalisation of the difference between x_{out} and x which the network tries to minimise. The operations performed by the encoder and a decoder are given in Eqs. 2.8 to 2.11 respectively. W is the encoder weight matrix, b is the encoder bias matrix, W^T is the decoder weight matrix, and b' is encoder bias matrix [106].

$$Encoder(x) = Wx + b \tag{2.8}$$

$$Decoder(x) = W^T x + b' \tag{2.9}$$

$$Loss = min(f_{loss} : W^T(Wx + b) + b', x) \tag{2.10}$$

$$C_{AE} = W_{AE} I + b \tag{2.11}$$

The Decoder section of the Autoencoder is only used for training and discarded for validation as it the compressed representation learned by the Encoder that is used for feature extraction in lip-reading [106].

Real Boltzmann Machines have an identical structure to Autoencoders, but they differ in that they use stochastic units with a particular distribution(usually Binary of Gaussian) instead of deterministic distribution. The learning procedure consists of several steps of Gibbs sampling where the weights are adjusted to minimize the loss function [106].

Petridis et al. proposed lip-reading systems in a number of works that use bottleneck RBMs to do the feature extraction for lip-reading sentences. Their work in [107] decoded phrases from the OuluVS2 using an LSTM backend with two visual input streams. The first input stream uses inputs of 2D image frames converted into grayscale, while the second stream uses the difference between two consecutive frames as the input. For the outputs of both bottlenecks, the first and second derivatives are processed and added to the bottleneck outputs. Each overall output stream is then is fed into an LSTM layer with both LSTM outputs then concatenated and passed into a Bidirectional LSTM with their information combined. The output layer is a softmax layer that performs the classification.

Petridis et al.'s architecture in [108] is similar to that of [107] except the second input stream takes audio as an input as opposed to taking in the differential of two consecutive images frames, as well using bidirectional LSTMs instead of unidirectional LSTMs. Petridis et al. [109] presented a third system for tackling multi-view lip-reading for sentence prediction. There are three architecturally identical streams to extract features from three images captured from

different angles. The outputs are concatenated and passed into a Bidirectional LSTM and a softmax layer that perform classification in an identical manner to [107] and [108]. Meanwhile, Petridis et al.'s fourth proposed architecture [91] is similar to [108] except that the system uses only visual inputs with no audio for assistance.

Autoencoders and RBMs do have advantages over CNNs; one is that they are unsupervised learning techniques and can map data from higher dimensions to lower dimensions in isolation without the need for any labelled classification. They also have simpler topologies to tune and are quicker and more compact for backpropagation [110].

Autoencoders and RBMs do have limitations in their feature extraction capabilities. Whilst Autoencoder or RBMs try to capture as much information as possible, they can be inefficient if information that is most relevant the classifier makes up only a small part of the input and so an autoencoder or RBM may lose a lot of it. CNNs are better at separating relevant information from redundant information [110].

2.7.3 2D CNNs

It is common to have a series of 2D CNN kernels whereby feature extraction is performed for each individual image frame. A CNN will extract features using architectural layers for convolution, pooling and normalization; and for a 2D CNN, the convolution stage involves convolving an input y with a weight ω of kernel width k_w and height k_h over the different channels and over a coordinate system where i and j are spatial coordinates such that y and ω ϵ $\mathbb{R}^{C\times k_w\times k_h}$. For the expression shown in Eq. 2.12, C represents the different channels for the image. There will be three channels for RGB pixels and 1 channel for grayscale pixels and the convolution may consist of an arbitrary bias b.

Videos of people's lips moving are sampled into image frames and there are three types of set-up used for visual frontends when extracting features from lip images:

1. A Series of 2D CNN kernels
2. Concatenation of 2D image frames

3. Differential between image frames

The most common set-up to use in a 2D CNN frontend is to extract features for individual image frames using a CNN kernel for each image within the sampled video(Figure 2.8) such that the output of the frontend will be a time series of image frames represented in a lower dimensional space [111].

$$(y \otimes w)_{ij} = \sum_{c=1}^{C} \sum_{i'=1}^{k_w} \sum_{j'=1}^{k_h} y_{ci'j'} w_{c,i'+i,j'+j} \tag{2.12}$$

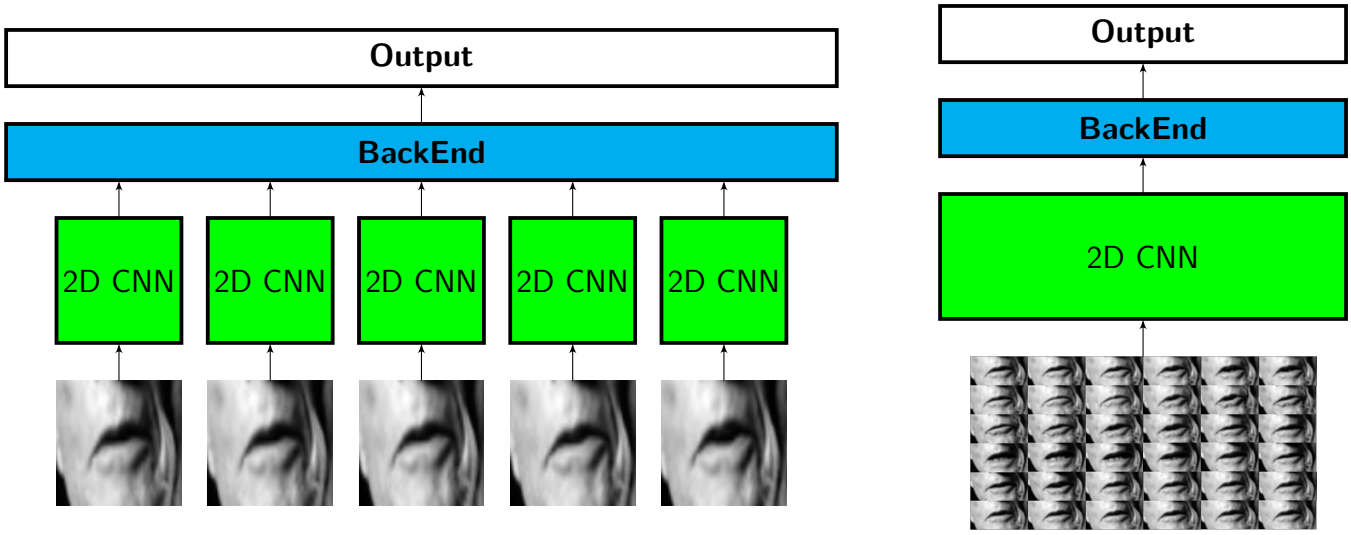


Figure 2.8: CNN diagrams with 2D kernel CNN shown on the **left** and Concatenated Image Frame CNN on the **right**.

Noda et al. [14] were among the first group to use CNNs for lip-reading in a task of extracting visual feature sequences for 6 people speaking 300 Japanese words whereby the output formed the input of a Gaussian Mixture Model-Hidden Markov Model(GMM-HMM) used for classification. Their results demonstrated that the visual features acquired by CNNs were significantly better than those acquired using traditional methods like PCAs. They later proposed a lip-reading system that incorporated audio as an input for assistance to create an audio-visual speech recognition system.

Chung and Zisserman proposed SyncNet [112], a CNN consisting of 5 convolution layers and 5 fully-connected layers. Grayscale images are the input, with a feature vector as the frontend output. The output of each CNN kernel is then concatenated and inputted into a single LSTM

and their overall model performs the classification of phrases from the OuluVS dataset. The LSTM processes the feature vector as a temporal sequence and with a Softmax layer, a class is predicted. They repeat the same task using almost the same architecture except with a VGG-M topology for the CNN kernels that was already pre-trained in ImageNet with its weights being frozen for training as opposed to the SyncNet. An accuracy rate for validation of the initial SyncNet model of 92.8% was recorded in comparison to a validation accuracy rate of just 25.4% and the main reason for the former model performing significantly better was that the SyncNet kernels were trained directly on the lip-reading data as opposed to the VGG-M kernels which were not.

Chung and Zisserman [42] used VGG-based CNNs for feature extraction when lip-reading words in continuous speech from the LRW dataset. They proposed two different structures including Early Fusion(EF) and Multiple Tower (MT), which both concatenate the outputs of the different CNN kernel streams at different stages. The EF model involves applying 2D CNN kernels to every grayscale ROI and concatenating the outputs before applying convolution layers and pooling layers. Whereas the MT model uses extracted ROIs with RGB pixels and applies one stage of convolution and pooling to the outputs of every stream individually before concatenating the streams. Performance results indicated that the MT model performed the best.

Other examples of lip-reading visual frontends that use series of 2D CNN kernels include Lee et al. [113]who devised a multi-view lip-reading system and experimented with three scenarios: single-view, cross-view, and multiple-view; Lu and Li [45] who introduced a hybrid neural network architecture composed of a VGG CNNs to lip image features from people uttering digits from 0 to 9; and Zhang et al. [114] proposed a visual speech recognition system called LipCH-Net using VGG-M kernels for lip-reading Chinese sentences. Finally, Lu et al. [115] constructed a lip-reading system for hearing impaired individuals and dysphonic people that combined lip-reading with sign language where one of the inputs streams used image frames of lip movements and the other input stream who used image frames of hand gestures.

One other set-up used for 2D CNN frontends is to concatenate all individual image frames into

one giant image frame to then be fed into one single CNN kernal. These inputs form the input of the frontend and are known as Concatenated Frame Images(CFIs), and the structure of a CFI based frontend is shown in Figure 2.8.

Garg et al. [116] were the first to use Concatenated Frame Images(CFIs) as shown in Figure 2.8 where a 2D CNN with the VGG topology was used as their frontend. Groups of successive image frames were intertwined within one giant image frame to form a CFI and a sequence of CFIs formed the input to an LSTM that was utilised for classification where they effectively transformed the temporal information per data-point into spatial information. Their model was trained and tested on videos from MIRACL-VC1 dataset and their best performance was achieved when freezing the VGG parameters and then training the LSTM, rather than training both the backend and frontend simultaneously.

Saitoh el. [117] devised a system that takes CFIs as an input, where lip images are merged into one single frame like the approach of Garg et al [116]. They used three different CNN models with three different topologies to extract features from CFIs that include the Network in Network(NIN) [118], AlexNet, and GoogLeNet.

Mesbah et al. [119] proposed a CNN structure (HCNN) based on Hahn moments that are effective in the sense that they can be used to extract the most useful information in image frames to reduce redundancy. Hahn moments are applied to the frames at the input to extract moments and input them to the CNN-based frontend and this helps to reduce the dimensionality of video images so that images can be represented with fewer dimensions. The frontend takes moment matrices as the input.

The third set-up used for 2D CNN frontends is to use not the static image frames themselves but a representation of how the image pixels change over time so the difference between successive image frames is used as the input to a visual frontend. Li et al. [120] acknowledged that dynamic features are a better representation of moving lips than static features, so they represented lip movements in the form of dynamic images. Dynamic images are obtained by calculating the first-order regression coefficients of every three consecutive image frames. The extracted features formed the input of an HMM which classified words from the Japanese word-based

ATR dataset that consisted of 2620 words for training and 216 words for testing.

It should be noted that the use of 2D CNNs for feature extraction in lip-reading when dealing with sequential inputs is limited because such an architecture would only learn spatial features without learning temporal features. Even if dynamic frames were to be used as opposed to static frames, the architecture would still be compromising on the loss of spatial features, so it is necessary to learn both spatial and temporal information. It is for this reason that 3D or spatiotemporal CNNs were introduced into lip-reading.

2.7.4 3D CNNs

The obvious difference between 2D and 3D networks is the extra dimension involved in the convolution process with the time dimension so the expression for convolution in Eq. 2.13 for a 3D CNN will be similar to that of Eq. 2.12 but with a kernel of temporal duration k_t and where t corresponds to the temporal coordinate such that y and $\omega \; \epsilon \; \mathbb{R}^{C \times k_t \times k_w \times k_h}$ [121] [122]. Figure 2.9 shows an outline a lip-reading system with a 3D CNN frontend.

$$(y \otimes \omega)_{ijt} = \sum_{c=1}^{C} \sum_{t'=1}^{k_t} \sum_{i'=1}^{k_w} \sum_{j'=1}^{k_h} y_{ct'i'j'} \omega_{c,t'+t,i'+i,j'+j} \tag{2.13}$$

Assael et al. [16] proposed an architecture with a frontend consisting of a spatiotemporal CNN, which extracts features from lip images with RGB pixels once pre-processing had been applied to videos from the GRID dataset which the architecture was trained and tested on. The backend consisted of 2 bidirectional GRUs, a softmax layer using ASCII characters as classes and a CTC for temporal alignments. Fung and Mak [123] proposed an architecture for decoding 10 sentences from the OuluVS2 corpus and they used a similar network for their backend, though their frontend used more 3D convolution layers and used max-out activation function instead of pooling. Their backend consisted of two bidirectional LSTMs with a softmax layer for classification whereby sentences were treated as individual classes, unlike Assael et al.'s [16] system which predicted sentences as sequences of ASCII characters.

Torfi et al. [123] proposed an audio-visual speech system that uses a coupled 3D CNN for the

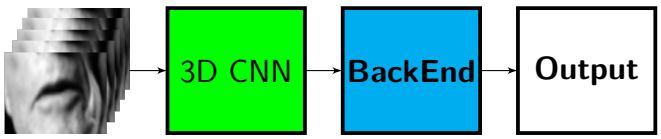


Figure 2.9: 3D CNN frontend.

visual stream with grayscale images as the input and four layers of 3D convolution in total. For the audio stream, the first layer uses a 3D convolutional layer to extract spatiotemporal features after extracting MFCC features from speech signals; whereas the second layer uses 2D convolution to extract spatiotemporal features. The outputs of both streams are then combined into a representation space, so that the correspondence between the audio and visual streams can be evaluated.

Chung et al. [17] constructed an audio-visual speech recognition system called Watch, Listen, Attend, and Spell (WLAS) which consists of four components: Watch, Listen, Attend, and Spell. The frontend consists of a "Watch" component for the visual stream and a "Listen" for the audio component, with "Attend" and "Spell" components making up the backend. The Watch component processes 5 consecutive grayscale images at a time with five 3D convolution layers, one fully connected layer, and three LSTM layers. Each LSTM at every timestep is part of an overall encoder LSTM configuration. The Listen component for the audio stream follows a similar structure except that Mel-frequency cepstrum coefficients(MFCCs) are used to extract features from the audio inputs as opposed to CNNs. The Spell component of the backend network consists of three LSTMs, two attention mechanisms [124], and a Multi-layer Perceptron(MLP). The attention mechanisms process the context information of Watch and Listen to generate the context vectors for the Watch and Listen components. The decoder LSTM network in Spell uses the previous step output, the previous decoder LSTM state and the previous context vectors of Watch and Listen to generate the decoder state and output vectors. Finally, a MLP and softmax layer predict the outputted sentence by generating probability distribution of possible output ASCII characters.

Xu et al. [125] proposed a network called LCANet specifically designed to encode rich semantic features, that was trained on the GRID corpus and decodes sentences on an ASCII character-level. The frontend of the LCANet entails 3D convolutional layers and a highway network, while

the backend uses Bidirectional GRU networks with a Cascaded Attention-CTC. The LCANet takes in images frames and uses the 3D-CNN to encode both spatial and temporal information with two layers of highway networks [126] on top of the 3D-CNN. The highway network module has two gates that allows the neural network to transfer some input information directly to the output.

Yang et al. [68] proposed an architecture called the D3D model for lip-reading Chinese words from the LRW-1000 dataset. It consists of a frontend with a spatiotemporal CNN following a similar topology to that of DenseNet that has stages of Convolution, Batch Normalization and pooling at the beginning; followed by three combinations of a DenseBlock and Trans-Block, plus a final Dense-Block at the end. Each Dense-Block contains two successive layers of convolution and batch normalisation while the Trans-Block contains three layers that include Batch Normalization, Convolution and Average Pooling. The backend consists of two Bidirectional GRUs with a softmax layer of 100 classes for each of the 100 words in the LRW-1000 dataset.

Chen et al. [90] constructed a neural network for Mandarin sentence-level lipreading consisting of two sub-networks. To predict the Hanyu Pinyin sequence for the input lip sequence, they combined a 3D CNN and a DenseNet with a two layer resBi-LSTM for the first part of the network, which was trained by a CTC loss function. The second part of the network converted Hanyu Pinyin into Chinese characters, and it consisted of a set of multi-headed attention that was trained using the cross-entropy loss function. The procedure in converting Hanyu Pinyin to Chinese characters does result in an 8% drop in accuracy rate. In consideration of the result, Chinese characters would be diverse on account of the different contexts whether Hanyu Pinyin is same or not.

3D CNNs can extract both spatial and temporal features more effectively than 2D CNNs. However, one drawback of 3D CNNs is that they require more powerful hardware and thus require high computation and storage costs. A compromise that is often made is to alleviate the limitations of both scenarios by using a 3D + 2D convolution neural network which consists of a mixture of 2D and 3D convolution layers. This helps to extract the necessary temporal features of lip movements and to limit the hardware capabilities required in performing feature

extraction for lip-reading.

2.7.5 2D + 3D CNNs

Frontends with a mixture of 2D and 3D CNNs will perform a combination of operations given in Eqs. 2.12 and 2.13. Figure 2.10 shows an outline a lip-reading system with a frontend containing 2D and 3D CNNs.

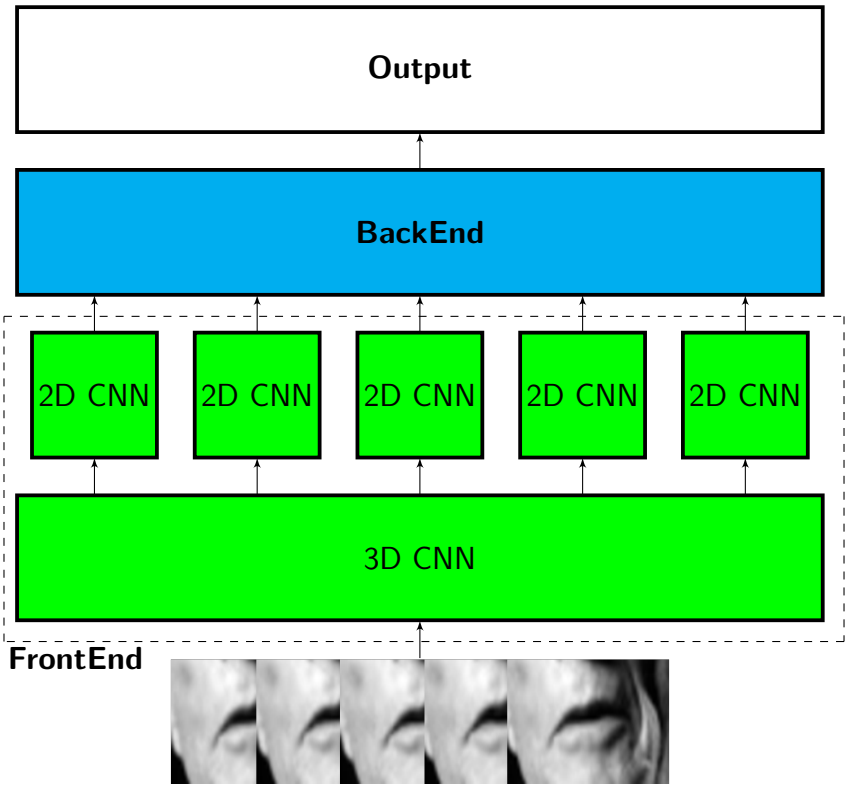


Figure 2.10: Frontend composed of 2D and 3D CNN kernels.

Stafylakis and Tzmiropoulos [127] proposed a visual speech recognition system for decoding words from the LRW corpus using grayscale images as an input. The frontend network consists of a 3D CNN and 2D ResNet, in which the 3D CNN has just one layer with which to extract short-term features of lip movements. The 2D ResNet has 34 layers which includes a max-pooling layer for reducing the feature vector's spatial dimensionality until the output is a one-dimensional feature vector. The backend is a two-layer Bidirectional LSTM with a softmax layer to classify one of 500 word classes.

Stafylakis and Tzmiropoulos proposed a visual speech system in [128] similar to that of [127]

but with modifications to the architecture which included the use of word embeddings, to summarize the information of the mouth region that is relevant to the problem of word recognition, while suppressing other varying attributes such as speaker, pose and illumination. Other modifications from their architecture of [127] include the use of a smaller ResNet to reduce the total number of parameters from ~24 million to ~17 million, and of word boundaries passed to the backend as an additional feature.

Margam et al. [129] devised a 3D+2D CNN architecture configuration for decoding ASCII character to predict spoken sentences from the GRID corpus, taking in RGB-pixelated images frames as an input. Their frontend consisted of two blocks of 3D CNNs followed by two blocks of 2D CNNs; where each 3D CNN block consists of a layer for convolution, pooling and batch normalisation, and each 2D CNN block will consist of layers for convolution and batch normalisation. Their backend consists of two bidirectional LSTMs with a CTC for temporal alignment.

In summary, CNNs are the most widely used network for feature extraction techniques in deep learning-based automated lip-reading. They have advantages over Autoencoders, RBMs and Feed-forward networks in that they are more effective at learning both spatial and temporal features as well as being the most effective in extracting relevant features from any redundant features. For spatio-temporal data, frontends will either deploy 2D CNNs, 3D CNNs or 2D+3D CNNs; but the use of 2D+3D CNNs appears to be the most preferred as they are a compromise between being able to extract the necessary temporal features of lip movements in the most effective way and to limit the hardware capabilities required in performing feature extraction.

The rationale behind the choice of feature extraction to use for the lip-reading system proposed in this thesis apart from being to utilise a pre-trained model is the ability use what appears to be most effective given the current trends in deep-learning based lip-reading and this is partly why the visual frontend uses a mixture of 2D and 3D CNN kernels.

2.8 Classification

The first neural network-based lip-reading systems were designed to classify isolated speech units such as individual letters, digits and words; where each speech segment or word was codified a class. This approach was sufficient for classifying visual speech that was limited to a limited number of discrete classes. For many systems that classified individual words such as Saitoh et al. [117] or Ngiam et al. [12], it was sufficient to use a backend that was composed of only a softmax layer for classification. Both of their architectures consisted of a frontend with a CNN for feature extraction a softmax layer backend to classify one of the possible words that had been uttered from the list of possible words contained within either of the OuluVS2 and LRW corpuses respectively.

A backend with solely a softmax layer would be sufficient for classifying speech in the form of a limited number of phrases where each phrase is treated as a class like Saitoh et al. [117] did with their approach. However when people utter phrases or even longer words, there is temporal information that can be exploited by neural networks to decipher between phrases and long words, which is why many visual speech recognitions systems use backends with networks for processing temporal sequences such as Recurrent Neural Networks(RNNs). They give a neural network architecture greater discriminative power when distinguishing between classes by learning conditional dependencies. Table 2.7(constructed as part of this research to highlight how lip-reading systems have advanced in order to generalise to natural everyday lip-reading), lists many of the automated lip-reading approaches which use deep-learning classification networks respectively. Many of them are listed in the works of [130] and [97].

2.8.1 Recurrent Neural Networks

RNNs are a sequence-based neural network used in many tasks including language modelling, machine translation and speech recognition. Recurrent Neural Networks(RNNs) can be used to predict sequences based on the output of particular timesteps which is what makes them useful for natural language processing tasks where in language models for instance, they can

predict the next character in a word or the next word in a sequence of words [35]. A vanilla RNN is the simplest form of RNN, but vanilla RNNs do suffer from the problem of vanishing gradients when trying to learn long-term dependencies. This is why RNNs used for lip-reading generally take the form of LSTMs or GRUs which consist of gates to control information that is transmitted through the network cells to control the gradient's value.

An LSTM is one variant of RNN which uses three gates to regulate the state and output at different timesteps [131]. An LSTM uses its gate structure to combine long and short-term memory to alleviate the problem of vanishing gradients. GRUs [132] are a more simplified form of RNN in comparison to LSTMs as they use just two gates instead of three. A diagram of an LSTM cell is shown in Figure 2.11 while a diagram of a GRU cell is shown in Figure 2.12 [133].

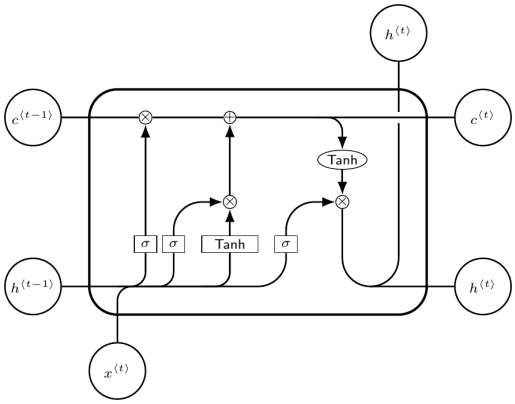


Figure 2.11: Long-Short Term Memory Cell [133].

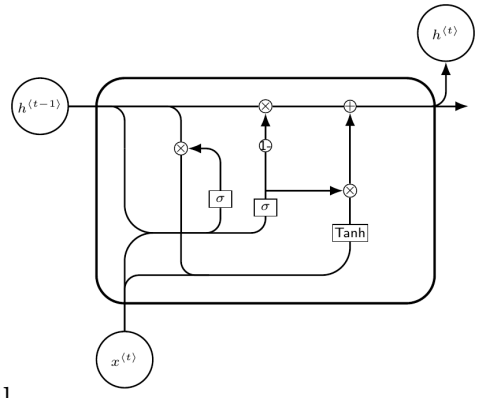


Figure 2.12: Gated Recurrent Unit Cell [133].

Unidirectional RNNs rely on just forward transmission, whereby the output depends on the input at that particular timestep and the output of the previous timestep. Bidirectional RNNs however rely on both forwards and backwards transmission where the output of a particular timestep relies not just on the current input and previous timestep output, but also on the successive timestep output too. A speech segment can be dependent on the successive segment as well as the previous one however. Bidirectional RNNs do use roughly double the number of parameters and so take longer to train.

For lip-reading sentences that are more wild and not repetitive such as those in the TIMIT and LRS2 corpuses, it is not possible to encode each sentence as a class and even to encode each word as a class is not feasible because of there are thousands of different possible words

to account for. Visual speech recognition systems that decode sentences will often use ASCII characters to decode sentences by learning conditional dependence relationships of how they appear in words.

When automating speech recognition in real time, information about where a particular character starts and ends in the image frame sequence will generally be unavailable and the use of RNNs to learn sequences of characters will not be sufficient without being able to learn the temporal alignment of the sequence.

Recurrent Neural Networks(RNNs) are capable of conditioning the output of a model on all the previous words in a sentence. Eq. 2.14 gives the expression for the hidden state h_t which is dependent on the current input x_t at time t, and hidden state from the previous step h_{t-1} which will in turn be dependant on the output of previous timesteps. The hidden state will therefore always be dependent on the hidden state from all previous timesteps. The output of a particular timestep y_t is given in Eq. 2.15. Unlike the feed-forward network, RNNs are not constrained by sequence length and longer sentences have no effect on the weight parameters [134].

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{2.14}$$

$$y_t = W_{hy}h_t + b_y \tag{2.15}$$

A GRU [135] consists of memory cells with weights W and a function H applied to the input according to Eq. 3.6. Each cell at a timestep t will have an input gate x, update gate u and reset gate r. All these parameters are updated according to Equations 2.16 to 2.19.

$$h_t = u \otimes \tilde{h}_t + (1 - u) \otimes h_{t-1} \tag{2.16}$$

$$u_t = \sigma(W_{xu}x_t + W_{hu}h_{t-1} + b_u) \tag{2.17}$$

$$\tilde{h}_t = \tanh(W_{xh}x_t + W_{hh}(r \otimes h_{t-1}) + b_h) \tag{2.18}$$

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \tag{2.19}$$

GRUs are able to select whether a unit for a timestep should have short or long term dependency. Reset gates help to capture short-term dependencies while update gates capture long terms dependencies and this helps to GRUs to ignore parts of sequences when needed. The reset gate r and update gate u can be switched on and off by containing values close to 1 and 0 respectively and Eqs. 2.20 to 2.22 have been derived indicating how a GRU behaves when the reset and update gate variables approach asymptotic limits.

A GRU behaves like a vanilla RNN when both gates are switched on as indicated by Eq. 2.20. When the update gate is switched off, the hidden state gives more attention to the previous hidden states(Eq. 2.21), while setting off the reset gate would cause the GRU to give more attention to the current input at that timestep(Eq. 2.22). With this in mind, a GRU-based a language model is better at modelling shorter length dependencies within a sentence for values of r close to zero which would make it less susceptible to the possibility of compound errors.

$$\lim_{(u,r)\to(1,1)} h_t = (W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{2.20}$$

$$\lim_{(u,r)\to(0,1)} h_t = h_{t-1} \tag{2.21}$$

$$\lim_{(u,r)\to(1,0)} h_t = (W_{xh}x_t + b_h) \tag{2.22}$$

2.8.2 Attention Mechanisms + CTCs

An Attention mechanism is one way of learning to temporally align predictions of an input sequence. For an input sequence of vectors $x = \{x_1, \ldots, x_{T_x}\}$, an attention-based RNN will predict a hidden state h, decoder state s and for every timestep t, a context vector c_t will be generated which is an indicator of how dependant the output at a timestep is to the output of another particular timestep.

The hidden state h_t for particular timestep is a function of the current input x_t and previous timestep h_{t-1} expressed in Eq. 2.23 while an expression for c is given in Eq. 2.24 for a series of hidden states across different timesteps. The symbols f and q are non-linear functions. The output y_t at an RNN timestep is predicted according to probability distribution of previous

outputs and the current input as expressed in Eq. 2.25 whereby g is a non-linear function; meanwhile Eq. 2.26 is an expression for the decoder state s.

$$h_t = f\left(x_t, h_{t-1}\right) \tag{2.23}$$

$$c = q\left(\{h_1, \ldots, h_{T_x}\}\right) \tag{2.24}$$

$$p(y_i|y_1, \ldots, y_{i-1}, x) = g\left(y_{t-1}, s_t, c\right) \tag{2.25}$$

$$s_i = f\left(s_{i-1}, y_{i-1}, c_i\right) \tag{2.26}$$

The context vector of a timestep is generated by calculating an alignment model e_{ij} which scores how well the input around position j and the output at position i match. This alignment model is then exponentiated and normalised by dividing by the sum of exponentiated alignment models to give a weight α_{ij}. Finally, the context vector for the timestep is calculated by summing over the all weights and annotations for that timestep. Using the decoder state and context vectors, the RNN can construct an output probability distribution to predict an output sequence. Relationships between the variables are shown in Eqs. 2.27 to 2.29.

$$e_{ij} = a\left(s_{i-1}, h_j\right) \tag{2.27}$$

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{T} \exp\left(e_{ik}\right)} \tag{2.28}$$

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{2.29}$$

There are two main problems posed by using attention mechanisms for temporal alignment in automated lip-reading. The first is the length variation between the input and output sequences in speech recognition that makes it more difficult to track the alignment and secondly, the basic temporal attention mechanism is too flexible and allows for extremely non-sequential alignments.

A Connectionist Temporal Classification (CTC) [136] model predicts frame labels and then looks for the optimal alignment between the frame predictions and the output sequence. A CTC can resolve the problem of input sequences and output sequences not being equivalent in length because of people speaking at different speeds.

If T is taken to the number of time steps in the sequence model, for example $T = 3$, a CTC defines the probability of the string "me" as $p(mme) + p(m\epsilon e) + \ldots + p(mee)$ and there exists a ϵ symbol in the case of repeated characters to make sure that the CTC does not group symbols when there are supposed to be repetitions.

For an input sequence $X = [x_1, x_2..., x_T]$ to a backend, an output sequence $Y = [y_1, y_2, ..., y_U]$ is predicted and the aim is to find the most likely sequence Y^*. A label l will have a set of possible paths with each path π corresponding to a possible frame prediction sequence. Eqs. 2.30 to 2.32 indicate how the CTC loss L_{CTC} is calculated.

$$p(\pi|x) = \prod_{t=1}^{T} p(\pi_t|x) \tag{2.30}$$

$$p(l|x) = \sum_{i} p(\pi_i|x) \tag{2.31}$$

$$L_{CTC} = -\ln p(l|x) \tag{2.32}$$

Assael et al. [16] were the first to introduce CTCs into lipreading when ASCII characters were used as units of classification. Bidirectional GRUs were used in the backend along with a CTC for temporal alignment and a CTC loss function to train the system.

The use of CTCs do have constraints, one being that input sequences must be longer than output sequences. CTCs also assume that character labels are conditionally independent and that each output is the probability of observing one particular label at a particular timestep. CTCs therefore focus more on local information from nearby frames than global information from all frames. It for this reason that lip-reading systems that use attention mechanisms perform better than those with CTCs for visual only speech recognition; whereas those that use CTCs are the better option for audio-visual speech recognition when there is available

audio.

Xu et al. [125] tackle the problem of the conditional independence limitation in CTCs by using a Cascaded Attention-CTC which tries to capture information from a longer context. Their frontend follows an Encoder-Decoder structure with two bidirectional GRUs in the Encoder and an Attention-CTC configuration with a hidden layer in between the Encoder and Decoder. The Decoder alleviates the conditional independence limitation by cascading the CTC with attention. This not only serves to address limitations of the CTC but also the limitations of using an Attention mechanism by itself because a Cascaded Attention-CTC can reduce uneven alignments during training in order to eliminate unnecessary non-sequential predictions between the decoded result and ground truth.

2.8.3 Transformers

RNNs account for the majority of frontend networks in neural network based lip-reading systems. However, a new trend in the use of Transformers has emerged in some of the most recent approaches to classification in lip-reading and they are appear to be replacing RNNs in many lip-reading systems.

Transformers are designed to allow parallel computation by processing entire inputs as at once rather than processing them sequentially like RNNs. Transformers require less time to train than RNNs because they avoid recursion, and they are better at capturing long term dependencies.

Afouras et al. [105] proposed three architectures that perform ASCII character-level classification for lip-reading sentences from the BBC LRS2 dataset. All three systems consist of an identical frontend with a 3D-CNN followed by a ResNet. The first architecture consisted of a backend with three stacked Bidirectional LSTMs trained with a CTC loss, and where decoding was implemented using a beam search that utilised information from an external language model. The second system used an attention-based transformer with an encoder-decoder structure that follows the baseline model of [137]. The Transformer model was the best performing

model and it attained better word accuracies than the Bidirectional LSTM for every evaluation scenario and the author observed for instance that the Transformer model was far better at generating to longer sequences than the Bidirectional LSTM model - particularly for sequences longer than 80 frames. Moreover, the Bidirectional LSTM model had a limited capacity for learning long-term, non-linear dependences and modelling complex grammar rules because of the CTC's assumption of timestep outputs being conditionally independent.

Ma et al. [104] proposed an audio-visual lip-reading system with a frontend composed of a spatiotemporal CNN and a ResNet-18 network. The visual backend uses the "Conformer" variant of the Transformer which follows a similar structure to that of Vaswani et al. [137]. It is convolution-augmented in that it uses convolutional layers in the Encoder because whilst Transformers are good at modelling long-range global context, they are less capable of extracting fine-grained local feature patterns - whereas CNNs can exploit local information.

A MLP is used to concatenate the outputs of the audio and visual streams whereby the output of the MLP forms the input of the Transformer Decoder which uses a hybrid CTC/Attention model that is specifically designed to address the individual limitations to the use of either a CTC or Attention model individually. This is done by generating a loss for the CTC and for the Conformer Encoder individually and adding them together using aggregated loss function [104](Eq. 2.33).

$$Loss = \alpha \log p_{CTC}(\boldsymbol{y}|\boldsymbol{x}) + (1-\alpha) \log p_{CE}(\boldsymbol{y}|\boldsymbol{x}) \tag{2.33}$$

2.8.4 Temporal Convolutional Networks

Temporal Convolutional Networks(TCNs) are another form of neural network that have emerged as an alternative to RNNs for sequence classification. Recently in many NLP tasks there has been a move towards the use of purely convolutional models for sequence modelling.

Like Transformers, TCNs have an advantage over RNNs in that they can process inputs in parallel as opposed to processing the input at every timestep sequentially. They are also

advantageous because they are flexible in changing receptive field size; which can be done by stacking more convolutional layers, using larger dilation factors, or increasing filter size which allows for better control of the model's memory size. Furthermore, TCNs do not suffer from the problem of exploding or vanishing gradients because they have a backpropagation path different from the temporal direction of the sequence, as well as lower memory requirement for training - particularly for long input sequences.

The third backend system used by Afouras et al. [105] for lip-reading sentences from the BBC LRS2 corpus was a Fully Convolutional(FC) model containing depth-wise separable convolution layers, which consists of layers for performing convolution along the spatial and temporal channel dimensions. The network contains 15 convolutional layers that were trained with a CTC loss where the decoding was performed in the same way as the Bidirectional LSTM system [105]. The FC model has advantages over the other two systems namely the transformer-based and Bi-LSTM-based systems, in that it uses fewer parameters and is quicker to train. Afouras et al. also noted that the FC model gave them greater control over the amount of future and past context by adjusting the receptive field. The FC model performed better than the Bidirectional LSTM model, though it did deliver diminishing returns on performance for sequences longer than 80 frames.

Martinez et al. [138] constructed a word-based lip-reading system similar to that of Petridis et al. [139] with a similar frontend that entails a spatiotemporal CNN followed by a ResNet-18 CNN. For the backend, the Bidirectional GRU has been substituted with a network in its place that they proposed called a Multi-Scale Temporal Convolutional Network(MS-TCN); devised to tailor the receptive field of a TCN so that long and short term information can be mixed up. A MS-TCN block consists of a series of TCNs, each with a different kernel size whereby the outputs are concatenated. Their system was trained and evaluated on the English datasets LRW and Mandarin dataset LRW-1000 achieving word accuracies of 85.3% and 41.4% respectively. In addition to improving on the accuracy of the system for Petridis et al. [139], they also noted a reduction in the overall GPU training time which was reduced by two thirds.

Ma et al. propose modifications to the system of Martinez et al. by using a Densely Connected

Temporal Convolutional Network (DC-TCN) instead of the MS-TCN contained within the frontend for the aim of providing denser and more robust temporal features. Two variants are used including Fully-Dense(FD) and Partially-Dense(PD) architectures, as well as an additional "Squeeze and Excitation" block within the network which is a lightweight attention mechanism to further enhance the model's classification power. They improve on the word accuracies of Martinez et al. to record word accuracies on the LRW and LRW-1000 datasets of 88.4% and 43.7%.

In summary of classification techniques, RNNs are the most frequently used backend network for predicting spoken sentences and are often used in conjunction with mechanisms for learning temporal alignment such as CTCs or Attention mechanisms. CTCs align sequences based on the conditional independence assumption, whereas attention mechanisms are better at modelling conditional dependence and this is why CTCs are the better option for audio-assisted speech recognition and why attention mechanisms are more effective for visual only speech recognition. RNNs however have started to be superseded by the use of Attention-Transformers and TCNs which both have advantages over RNNs in that they can perform parallel computation and are better at learning long-term dependencies. Out of all three networks, Attention-Transformers appear to have attained the best classification performance results when predicting sentences. However, TCNs do have advantages over both RNNs and transformers in that they take less time to train and are more flexible in changing receptive field size.

Table 2.7: Performance of lip-reading systems with deep learning-based classification algorithms.

Year	Reference	Feature Extractor	Classifier	Dataset	Class	Segment	Accuracy(%)
2011	Ngiam et al. [12]	Sparse Tensor PCA	Autoencoder	AVLetters	Alphabet	Alphabet	64.40
2013	Huang and Kingsbury [140]	DCT *plus* LDA	Deep Belief Network	Own data	Digits	Digits	35.70
2015	Moon et al. [141]	Deep Belief Network		AVLetters	Alphabet	Alphabet	55.30
2015	Mroueh et al. [63]	Scattering coefficients *plus* LDA	Feed-forward	IBM AV-ASR	Phonemes	Sentences	30.64^{P}
2015	Thangthai et al. [142]	AAM	Feed-forward	RM-3000	Phonemes	Sentences	77.49
2015	Thangthai et al. [142]	HiLDA	Feed-forward	RM-3000	Phonemes	Sentences	84.67
2016	Almajai et al. [143]	LDA *plus* MLLT *plus* SAT	Feed-forward	LILiR	Phonemes	Phrases	53.00
2016	Assael et al. [16]	3D-CNN	Bidirectional GRU *plus* CTC	GRID	ASCII	Phrases	93.40
2016	Chung and Zisserman [112]	VGG-M	LSTM	OuluVS2	Phrases	Phrases	31.90
2016	Chung and Zisserman [112]	SyncNet	LSTM	OuluVS2	Phrases	Phrases	94.10

Continued on next page

Table 2.7 – continued from previous page

Year	Reference	Feature Extractor	Classifier	Dataset	Class	Segment	Accuracy(%)
2016	Chung and Zisserman [42]	CNN		LRW	Words	Words	61.10
2016	Chung and Zisserman [42]	CNN		OuluVS	Phrases	Phrases	91.40
2016	Chung and Zisserman [42]	CNN		OuluVS2	Phrases	Phrases	93.20
2016	Lee et al. [113]	CNN	LSTM	OuluVS2	Phrases	Phrases	81.10
2016	Petridis and Pantic [144]	DBNF *plus* DCT	LSTM	AVLetters	Visemes	Alphabet	58.10
2016	Petridis and Pantic [144]	DBNF *plus* DCT	LSTM	OuluVS	Visemes	Phrases	81.80
2016	Saitoh et al. [117]	CFI *plus* NIN		OuluVS2	Phrases	Phrases	81.10
2016	Saitoh et al. [117]	CFI *plus* AlexNet		OuluVS2	Phrases	Phrases	82.80
2016	Saitoh et al. [117]	CFI *plus* GoogLeNet		OuluVS2	Phrases	Phrases	85.60
2016	Garg et al. [116]	CFI *plus* VGG	LSTM	MIRACL-VC	Words and Phrases	Words and Phrases	76.00
2016	Wand et al. [15]	Feed-forward	LSTM	GRID	Words	Phrases	79.50[b]
2017	Chung and Zisserman [17]	CNN	LSTM *plus* attention	OuluVS2	ASCII	Phrases	91.10
2017	Chung and Zisserman [17]	CNN	LSTM *plus* attention	MV-LRS	ASCII	Sentences	43.60
2017	Chung et al. [74]	CNN	LSTM *plus* attention	LRW	ASCII	Words	76.20
2017	Chung et al. [74]	CNN	LSTM *plus* attention	GRID	ASCII	Phrases	97.00
2017	Chung et al. [74]	CNN	LSTM *plus* attention	LRS	ASCII	Sentences	49.80
2017	Petridis et al. [107]	Autoencoder	LSTM	OuluVS2	Phrases	Phrases	84.50
2017	Petridis et al. [108]	Autoencoder	Bidirectional LSTM	OuluVS2	Phrases	Phrases	91.80
2017	Petridis et al. [109]	Autoencoder	Bidirectional LSTM	OuluVS2	Phrases	Phrases	94.70
2017	Torfi et al. [123]	3D CNN	Contrastive Loss	LRW	Words	Words	98.50
2017	Stafylakis and Tzimiropoulos [127]	3D-CNN *plus* ResNet	Bidirectional LSTM	LRW	Words	Words	83.00
2017	Stafylakis and Tzimiropoulos [128]	3D-CNN *plus* ResNet *plus* word boundaries	Bidirectional LSTM	LRW	Words	Words	88.08
2017	Wand and Schmidhuber [145]	Feed-forward	LSTM	GRID	Words	Phrases	42.40
2018	Afouras et al. [105]	3D-CNN *plus* ResNet	Bi-LSTM *plus* Language Model	LRS2	ASCII	Sentences	37.80
2018	Afouras et al. [105]	3D-CNN *plus* ResNet	Depthwise CNN	LRS2	ASCII	Sentences	45.00
2018	Afouras et al. [105]	3D-CNN *plus* ResNet	Attention-Transformer	LRS2	ASCII	Sentences	50.00
2018	Fung and Mak [108]	3D-CNN	Bidirectional LSTM	OuluVS2	Phrases	Phrases	87.60
2018	Hashmi et al. [146]	CFI *plus* CNN		MIRACL-VC	Words and Phrases	Words and Phrases	52.90
2018	Petridis et al. [139]	3D-CNN *plus* ResNet	Bidirectional GRU	LRW	Words	Words	82.00
2018	Petridis et al. [91]	Autoencoder	Bidirectional LSTM	AV Digits	Phrases	Phrases	69.70
2018	Petridis et al. [91]	Autoencoder	Bidirectional LSTM	AV Digits	Digits	Digits	68.00
2018	Wand et al. [147]	Feed-forward	LSTM	GRID	Words	Phrases	84.70
2018	Xu et al. [125]	3D-CNN *plus* Highway	Bidirectional GRU *plus* Attention	GRID	ASCII	Phrases	97.10
2018	Afouras et al. [148]	3D-CNN *plus* ResNet	Transformer-CTC	LRS2	ASCII	Sentences	45.30
2018	Afouras et al. [148]	3D-CNN *plus* ResNet	Transformer-Seq2seq	LRS2	ASCII	Sentences	51.70
2018	Yang et al. [68]	2D CNN	Bidirectional GRU	LRW-1000	Words	Words	25.76
2018	Yang et al. [68]	DenseNet3D	Bidirectional GRU	LRW-1000	Words	Words	34.76
2018	Yang et al. [68]	2D+3D CNN	Bidirectional GRU	LRW-1000	Words	Words	38.19
2018	Mattos et al. [149]	CNN		GRID	Visemes	Visemes	64.80
2018	Oliveira et al. [18]	CNN		GRID	Visemes	Visemes	67.30
2019	Lu et al. [45]	CNN	LSTM *plus* Attention	Own data	Digits	Digits	88.20
2019	Shillingford et al. [41]	3D-CNN	Bidirectional LSTM *plus* Finite-state transducer	LSVSR	Phonemes	Sentences	59.10
2019	Shillingford et al. [41]	3D-CNN	Bidirectional LSTM *plus* Finite-state transducer	LRS3-TED	Phonemes	Sentences	44.90
2019	Courtney and Sreenivas [150]	Res-Bi-Conv-LSTM		LRW	Words	Words	85.20

Continued on next page

Table 2.7 – continued from previous page

Year	Reference	Feature Extractor	Classifier	Dataset	Class	Segment	Accuracy(%)
2019	Jang et al. [151]	CFI *plus* QVGG *plus* Committee		OuluVS2	Phrases	Phrases	90.90
2019	Zhou et al. [152]	CNN	Bidirectional LSTM *plus* Modality Attention Mechanism	Chinese TV	Chinese *plus* ASCII	Sentences	93.15
2019	Mesbah et al. [17]	CFI *plus* Hahn CNN		OuluVS2	Phrases	Phrases	93.72
2019	Mesbah et al. [119]	CFI *plus* Hahn CNN		LRW	Words	Words	58.20
2019	Margam et al. [129]	2D+3D CNN	Bidirectional LSTM *plus* CTC	GRID	Words	Sentences	98.70
2019	Margam et al. [129]	2D+3D CNN	Bi-LSTM *plus* CTC	Indian English	Words	Sentences	87.70
2019	Weng and Kitani [153]	3D-CNN	Bi-LSTM	LRW	Words	Words	84.11
2019	Zhang et al. [114]	VGG-M *plus* ResNet *plus* Bi-LSTM *plus* CTC	GRU *plus* Attention	CCTC	Pinyin-to-Hanzi	Sentences	50.20
2019	Wang et al. [154]	3D-CNN	Bi-Conv-LSTM	LRW	Words	Words	83.34
2019	Wang et al. [154]	3D-CNN	Bi-Conv-LSTM	LRW-1000	Words	Words	36.91
2020	Lu et al. [115]	CNN *plus* ResNet	LSTM	Own data	Digits	Digits	87.00
2020	Chen et al. [90]	3D-CNN	resBi-LSTM	NSTDB	Pinyin-to-Hanzi	Words	49.56
2020	Zhang et al. [155]	3D-CNN *plus* ResNet	Bidirectional GRU	LRW	Words	Words	85.20
2020	Zhang et al. [155]	3D-CNN *plus* ResNet	Bidirectional GRU	LRW-1000	Words	Words	45.24
2020	Xiao et al. [156]	3D-CNN *plus* ResNet	Bidirectional GRU	LRW	Words	Words	84.13
2020	Xiao et al. [156]	3D-CNN *plus* ResNet	Bidirectional GRU	LRW-1000	Words	Words	41.93
2020	Luo et al. [157]	3D-CNN *plus* ResNet	Bidirectional GRU	LRW	Words	Words	83.50
2020	Luo et al. [157]	3D-CNN *plus* ResNet	Bidirectional GRU	LRW-1000	Words	Words	38.70
2020	Zhao et al. [158]	3D-CNN *plus* ResNet	Bidirectional GRU	LRW	Words	Words	84.41
2020	Zhao et al. [158]	3D-CNN *plus* ResNet	Bidirectional GRU	LRW-1000	Words	Words	38.79
2020	Fenghour et al. [159]	3D-CNN *plus* ResNet	Linear Decoder Transformer *plus* GPT Transformer	LRS2	Visemes	Sentences	65.00
2020	Martinez et al. [138]	3D-CNN *plus* ResNet	Temporal CNN	LRW	Words	Words	85.30
2020	Martinez et al. [138]	3D-CNN *plus* ResNet	Temporal CNN	LRW-1000	Words	Words	41.40
2020	Ma et al. [160]	3D-CNN *plus* ResNet	Temporal CNN	LRW	Words	Words	88.36
2020	Ma et al. [160]	3D-CNN *plus* ResNet	Temporal CNN	LRW-1000	Words	Words	43.65
2021	Ma et al. [104]	3D-CNN *plus* ResNet *plus* Conformer Encoder	Decoder Transformer	LRS2	Pinyin-to-Hanzi	Sentences	62.10
2021	Ma et al. [161]	3D-CNN *plus* ResNet	Temporal CNN	LRW	Words	Words	88.50
2021	Ma et al. [161]	3D-CNN *plus* ResNet	Temporal CNN	LRW-1000	Words	Words	46.60
2021	Prajwal et al. [162]	3D+2D CNN *plus* Visual Transformer Pooling	Attention-Transformer	LRS2	Sub-Words	Sentences	77.40
2021	Prajwal et al. [162]	3D+2D CNN *plus* Visual Transformer Pooling	Attention-Transformer	LRS3	Sub-Words	Sentences	69.3

b - Speaker Dependent V - Viseme accuracy P - Phoneme accuracy C - Correctness

The Viseme Classifier proposed in this thesis decodes continuous sequences of visemes and so the backend needs to be trained to automatically perform the temporal alignment of visemes. In real time, the boundaries of where each viseme starts and stops is unknown.

Of course, a CTC or Attention Mechanism can be trained to learn the temporal alignment of continuous class sequences and either of these two mechanisms can be utilised. The choice of mechanism is not a major focus of this thesis. However the use of attention is preferred because

unlike CTCs, they do not suffer from the constraint of input sequence needing to be longer than the output sequence.

2.9 Summary

One can see a progressions of visual speech recognition systems moving from the use of traditional algorithms for letter and digit classification to the use of deep neural networks for predicting words and sentences thanks to the development of more advanced corpuses such as BBC-LRS2, LRS3-TED, LSVSR and LRW-1000. New datasets not only cover larger vocabularies covering thousands of words and uttered by thousands of people, they also feature people speaking in varying poses, lighting conditions and resolutions.

Lip-reading systems consist of components for feature extraction and classification. 2D+3D CNNs are the most preferred network for frontends because of their ability to learn spatial and temporal features though Autoencoders do have the advantage of being able to map visual feature data from higher dimensional space into lower dimensional space without the need for any labelled classification.

RNNs in the form of LSTMs and GRUs form the majority of classification networks. In recent years though, Transformers and TCNs have started to replace RNNs due to their ability to better perform parallel computation, learn long-term dependencies and be trained in a shorter period of time.

Chapter 3

Literature Review

This Chapter gives a review of the latest trends in automated lip-reading where lip-reading systems have seen an evolution in recognising small isolated speech segments in the form of isolated numbers and letters to predicted words and sentences from videos with people speaking from both frontal and profile views. Possible classification schemas used for lip-reading is one area of lip-reading research that does deserve more research attention and the use of visemes has been identified as a gap in the literature review.

The rest of the chapter is organized as follows: First in Section 3.2, a discussion of many of the latest trends in automated lip-reading is provided based on a review of the most up-to-date lip-reading systems up until early 2021; then in Section 3.3, a comparison of different classification schemas used for lip-reading including ASCII characters, phonemes and visemes; while Section 3.4 talks about language models and the importance of language model in a speech recognition system not only to distinguish between words that share identical lip movements but in that their inclusion can boost the performance accuracy of lip-reading systems for all classification schema. Finally, Section 3.5, the rationale behind the proposed lip reading system is explained with full details about all the different components of the overall lip reading system.

3.1 Introduction

Research in automated lip-reading is a multifaceted discipline. Due to breakthroughs in deep neural networks and the emergence of large-scale databases covering vocabularies with thousands of different words, lip-reading systems have evolved from recognising isolated speech units in the form of digits and letters to decoding entire sentences.

Traditional non-deep learning methods with hand-crafted techniques were the first methods used for the automation of lip-reading and such methods include, for instance, Hidden Markov Models (HMMs) [30] [32] [33] [34] [35]. A variety of different feature extraction techniques have been used including Linear Discriminant Analysis(LDA), Principal Component Analysis(PCA), Direct Cosine Transformations(DCTs) and Active Appearance Models(AAMs).

In recent years, more visual speech recognition systems have moved towards the use of deep learning networks for both feature extraction and classification and in 2011, Ngiam et al. [12] first proposed a deep audio-visual speech recognition system based on Restricted Boltzmann Machines(RBMs) [13]. This means that traditional feature extraction techniques like PCA have been superseded by the use of neural networks. Feed-forward networks, Autoencoders and Convolutional Neural Networks(CNNs) are examples of networks that are used in lip-reading frontends. CNNs account for majority of neural network frontends as they are better at learning both spatial and temporal features, and more effective at extracting relevant features.

For classification, lip-reading backends predict speech sequential in nature like words or sentences and tend to use sequence processing networks like Recurrent Neural Networks(RNNs). RNNs take the form of either Long-Short Term Memory networks(LSTMs) or Gated Recurrent Units(GRUs). Recently, alternative classification networks to RNNs such as Attention-based Transformers and Temporal Convolutional Network(TCNs) have been used in lip-reading backends.

A number of surveys on the topic of automated lip-reading with a particular focus on deep learning have been written, for example, [130] and [97]. This chapter has some unique insights in that there is a more in-depth comparison of some of the advantages of other alternative frontend

networks to CNNs such as feedforward neural networks and autoencoders; and for classification, there is focus on lip-reading architectures with Attention-Transformers and TCNs which have advantages over RNNs; as well as there being a comparison of the different classification schema used in lip-reading. This literature review also covers some of the most up-to-date approaches of late 2020 and early 2021.

3.2 Trends in Lip-Reading

The previous chapter reviewed many automated lip-reading systems running that had been proposed running from 2007 to 2021. One aspect of lip-reading systems that is noticeable is that a particular system will have been trained to classify a particular speech segment whether it is speech in the form of letters, digits, words or phrases or sentences.

The AVLetters database is the most widely used corpus for alphabet recognition. Zhao et al. [76] used LBP-TOP for feature extraction and a Support Vector Machine(SVM) for classification and they attained a 62.80% word accuracy rate(WAR). Pei et al. [164] recorded the highest WAR of 69.60% with a RFMA based lip-reading system. Petridis and Pantic [144] used a frontend that combined Deep Belief Network features and DCT features, with an LSTM for the backend achieving a 58.10% classification accuracy. Hu and Li [165] proposed a system based on multimodal RBMs called Recurrent Temporal Multimodal Restricted Boltzmann Machines and achieved a WAR of 64.63%.

CUAVE is the most frequently used database for digit recognition. Papandreou et al. [166] used an AAM for feature extraction with a HMM for classification for performing digit recognition and they recorded a 83.00% word recognition rate. Ngiam et al. [12] achieved a 68.70% word recognition rate using an RBM-Autoencoder. Rahmani [167] extracted deep bottleneck features, and then used a GMM-HMM for the language model to achieve a WAR of 63.40%. Petridis et al. [107] achieved a WAR of 78.60% using the dual flow method.

GRID is one of the oldest and most frequently used databases for predicting phrases. Wand et al. [15] experimented with three different feature extraction techniques for their backend

that included Eigenlips, HOG, and feedforward neural networks. The lip-reading systems that used Eigenlips and HOG for the respective frontends utilised an SVM for the backend, while the lip-reading system with the feedforward network in the frontend used an LSTM for the backend. Performance results indicate that the combination of the feedforward network with an LSTM was the best model. Assael et al. [16], Xu et al. [125] and Margam et al. [129] obtained word accuracies of 95.20%, 97.10%, and 98.70% respectively through the use of spatiotemporal convolutional networks and Bidirectional RNNs.

OuluVS2 is the most widely used multi-view database. Lee et al. [113] used a frontend that combined DCT and PCA features, and an HMM to attain a 63.00% word accuracy rate for phrase prediction. They also constructed a lip-reading system that utilised a CNN for feature extraction and an LSTM for classification achieving a 83.80% word accuracy rate. Wu et al. [168] combined SDF features with STLP features while using an SVM for classification, to achieve a 87.55% classification accuracy. Petridis et al. [1] obtained a 96.90% word recognition rate based on the three-stream method.

LRW is one of the most challenging datasets there is for word classification which Chung and Zisserman [42] used for training and validation. They obtained a word accuracy rate(WAR) of 61.10% with a spatiotemporal CNN, while Torfi et al. [123] used a coupled 3D CNN for their lip-reading system achieving a WAR of 98.50%. Stafylakis and Tzimiropoulos [127] used a 3D CNN and ResNet for their frontend with a Bidirectional LSTM backend obtaining a WAR of 83.00%. In recent years; Zhang et al. [155], Xiao et al. [156], Luo et al. [157] and Zhao et al. [158] have all used a frontend with a 3D CNN and ResNet along with a Bidirectional GRU for the backend and they all recorded state-of-the-art performance results on the LRW corpus with WARs of 85.20%, 84.13%, 83.50% and 84.41% respectively. The best results that were recorded for the validation on the LRW set were for the systems proposed by Martinez et al. [138] and Ma et el. [160] [161] who all used a 3D CNN and ResNet for the frontend with a TCN for the backend and they correspondingly achieved WARs of 85.30%, 88.36% and 88.50%. As discussed in Section 2.8, TCNs have advantages over RNNs and they are set to replace RNNs for many sequence processing tasks.

For the BBC-LRS2 database, Chung et al. [17] proposed a Watch-Attend-and-Spell system that achieved a WAR of 49.80%. Afouras et al. [148] proposed two approaches which both used a 3D CNN plus ResNet for the frontend. One of their approaches used an attention-transformer for the backend that trained with a CTC loss achieving a WAR of 45.30%. Their other approach also used a backend with a Transformer, but that was trained with a seq2seq loss and achieved a WAR of 51.70%. Ma et al [104] proposed a frontend with a 3D-CNN, ResNet plus Conformer Encoder in tandem with a backend that used Decoder Transformer and accomplished a word accuracy rate of 62.1%. Fenghour et al. [159] devised a system that decoded videos in two stages where visemes were predicted for the first stage using a 3D-CNN plus ResNet with a Linear Decoder Transformer, and then words where predicted using a converter that calculated perplexity scores using the pre-trained GPT transformer where a WAR of 64.00%. More recently, Prajwal et. [162] proposed an architecture consisting of a backend with a spatiotemporal CNN and Visual Transformer Pooling in conjunction with an attention-transformer backend with recorded a WAR of 77.40%.

For the task of recognising shorter speech segments, traditional methods have outperformed deep learning-based methods in terms of performance when the dataset used to provide video samples was too small to be used with deep learning. This is because deep learning requires large numbers of training samples and because the focus of automated lip-reading research has moved towards classifying larger speech units in the form of words and entire sentences in continuous speech, plus there is very little demand and effort to attempt to increase the volume of training samples for people uttering isolated digits and letters. For sentences prediction, deep learning methods significantly outperform traditional methods. For word and sentence prediction, Transformers and TCNs are starting to replace RNNs due to their ability to better perform parallel computation and learn long-term dependencies.

The most recent approaches to automated lip reading are deep learning-based and they largely focus on decoding long speech segments in the form of words and sentences using either words or ASCII characters as the classes to recognise [127] [17] [42] [105] [16] [41]. Lip reading systems that are designed to classify words often use individual words as the classification schema where every word is treated as a class. In recent years, very good accuracies have been achieved for

word-based classification on some of the most challenging audio-visual datasets for words, such as LRW [42] and LRW-1000 [68].

Contrastingly, however, lip reading sentences have not succeeded in attaining accuracies as good as word-based approaches. It still remains an ongoing challenging task to automatically lip reading people uttering sentences which cover a wide range of vocabulary and contain words that may not have appeared in the training phase while using the fewest classes possible. The main obstacles to lip reading sentences are:

- Lip reading systems that use words or ASCII characters as classes can only predict words that the systems have been trained to predict because in the case of using words as a class, the word needs to be encoded as a class and presented in the training phase; while in the case of ASCII characters, the prediction of words is based on combinations of characters having been presented in the training phase as patterns.
- The models must be trained to cover a wide range of vocabulary which requires a significant number of parameters in the models to be optimised and a significant volume of training data to be used.
- They often require curriculum learning-based strategies [169] [170] which involve further pre-processing, whereby the videos of individuals speaking in the training data have to be clipped so that the models can be trained on single word examples initially, with the length of the sentences being gradually incremented.

3.3 Classification Schema

The first automated approaches to lip-reading started off with recognising a limited number of speech units in the form of digits, letters and words; especially as the first audio-visual datasets that were available for training lip-reading systems were limited and only focused on the classification of small isolated speech segments. For this reason it was sufficient to encode each speech segment as a class.

Eventually, the emergence of more audio-visual training data covering a wider range of vocabulary saw the development of lip-reading systems with entire words a classes. Some approaches encoded entire phrases when performing the task of speech recognition in videos of people uttering a limited number of structured and repetitive phrases.

Some of the largest and most recent of lip-reading corpuses consist of people speaking in a continuous manner with vocabularies coverings thousands of different words, and so many lip-reading systems that have been trained to predict entire sentences have opted for the use ASCII characters as a classification schema as opposed to encoding every word as a single class. This allows for fewer classes to be used and for a reduction in the creation of computational bottleneck [36]. The use of ASCII characters also allows for natural language to be modelled due to the conditional dependence relationships that exist between ASCII characters. This makes it easier to predict characters and words [17] [16] [127].

However, even the use of ASCII characters for automated lip-reading of speech covering an extensive range of vocabulary has its limitations. Neural networks for speech recognition systems that use either words or ASCII characters as classes are only able to predict words that the system has been trained to predict, because in the case of using words as a class, the word needs to be encoded as a class and have been present in the training phase. While for the case of ASCII characters, the prediction of words is based on combinations of characters having been observed in training as patterns.

Furthermore, the models must be trained to cover a wide range of vocabulary which would require a significant number of parameters, lots of hyperparameters to be optimised and a significant volume of training data to be used. This is in addition to the requirement of curriculum learning-based strategies [169] [170] which involve further pre-processing, such as the clipping of training videos with individuals speaking so that the models can be trained on single word examples to begin with, before gradually incrementing the length of the sentences being spoken.

There are alternative class systems that have been proposed for automated lip-reading systems such as byte-pairs and sub-words. Byte-pairs were suggested as a potential schema in the

conclusion to [105]'s work though to the best of our knowledge, there is no lip-reading system that uses byte-pair encoding to predict sentences. More recently, Prajwal et al. [162] proposed a lip-reading a system that uses sub-words as classes and their architecture achieved a 77.4% word accuracy rate when predicting sentences from LRS2 corpus. They noted several advantages of using sub-words to ASCII characters which include reduction in output sequence length which accelerates both training and inference, but also the ability to encode prior language information to improves the overall performance. The use of sub-words as class also helps to solve the "out-of-vocabulary problem" by predicting words not seen in the training data as long as the word consisted of sub-tokens, but still suffers from the problem of unseen sub-word tokens, misspelled words and abbreviations [163]. Sub-word encoding also has some of the same limitations that come with using ASCII characters such as the need to encode an extensive number of softmax classes to cover all potential sub-word tokens, and the requirement of curriculum learning strategies.

Other less frequently used classification schema include visemes and phonemes. The usage of visemes for decoding speech when trying to predict sentences has some unique advantages. Firstly, the prediction of speech as sequences of visemes as classes as opposed to sequences of either words or ASCII characters would require a smaller overall number of classes which alleviates computational bottleneck. In addition, the use of visemes does not require pre-trained lexicons, which means that a lip-reading system which classifies visemes can in theory be used to classify words that have not been seen during training. A lip-reading system that predicts speech using visemes as classes can be generalised to decoding speech from people speaking in other languages because many different languages often share identical visemes.

The general classification performance for recognising individual segmented visemes has been less satisfactory compared with the classification of words. This is due to the natures of visemes tending to have a shorter duration than words which results in there being less temporal information available to distinguish between different classes, as well as there being more visual ambiguity when it comes to class recognition [18].

Moreover, the eventual prediction of words and sentences based on decoding visemes requires a

two-stage procedure where visemes will be decoded as the first stage and with a viseme-to-word conversion process being performed as the second stage. One set of visemes can correspond to multiple different sets of phonemes or sounds; unlike the use of ASCII characters where there is one-to-one mapping relationship when mapping characters to possible words or sentences.

The viseme-to-word conversion is a challenge because once visemes have been classified, there is a need to disambiguate between homopheme words(words that look identical when spoken but sound different [19]). This bottleneck exists because of the one-to-many mapping correspondence between visemes and phonemes. The conversion process requires a language model to determine the most likely words that have been uttered.

For the specific task of viseme identification, recent approaches have included the use an SVM to distinguish between 6 viseme classes [171] achieving a 63.0% accuracy, a CNN-based approach which obtained 55.7% accuracy on the identification of 12 viseme classes [172], an HMM for recognizing 13 viseme classes with 46.6% accuracy [173], and the use of an Active Appearance Model in classifying 18 viseme classes from a small dataset of two users that achieved around 45% accuracy [174]. Lower accuracies are generally expected for short speech segments and previous work has demonstrated that the success of automated lip reading increases for longer words, indicating the importance of temporal features [16], for visual speech recognition.

A Generative Adversarial Network-assisted CNN achieved an accuracy of 67.3% [18] for recognising 16 visemes on large synthetic dataset of 40,800 images extracting from the GRID [59] audio-visual corpus consisting of 34 speakers. However the large number of training samples that was required to train the model may limit the application of the approach in terms of reproducibility.

Phonemes have been more frequently used than visemes as an intermediate classification schema in lip-reading where speech is decoded in the form of phonemes, which are then converted to words [41] [142] [175] [176] [173]. The classification of phonemes as individual units using only visual speech can never be done with as much precision as classifying individual visemes due to the fact that many phonemes share identical visemes and therefore look the same so context is needed to resolve that problem.

Phonemes are more preferred to visemes though because the conversion of phonemes to words will always comprise of less ambiguity than the conversion of visemes to words. This is because there are significantly fewer homophone words, or words that sound the same in the English language than homopheme words. Some of the language models used to perform the phoneme-to-word conversion such as WFSTs and HMMs use Markov chains and are limited in performing viseme-to-word conversion with good precision due to their inability to detect semantic and syntactic information needed to discriminate between words with identical visemes.

It still remains to be seen which is the most accurate classification schema to utilise out of visemes, phonemes and ASCII characters. The performance of a lip-reading system that uses ASCII characters can itself be enhanced by the inclusion of a language model which means the decoding of ASCII characters in predicting sentences can be performed as a two-stage procedure. Afouras et al. [105] do include a character-based language model to increase the likelihood of a word being correctly predicted however, some of the sentences that the model does not predict correctly are not as grammatically sound as the ground-truth sentences. So the model's performance itself could be enhanced by including a word-based language model to ensure that sentences being predicted are the most likely given the combination of words using a word-based language model to calculate sentence perplexity.

3.4 Language Model Implementations

Viseme-to-word conversion is related to a language model and various ways of the implementing a language model. Conversion methodologies used to predict word from visemes can be grouped into two categories: statistical conversion models and neural conversion models. This section provides the essential fundamentals of a language model and the different ways of implementing of a language model to analyse how effective they are when applied in a viseme-to-word conversion model.

3.4.1 Implementation of a language model

The language model will predict the most likely set of words to have been spoken given the spoken visemes and the two ways to implement a language model include statistical language models and neural models. Statistical language models predict words based on the preceding words in the sequence according the Markov assumption whereas neural language models use deep neural networks.

Algorithms like Weighted Finite State Transducers(WFSTs) [177] and Hidden Markov Models(HMMs) [19] are some examples of statistical conversion models as they implement language models based on Markov chains or N-grams, which assume that each word in a sentence depends only its previous N-1 predecessors.

Statistical models based on N-grams are limited in comparison to neural models because they are a sparse representation of language which model sentences based on the probability of words in combination and would naturally give a zero probability to combinations of words that have not previously appeared [178]. Furthermore N-grams fail to accurately predict semantic and syntactic details of sentences [178], but one fundamental problem with N-grams is that they need a large value of N to produce an accurate language model which requires lots of computational overhead.

An N-gram model predicts sequences of words according to the Markov process where the probability of the next word in a sequence is predicted based on the previous $(N-1)$ words. Eq. 3.1 gives the ideal chain rule of probability P to apply to any language model with a sequence of K words. However as K increases, the computation because impossible so statistical language models use the Markov assumption given in Eq. 3.2.

$$P(w_1, w_2, ..., w_K) = \prod_i P(w_i | w_1, w_2, ..., w_{i-1}) \tag{3.1}$$

$$P(w_1, w_2, ..., w_K) = \prod_i P(w_i | w_{i-N+1}, ..., w_{i-1}) \tag{3.2}$$

N-grams are an approximation of the Markov assumption and the problem with N-grams is

that context is only limited to the preceding $N-1$ words(Eq. 3.3), and though one exploit more contextual information by increasing the value of n; this comes at the cost of increasing the computation of the model [179]. Bigrams language models will only be able to predict words based on the previous word in a sentence(Eq. 3.4) which in practice is insufficient to disambiguate words sharing identical visemes let alone even homophone words that share identical phonemes.

$$P(w_i|w_{i-N+1},...,w_{i-1}) = \frac{count(w_{i-N+1},...,w_{i-1},w_i)}{count(w_{i-N+1},...,w_{i-1})} \tag{3.3}$$

$$P(w_i|w_{i-1}) = \frac{count(w_i,w_{i-1})}{count(w_{i-1})} \tag{3.4}$$

One major difference between statistical models and neural models is that whilst the former treats each word a fixed representation like a one-hot-vector [179], neural models use the concept of distributed representations where words are treated as continuous vectors each with a discrete number of features where each feature represents a semantic dimension in feature space. This means that words which are semantically similar are closer together in vector space. Neural models are a dense representation of language which avoid what is known as the curse of dimensionality [179].

Eq. 3.5 gives the expression for cosine similarity $S_C(\boldsymbol{w}_a, \boldsymbol{w}_b)$ which can be used to calculate the similarity between two word vectors w_a and w_b [180]. One hot vectors for two semantically similar words would automatically result in a value of S_C equal to 0 because the vectors are orthogonal but for continuous word vectors, one would expect a value of $S_C \approx 1$ [180].

$$S_C\left(\boldsymbol{w}_a, \boldsymbol{w}_b\right) = \frac{\boldsymbol{w}_a \cdot \boldsymbol{w}_b}{\|\boldsymbol{w}_a\|\|\boldsymbol{w}_b\|} \tag{3.5}$$

Feed-forward neural networks are an example of a neural conversion model and they have advantages over statistical conversion models modelling N-grams that use HMMs or WFSTs in that they are not limited by data sparsity or the inability to learn semantic and syntactic information which means that they can even model unseen combinations of words not seen

in training. The modelling of unseen word combinations is necessary for ensuring that the viseme-to-word converter is not limited to predicting combinations of words seen in training for any given combination of words.

The output of a feed-forward network at a certain timestep will always be conditioned on a window of the previous $N-1$ outputs which a softmax layer is applied to. As seen in Eq. 3.6, the fully connected layer a_k(for class k corresponding to one of N classes) uses only hidden states from the previous $n-1$ steps. Increasing the window size requires more weight parameters and increases the complexity of the model [179].

$$P(w_t = k|w_{t-N+1}, ..., w_{t-1}) = \frac{e^{a_k}}{\sum_{i=1}^{N} e^{a_i}} \tag{3.6}$$

However like N-grams, feed-forward networks still suffer from the fundamental problem in that they used fixed-size windows to give context where the output of a timestep is only conditioned on a limited number of previous timesteps. They are not always able to utilise all the context necessary in exploiting semantic or syntactic information needed to distinguish between words that share identical visemes. Recurrent Neural Networks(RNNs) on the other hand are capable of conditioning the output of a model on all the previous words in a sentence.

Statistical models predict words according to ratios of counts for sequences of words within window of n words according to Eq. 3.7. Neural models with a fixed context predict words according to the relationship of feature vectors within a fixed window of n words according to Eq. 3.8. Neural models with limited context predict words according to the relationship of feature vectors for all previous words (Eq. 3.9). Figure 3.1 shows the taxonomy of the different viseme-to-word conversion models and they can be decomposed into statistical models, neural models with fixed context and neural models with unlimited context.

$$P(w_t|w_1, ..., w_{t-1}) = \frac{count(w_{t-N+1}, ..., w_{t-1}, w_t)}{count(w_{t-N+1}, ..., w_{t-1})} \tag{3.7}$$

$$P(w_t|w_1, ..., w_{t-1}) = f(w_t|w_{t-n+1}, ..., w_{t-1}) \tag{3.8}$$

$$P(w_t|w_1, ..., w_{t-1}) = f(w_t|w_1, ..., w_{t-1}) \tag{3.9}$$

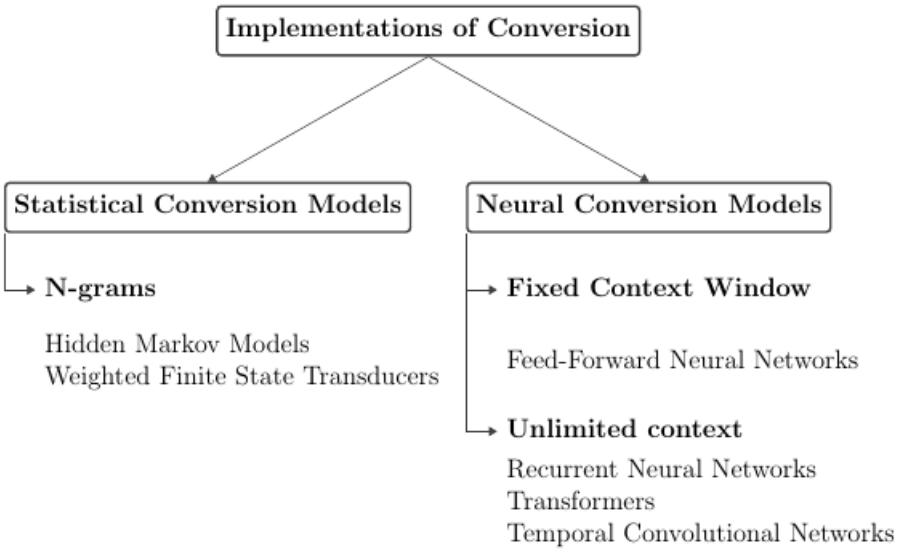


Figure 3.1: Taxonomy of viseme-to-word conversion models.

As discussed in Chapter 2, most RNNs used for language modelling take the form of either LSTMs or GRUs because traditional vanilla RNNS are susceptible to the problem of vanishing or exploding gradients for very long sequences. This allows them to select whether the output of a timestep should be give more focus to either to the input at that current timestep or the outputs of the previous timesteps.

3.4.2 Comparison of viseme-to-word conversion models

Table 3.1 gives a summary of some of the approaches to two-stage visual speech recognition that use visemes as the intermediate class. TCD-TIMIT [80], LiLiR [67], RM-3000 [79] and BBC-LRS2 [17] are examples of sentence-based audio-visual datasets that were used for training and validation and they contain videos different people speaking a variety of sentences. Accuracies in this field tend to be low for reasons discussed earlier such as the short duration of visemes, the limited number of datasets available with isolated visemes and the lack or research attention given to viseme classification generally in comparison to other speech segments. There doesn't appear to be a copious amount of literature devoted viseme-to-word conversion for other languages.

In one work by Fenghour et al. [181], a Long-Short Term Memory Network (LSTM) was used

that takes visemes as an input and predicts the words that were spoken by individuals from a limited dataset with some satisfactory results. This configuration pre-supposes that the identity of individual visemes are already known(hence the reason why several values in Table 3.1 are listed as N/A), so its robustness to misclassified visemes has not been verified. Moreover, the sentences that are predicted are often not grammatically correct in terms of syntax, and many sentences predicted incorrectly have a large grammatical uncertainty or entropy.

Table 3.1: Two-stage speech recognition approaches where CI and CD refer to context-independent and context-dependent models and SAT refers to speaker adaptive training.

Approach	**Viseme representation**	**1st Stage Feature Extractor**	**1st Stage Classifier**	**2nd Stage Classifier**	**Dataset**	**Unit Classification Accuracy(%)**	**Word Classification Accuracy(%)**
Lan and Harvey [182]	Bigram	LDA + PCA	HMM-GMM	HMM	LiLiR	45.67	14.08
Almajai [143]	Bigram	LDA HMM	HMM	HMM	LiLiR	-	17.74
Almajai [143]	Bigram	LDA+MLLT	HMM	HMM	LiLiR	-	22.82
Almajai [143]	Bigram	LDA+MLLT+SAT	HMM	HMM	LiLiR	-	37.71
Almajai [143]	Bigram	LDA+MLLT+SAT	HMM	Feed-forward	LiLiR	-	47.75
Bear and Harvey [31]	Bigram	Active Appearance Model	HMM	HMM	LiLiR	8.51	4.38
Thangthai [173]	Bigram	Discrete Cosine Transform	CD-GMM+SAT	WFST	TCD-TIMIT	42.48	10.47
Thangthai [173]	Bigram	Discrete Cosine Transform	CD-DNN	WFST	TCD-TIMIT	38.00	9.17
Thangthai [173]	Bigram	Eigenlips	CD-GMM+SAT	WFST	TCD-TIMIT	44.61	12.15
Thangthai [173]	Bigram	Eigenlips	CD-DNN	WFST	TCD-TIMIT	44.60	19.15
Howell [175]	Bigram	Active Appearance Model	CD-HMM	HMM	RM-3000	52.31	43.47
Fenghour [181]	Cluster	N/A	N/A	Encoder-Decoder LSTM	BBC-LRS2	N/A	72.20
Fenghour [159]	Cluster	ResNet CNN	Linear Transformer	GPT-Transformer based Iterator	BBC-LRS2	95.40	64.60

Lan and Harvey [182] classified words from decoded visemes using HMMs with a bigram language model to predict words once spoken visemes had been classified from videos of spoken sentences from the LiLiR corpus. Visemes were classified with an accuracy of 45.67% while the word accuracy achieved was 14.08%.

Thangthai et al. [173] decoded visual speech in the form of both visemes and phonemes for four different first-stage classification methods whilst using a WFST for the second-stage conversion when predicting spoken words. Every one of the four systems performed viseme classification with greater accuracy than phoneme classification, though a greater accuracy was observed at the second stage in the word conversion process because the efficiency in performing phoneme-to-word conversion was higher than that of the viseme-to-word conversion. The main reason for better results being achieved when using phonemes instead of visemes is that there will

be always be more ambiguity with the use of visemes as there are significantly more mapping options available [173].

It may seem inherent that the intermediate units to be modelled should be visemes when there is no audio available. However, the availability of context and good accuracy being attained at the first stage, would make the use of phonemes more preferable for the prediction of sentences than the use of visemes. The increased ambiguity that one has to overcome with the use of visemes as opposed to phonemes is due to there being far more words in the English language that share visemes than phonemes [175] [41] [176] [173] [142] meaning that there are significantly fewer mapping options to be considered when doing the conversion for word prediction.

It is for this reason that Howell et al. [175] prefer to use phonemes as the intermediate class. They acknowledge that even with perfect feature extraction and performance for first stage classification, the second stage conversion will always be limited because and HMM or WFST based conversion model will fail to predict semantic details when distinguishing between semantically different words like "Hepburn"/"Campbell", "barge"/"march", "six"/"since"; or syntactic details when there is confusion of plural and singular versions of a word that ends with a viseme corresponding to same for the letter /s/ e.g. "threat"/"threats" [175].

Almajai [143] experimented with three different methods of classifying visemes including Linear Discriminant Analysis(LDA) with a HMM, LDA with Maximum Likelihood Linear Transform(MLLT), and a LDA/MLLT/Speaker Adaptive Training(SAT) hybrid but it was the LDA+MLLT+SAT classifier recorded the best result for viseme classification. They then used two different algorithms, a HMM and a feed-forward neural network to do the word conversion and the feed-forward network was the better performing of the two recording an accuracy of 47.75% compared with the feed-forward network achieving 37.71%.

The lip reading system proposed by Fenghour et al. [159] used a viseme-to-word converter to match clusters of visemes to word combinations by iteratively combining words and calculating the perplexity scores. A Generative Pre-Training (GPT)-based transformer [183] is used to calculate perplexity scores of word combinations in order to determine the most likely combination of words given the clusters of visemes that are inputted. Perplexity is a measure of

grammatical correctness, so it is expected that the most likely combination of words to have been uttered given a set of visemes is the combination with the lowest perplexity score.

The model used by Fenghour et al. [159] matches clusters of visemes to words in a lexicon mapping and is contingent on visemes being classified correctly. Visemes being misclassified in one cluster will not only cause error in the word matching for that one word but will in turn cause compound errors in the combination process during the iterations due to conditional dependence of word combinations. This means that one word being misclassified can cause other words to also be misclassified as well.

Other important works in the field who have utilised visemes as part of a two-stage conversion process include Sterpu and Harte who used Discrete Cosine Transformation with an HMM to classify visemes with a HMM for the conversion [184]; as well as Peymanfard et al. who used neural network architecture consisting of a CNN frontend with an attention transformer backend to classify visemes and an attention-transformer to predict words [185]. The full results of the viseme classification for both these works has not been disclosed.

Many of the conversion models listed in Table 3.1 lack the discriminative power to be able to learn semantic and syntactic information needed to be able to distinguish between words that share identical visemes [175]. This is because of the lack of context available due to their fixed size context windows and to increase the size of the context window only increases the computational complexity of the model. Whereas the conversion model proposed here uses a GRU network which can exploit context from an unlimited number of timesteps regardless of the length of sentence which itself would not affect the model complexity.

As well as being effective at exploiting unlimited previous context to discriminate between words sharing identical visemes, the proposed conversion model is also robust to misclassified visemes because it can capture both long and short term dependencies unlike the conversion model used by Fenghour [159]. It is therefore less prone to cascading errors.

Another limitation of the word converter used in [159] is it's inefficiency. The best performing architectural model proposed here uses significantly less parameters and takes significantly less

time in executing the prediction of a spoken sentence for one viseme sequence.

The word converter proposed here has been trained on a large dataset with a wider range of vocabulary than the converter in Fenghour et al.'s work [181] which was only trained on words and sentence contained in the much smaller and limited TIMIT corpus [186]. A curriculum learning strategy like that of Chung et al. [181] is also used in the training phase to ensure that the network can better model natural language by predicting shorter N-grams. The proposed approach falls into the category of neural conversion models.

For a viseme-to-word converter to be accurate, it has to be effective in classifying words from visemes that have been classified correctly but also be robust to the possibility of visemes that have not been classified correctly. This section is devoted to presenting theoretical justifications for why the proposed approach is more effective in addressing these scenarios than other conversion methods.

It is apparent that the majority of viseme-to-word converters that used either HMMs, WFSTs or even Feed-forward networks were ineffective with a low conversion performance and the reason for this is because such models are unable to use enough context to disambiguate commonly confused words that share visemes. In subsection 3.4.3, an explanation is given as to why the attention-based GRU model [187] proposed is more effective in discriminating between words sharing identical visemes and it is because they are able to use more contextual information in extracting lexical rules to learn the syntactic and semantic differences between words.

The GPT-based iterator used in [159] that predicts word using perplexity calculations has a very high conversion performance for correctly visemes, but it is nonetheless highly susceptible to the presence of incorrectly classified visemes. Incorrectly classified visemes leading wrong predictions which in turn causes error propagation in the predicted word sequence of the outputted sentences. This section demonstrated how the GRU model proposed here is less susceptible to the impact of incorrectly classified visemes because of its ability to model both short and long term dependencies.

3.4.3 Syntactic and Semantic Disambiguation

Given the visemes decoded, a language model is required to determine the most probable combination of words to have been uttered and the language model has to be robust to the possibility of either visemes being misclassified. According to Eq. 3.10, for a set of given visemes V, a language model will predict the most likely set of words W^* to have been uttered for different combinations of words W [188]. Table 3.2 gives an example of a set of visemes and the words that most likely correspond those visemes.

$$W^* = \arg\max_W P(V|W)P(W) \tag{3.10}$$

Table 3.2: A sequence of visemes and its corresponding word match.

Visemes	<sos>	'T'	'AH'	<space>	'T'	'ER'	'P'	'W'	'AH'	'T'	<space>	'W'	'AA'	'T'	<eos>
Words	<sos>	"THE"			"SURPRISE"							"WAS"			<eos>

Classifiers with language models that have been previously used for viseme-to-word conversion such as N-grams have been ineffective due to the algorithms' inability to discriminate between words that share visemes but are different either syntactically or semantically. An example of syntactically distinct words sharing identical visemes would be the case of plural and singular versions of a word that end with a viseme corresponding to a consonant with same viseme as that for the letter /s/. Examples of semantically distinct words sharing identical visemes being confused are those words that have an identical likelihood of being preceded by a common word in a bigram [175].

If the context window is long enough, it can capture the subject-verb agreement which is a grammar rule that can be used to determine if a noun is singular or plural and thus address the problem of syntactic disambiguation. The subject-verb agreement is a situation whereby the status of a noun subject being singular or plural can be determined by the form of the verb. If one takes the sentence "the keys to the cabinet are on the table" as an example; the word "keys" is an agreement with the word "are" and if enough context is captured, the correct syntactic form of the noun can be determined [189]. Figure 3.2 shows the syntax tree.

To maximise the probability of distinguishing between words that are syntactically different, one would need to utilise context either side of the subject noun meaning that both left and right context [190] are required and unidirectional RNNs will only be able to exploit left context. Bidirectional RNNs can exploit left and right context, however a bidirectional network uses twice the number of parameters and more computational overhead to train and evaluate.

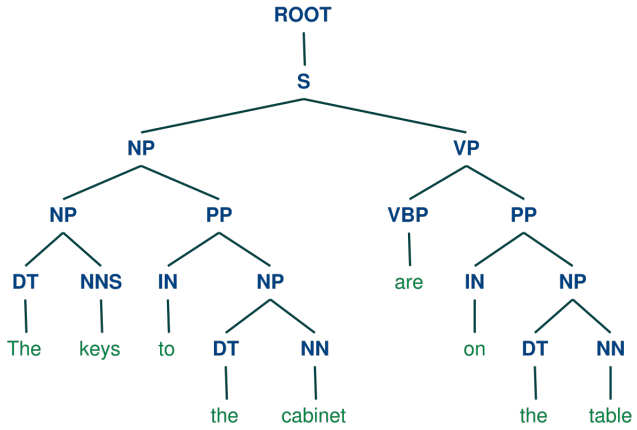


Figure 3.2: Syntax tree for the sentence "the keys to the cabinet are on the table".

Intuitively, one can conclude that having access to greater context gives language models more discriminative power and would help to differentiate between words that are semantically different. A noun can be disambiguated through relationship analysis. Noun phrases will follow different patterns that are characterised between the different types of words that noun phrases contain. The phrase categories can be narrowed to adjective-noun phrases, verb-adjective-noun phrases and subject-verb-object phrases [191]. The identity of a noun can be determined by the adjective describing it or the verb action it is performing it and the more context there is available for a language model, the greater the probability of it being predicted.

3.5 Proposed Lip-Reading System

One of the main contributions being made in this thesis is the proposition of a lip-reading system that decodes entire sentences using solely visual cues from videos of people speaking in real-time whereby visemes are used as the classification schema as opposed to ASCII characters.

To do this, the proposed system predicts speech in two stages with two main components.

The first main component classifies visemes to address the question "Can a good classification performance of individual visemes be attained?", while the next component predicts sentences using a model to predict spoken words from those decoded visemes to address the question "Can a language model be implemented that is effective at converting visemes to words?". The proposed system consists of a number of components:

1. Viseme Classifier
    (a) Preprocessing
    (b) Visual FrontEnd
    (c) Classifier Backend
2. Viseme-to-Word Converter

Speech recognition can be performed by classifying individual lip movements and then mapping these lip movements to possible spoken words whereby the most probable combination of words mapping to these lip movements is outputted as the decoded result.

Visemes are the most fundamental units of visual speech and the work reported in this thesis starts off with classification of individual isolated visemes from a limited number of different speakers. In real time however, visemes would be uttered continuously whereby the boundary of where a viseme starts and stops would be unknown so it was necessary to be able to decode visemes in continuous speech.

To decode speech for a person speaking in real time, it would be necessary to predict the words that had been spoken if the identity of the spoken visemes had been decoded. There is a bottleneck that needs to be overcome when predicting words having decoded the spoken visemes in that because multiple phonemes share identical visemes, one set visemes can correspond to different possible combinations of words having been spoken. A language model, i.e., a probability distribution over sequences of words must be used in performing the viseme-to-word conversion; and it must be effective in:

1. Disambiguating between words that share identical visemes
2. Be robust to the possibility of visemes having been decoded incorrectly

Figure 3.3 shows an outline of the different components that make up the proposed lip reading system. Videos are sampled into image frame and some preprocessing steps are implement.

Part of the preprocessing involves locating the region-of-interest i.e. the lips within the image frames and cropping them to leave only the region-of-interest.

The Visual Frontend is the component used for feature extraction where lip pixels are converted into a lower dimensional and the Viseme Classifier classifies visemes based on the extracted features. Once visemes have been decoded, the viseme-to-word detector classifies uses a language model to classify the spoken words.

The proposed lip reading system utilises a ResNet architecture for Visual Frontend like that of [105] and an Attention-Transformer similar to that of Vaswani et al. [137] except that the decoder has been modified with Multi-Perceptron layers to take the form of a Linear Decoder. Initially, a GPT transformer was used to predict sentences by matching words to visemes and determining to most likely combination using perplexity [192] scores but this methodology was then replaced with an attention-based GRU that directly predicts words based in the inputted visemes.

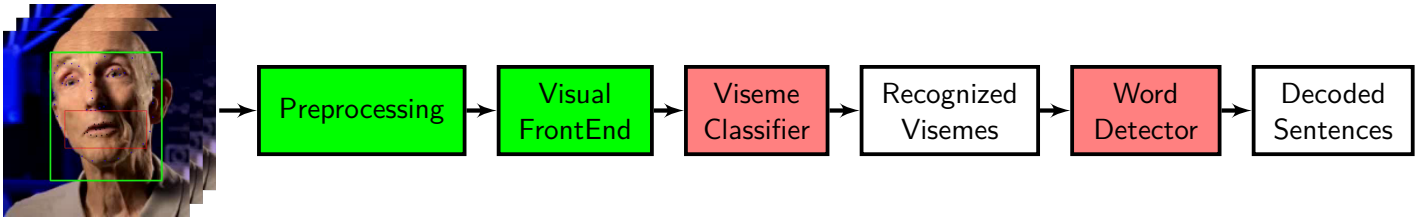


Figure 3.3: An overview of the proposed lip reading system.

3.6 Summary

The Literature Review has helped to identify knowledge gaps in existing research with regards to speech data. Lip-reading systems will need to predict entire sentences with good accuracies covering ever expanding vocabularies and so it would help to devise a lip-reading system that is lexicon-free as lip-reading systems are becoming more generalized in predicting sentences with thousands of different possible words.

A variety of different classification schema have been deployed where earlier classification networks encoded single words as a class and later networks have used ASCII characters to predict

sentences covering huge lexicons. In theory, the use of phonemes and visemes could mean that lip-reading systems could be lexicon-free whereby a lip-reading system could predict a word spoken by an individual that did not appear in the training phase.

Other challenges inhibiting the progress of automated lip-reading still remain. These include the need to predict unseen words, i.e. predict spoken words that did not appear in training phase and are not covered by the lexicon as well as visual ambiguities where the semantic and syntactic features of words can be learned for words that look the same when spoken. From a visual perspective, there remains challenges such as speaker dependency, especially when attempting to generalise to speakers who have not appeared in the training data; the need to generalise to videos of varying spatial resolution and the need to generalise to videos of different frame rates while consisting of varying quantities of temporal data.

The conclusions of this literature review has informed the focus of research in this thesis with a new classification schema being used as part of the proposed lip-reading system in the form of visemes with advantages and disadvantages to using visemes having been discussed. With the emergence of large-scale databases and lip-reading systems covering vocabularies with thousands of different words, there is a need to develop a speech recognition system that is lexicon-free and not to constrained to a fixed-list of vocabulary. Even the ability to predict sentences as a two-stage procedure regardless of whether one uses visemes or ASCII characters as the intermediate stage has merit in that one can ensure that the sentences being predicted are grammatically correct.

The key findings of the literature review are the following:

- Lip-reading systems have moved towards the use of deep learning for both feature extraction and classification due to both advances in networks and the availability of large-scale databases
- Lip-reading systems tasked for word classification where every single word is encoded as a class have attained very good performances
- Lip-reading systems tasked for predicting good accuracies for entire sentences which cover

entire vocabularies have failed to attain good performances

- The majority of lip-reading systems which are designed to predict sentences do so using ASCII characters as a classification schema
- The use of alternative possible classification schema to predict entire sentences is not something that has been given much research attention
- The use of ASCII characters as class do have limitations including the large number of classes required, the need to have been pre-trained on a lexicon to have good word coverage and the need for curriculum learning strategies.
- Visemes are an alternative classification scheme with the following advantages: they use fewer classes than visemes, can be generalised to predict speech from people speaking different languages and they do not need pre-trained lexicons
- Visemes however do suffer from the bottleneck not only is there one-to-many mapping relationship between visemes and words, a small drop in viseme classification performance significantly affects the word prediction performance
- To predict words by classifying visemes requires both a good viseme classification performance and an efficient viseme-to-word conversion performance
- The viseme-to-word conversion not only requires a language model that is effective in using semantic and syntactic information to accurately predict words but must also be robust to the possibility

Chapter 4

Sentence Prediction using Visual Cues

This Chapter addresses Research Question 6: *"Can a good overall performance be attained for word classification when predicting sentences?"* and presents a neural network-based lip reading system that uses visemes as a classification schema. This chapter also addresses the question *"Can a good classification performance of individual visemes be attained?"* fulfilling all of the relevant criteria including the need to classify visemes from profile and frontal views, the need to perform temporal alignment for visemes of varying duration where the start and stop time is unknown, and the need to have good generalisation capabilities.

The prediction of sentences as a two-stage procedures with visemes being classified in the first stage and words being predicted in the second stage raises the question *"Can a language model be implemented that is effective at converting visemes to words?"*. For words that have a unique set of visemes(approximately half of words in the English language), the classification performance of these words must in theory be perfect. For homopheme words, the conversion model bust be effective at disambiguating between words that share identical visemes. The speech recognition model reported in this Chapter attains good word accuracies for both words with unique sets of visemes and homopheme words.

This Chapter is organised as follows: First in Section 4.1 is the Chapter Introduction, then in Section 4.2, details of all the components that make up the whole lip reading system including pre-processing, visual feature extraction, viseme classification and word detection are given.

In Section 4.3, the classification results for the overall lip reading system are discussed and compared followed by concluding remarks given in Section 4.4 along with suggestions for further research.

4.1 Introduction

This chapter focuses on improving the accuracy of lip reading sentences and this is achieved by using visemes as a very limited number of classes for classification, a specially designed deep learning model with its own network topology for classifying visemes, and a conversion of recognised visemes to possible words using perplexity analysis.

Using visemes for lip reading sentences has some unique advantages. The use of visemes as classes in comparison to the use of either words or ASCII characters as classes requires an overall smaller number of classes which alleviates bottleneck in the computation. In addition, using visemes does not require pre-trained lexicons, meaning that a viseme-based lip reading system can be used to classify words that have not presented in the training phase, and they can be generalised to different languages because many different languages share the same visemes.

On the other hand, there are some specific issues to be considered when designing a viseme-based lip reading system for sentences. The general classification performance for individual segmented visemes has been less satisfactory in comparison to the classification of words due to the fact that visemes tend to have a shorter duration than words. This results in there being less temporal information available to distinguish between different classes, as well as there being more visual ambiguity when it comes to class recognition [18]. One possible way to address this problem is to significantly increase the training data available to enhance the system's ability to distinguish between classes, and this is why a high volume of training videos have been utilised. Moreover, there is a direct conversion of recognised ASCII characters to possible words in a one-to-one mapping relationship, whereas this one-to-one mapping relationship does not exist when using visemes, because one set of visemes can map to multiple different sounds or phonemes. This also means that once visemes have been classified, there is still the need to perform a

viseme-to-word conversion. This approach also helps to distinguish between homopheme words or words that look the same when spoken but sound different [193], a phenomenon that exists because of the one-to-many mapping relationship between visemes and phonemes.

The proposed automated lip reading system contains a component to classify spoken visemes from people speaking in silent videos, and a component to perform viseme-to-word conversions using perplexity analysis [192]. The proposed model also has a good robustness to varying levels of lighting.

4.2 Proposed Approach for Sentence Prediction

Given a silent video of a talking face, the objective here is to predict the sentences being spoken by extracting their lip movements. In this Section, an overall architecture is proposed for decoding visual speech illustrated in Chapter , Figure 3.3. The entire process consists of different stages, starting off with a Data Preprocessing stage where the region of interest is extracted from the videos using facial landmark detection to provide the input to the Visual Frontend. The components of the overall architecture include: a spatial-temporal visual frontend that inputs a sequence of images of loosely cropped lip regions, and outputs one feature vector per frame; a sequence processing module known as the viseme classifier that inputs the sequence of per-frame feature vectors and outputs a sequence of visemes, and finally a module that matches visemes to words and predicts the uttered sentence using perplexity analysis. The performance of the system is evaluated by comparing the sentences predicted by the lip reading system to the ground truth of the spoken sentences and measuring the edit distance. In the following Sections, details of the systems components are discussed.

4.2.1 Architecture

The overall system used for decoding speech consists of two separate neural network architectures used to perform two different tasks. The first architecture is used for the task of viseme

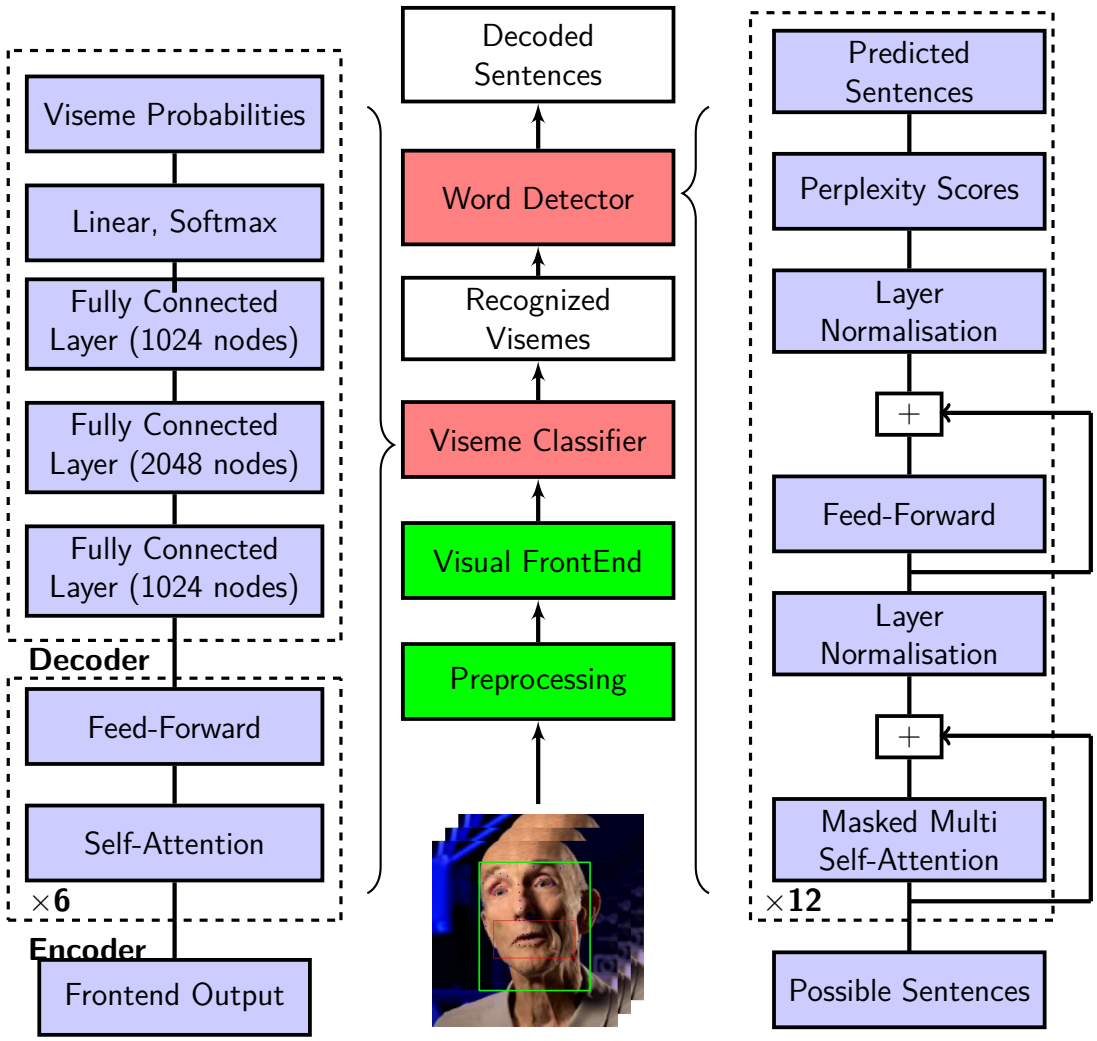


Figure 4.1: The breakdown stages of how sentences are predicted from silent videos.

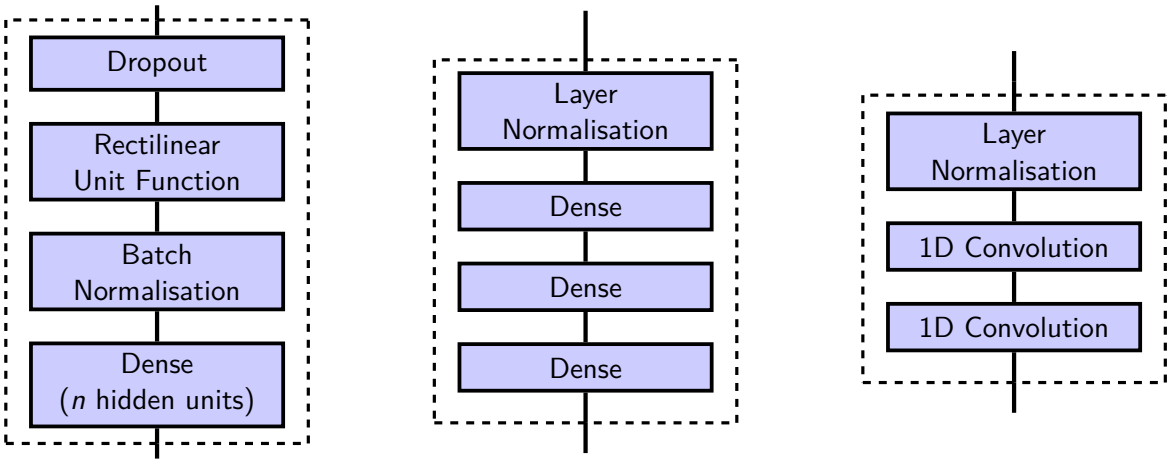
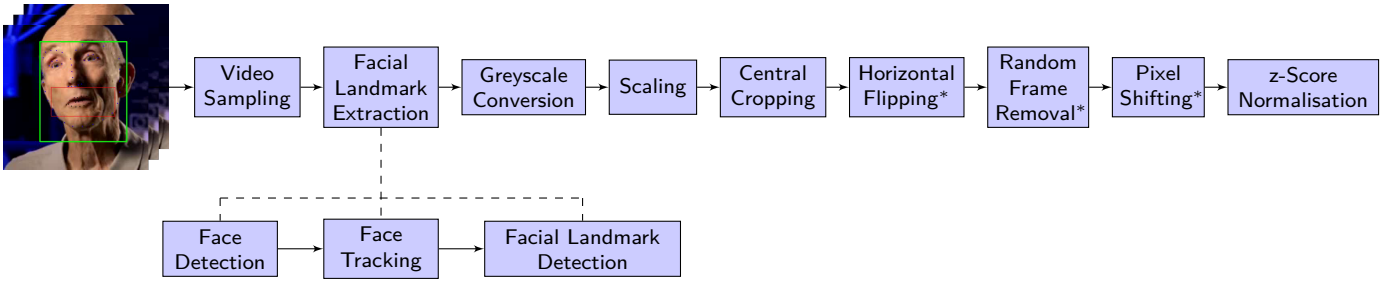


Figure 4.2: The different transformer components with the fully connected layer on the **left**, self-attention in the **middle** and feed-forward on the **right**.



* Only performed during the training of the Visual Frontend.

Figure 4.3: The stages of video image pre-processing.

classification and consists of a spatial-temporal visual frontend in tandem with an attention-based transformer and the predicted visemes provide the input of the next architecture. The second architecture, also an attention-based transformer, is used to predict the spoken words given the uttered visemes using a calculated metric called perplexity. As illustrated in Figure 4.1, each of these modules are briefly described along with the overall framework for the lip reading system. Both the viseme classifier and the word detector consist of common blocks including fully connected layers, self-attention layers and feed-forward layers and the breakdown of these three blocks is given in Figure 4.2.

The attention-transformer structure used in [137] has been changed to fit visemes, and this will be discussed in 4.2.5. Unlike [137], there is no embedding layer, and the Decoder has been altered with the final softmax layer trained on visemes instead of ASCII characters.

4.2.2 Data

The dataset used in this research is the BBC LRS2 dataset [17]. It consists of approximately 46,000 videos covering over 2 million word instances and a vocabulary range of over 40,000 words. The video with the longest duration has a length of 180 frames with every video have frame rate of 25 frames per second. The dataset contains sentences of up to 100 ASCII characters from BBC videos, with a range of facial poses from frontal to profile. The dataset is extremely difficult due to the variety of viewpoints, lighting conditions, genres and the number of speakers.

Table 4.1 gives a breakdown of the different sections of the BBC LRS2 data with statistics of how many sentences there are, the number of word instances, the vocabulary range and the ratio of profile to frontal videos in that particular section of the corpus.

Table 4.1: Statistics of BBC LRS2 dataset.

Split	**Utterances**	**Word Instances**	**Vocabulary**	**Frontal/Profile Split (%)**
Train	45839	329180	17660	64.8:35.2
Test	1243	6660	1697	63.5:36.5

4.2.3 Data Pre-processing

All the videos are pre-processed according to the stages given in Figure 2.7. Videos consist of images with red, green and blue pixel values and resolution 160 pixels by 160 pixels; with a frame rate of 25 frames/second. Videos are first sampled into image frames, then once the videos are sampled, facial landmarks need to be located as the speaking person's lips are the region of interest and feature input to the visual frontend. The Single Shot MultiBox Detector (SSD) [200], a CNN-based detector, is used for detecting face appearances within the individual frames and to recognise facial landmarks according to the iBug [201] landmark convention of 68 landmarks, and it can be used on faces pointing at different angles. Landmarks are applied according to the stages shown in Figure **??** with the face detected shown on the left, the face being tracked in the middle and where facial landmarks are detected on the right.

The video frames are then converted to greyscale, scaled, and then centrally cropped around the boundary of the facial landmarks resulting in reduced image dimensions of $112 \times 112 \times T$ dimensions (where T corresponds to the number of image frames). Data augmentation in the form of horizontal flipping, removal of random frames [202, 203], and random shifts of up to ± 5 pixels in the spatial dimension and ± 2 frames in the temporal dimension respectively, respectively, are also applied. At the end, pixels are normalized with respect to the overall mean and variance of every pixel in each frame.

Pre-processing is needed in order to ensure that the appropriate region of interest (ROI) can be extracted as the input to the neural network with resolution 112×112 pixels that contains

the lips. The ROI must also undergo greyscale conversion and z-score normalization. The facial landmark detection described earlier has already been performed on every single video contained within the BBC LRS2 corpus. Some of the pre-processing steps described in Figure 4 may not be necessary for this corpus, as the 112×112 set of pixels can be extracted through central cropping of the original image frames with 160×160 pixels. The entire pre-processing process would however be a necessity for a lip reading system that can be generalized to other real-time applications.

4.2.4 Visual Frontend

The spatial-temporal visual frontend is based on [203]. The network applies a spatial-temporal (3D) convolution on the input image sequence, with a filter depth of five frames, followed by a 2D ResNet(composed of convolutional 2D layers) that gradually decreases the spatial dimensions with depth. For an input sequence of $T \times H \times W$ frames, the output is a $T \times \frac{H}{32} \times \frac{W}{32} \times 512$ tensor (i.e., the temporal resolution is preserved) and it is then average-pooled over the spatial dimensions, yielding a 512-dimensional feature vector for every input video frame. Details of the architecture for the Visual Frontend are given in Table 4.2 where the output dimensions of each layer are given along with the filter dimensions and stride width($\times 2$ refers to the number of filters). Weights from the trained Visual Frontend network used in [105] has been applied in this work and the Frontend used her is identical to that of [105].

Table 4.2: Details of spatial-temporal network for visual frontend.

Layer Type	**Filter**	**Output Dimensions**
3D Convolution	$[5 \times 7 \times 7, 64]/(1,2,2)$	$180 \times 56 \times 56 \times 64$
3D Max Pooling	(1,2,2)	$180 \times 28 \times 28 \times 64$
Residual 2D Convolution	$[3 \times 3, 64] \times 2/(1, 1)$	$180 \times 28 \times 28 \times 64$
Residual 2D Convolution	$[3 \times 3, 64] \times 2/(1, 1)$	$180 \times 28 \times 28 \times 64$
Residual 2D Convolution	$[3 \times 3, 128] \times 2/(2, 2)$	$180 \times 14 \times 14 \times 128$
Residual 2D Convolution	$[3 \times 3, 128] \times 2/(1, 1)$	$180 \times 14 \times 14 \times 128$
Residual 2D Convolution	$[3 \times 3, 256] \times 2/(2, 2)$	$180 \times 7 \times 7 \times 256$
Residual 2D Convolution	$[3 \times 3, 256] \times 2/(1, 1)$	$180 \times 7 \times 7 \times 256$
Residual 2D Convolution	$[3 \times 3, 512] \times 2/(2, 2)$	$180 \times 4 \times 4 \times 512$
Residual 2D Convolution	$[3 \times 3, 512] \times 2/(1, 1)$	$180 \times 4 \times 4 \times 512$

4.2.5 Viseme Classifier

Lip reading datasets consist of labels in the form of subtitles. These subtitles are strings of words that need to be converted to sequences of visemes to provide labels for the viseme classifier. The conversion is performed in two stages: first, they are mapped to phonemes using the Carnegie Mellon Pronouncing Dictionary [204], and then the phonemes are mapped to visemes according to Lee and Yook's approach [35]. The mapping used can be found in Subsection 2.2 of this thesis. The attention transformer which predicts the spoken visemes from a person speaking in a silent video uses 17 classes in total; these include the 13 visemes, a space character, start of sentence (SoS), end of sentence (EoS) and a character for padding. All the defined classes are listed in Table 4.3. All videos are padded to 180 characters.

The Transformer [137] model has an encoder-decoder structure with multi-head attention layers used as building blocks. The encoder used is a stack of self-attention layers, where the input tensor serves as the attention queries, keys and values at the same time. The decoder here consists of 3 fully connected layer blocks structured as shown in Figure 4.4; and each fully connected layer blocks consists of a dense layer, batch normalisation, rectilinear unit function and a dropout layer of probability 0.1. The dense layer within the middle fully connected layers consists of 2048 nodes while the dense layers within the first and last fully connected layer blocks only contain 1024 nodes. The decoder produces viseme probabilities which are directly matched to the ground truth labels and trained with a cross-entropy loss. The encoder follows the base model of [137] with 6 layers, model size 512, 8 attention heads and dropout with probability 0.1.

However, it should be noted that the decoder utilised in this work follows a completely different structure from that of [105] for the following reasons:

1. There are no embeddings;

2. The predicted labels from the previous timestep are not fed into the decoder as it is assumed that visemes do not have the conditional probability relationship that ASCII

characters have. This means that no teacher forcing is used whereby the ground truth of the previous decoding step has to be supplied as the input to the decoder.; and

3. It is only the decoder and dense layer that differ, so the trained weights from [105] have been used and applied to both the visual frontend and encoder, where only the decoder layers and dense layers are trained.

Table 4.3: Classes used by Viseme Classifier.

[pad], AA, AH, AO, CH, ER, EY, F, IY, K, P, T, UH, W, <sos>, <eos>, [space]

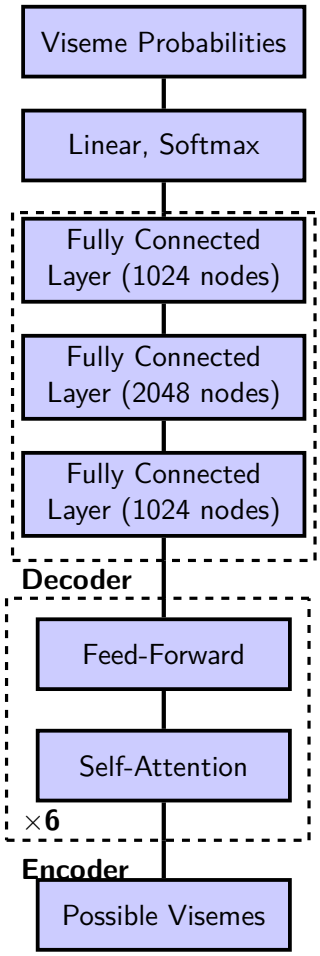


Figure 4.4: The architecture of transformer for the Viseme Classifier.

Because the encoder has an identical topology to that used by [105], the trained weights from their model have been applied to here and it is only the decoder and the final softmax layer in Figure 4.4 that are to be trained. During the training phase, the Adam optimiser [205] is used with default parameters and initial learning rate 10^{-3}, reducing it on plateau down to 10^{-4} and

all operations are implemented in TensorFlow and trained on a single GeForce GTX 1080 Ti GPU with 11GB memory.

4.2.6 Word Detector

The outputted visemes from the viseme classifier need to be further converted to meaningful sentences or strings of words. Every word in a sentence contains a set of visemes and therefore can be mapped to a cluster of visemes, such that a cluster of visemes is a set of visemes which make up a word. Once visemes have been classified, the viseme-to-word conversion process needs to be performed. Because a cluster of visemes can map to several different words, the combination of the words that were uttered by the speaker still needs to be deciphered. The solution to the problem is to select the most likely combination of words. The general procedure for converting visemes to words with different stages is given in Figure 4.5.

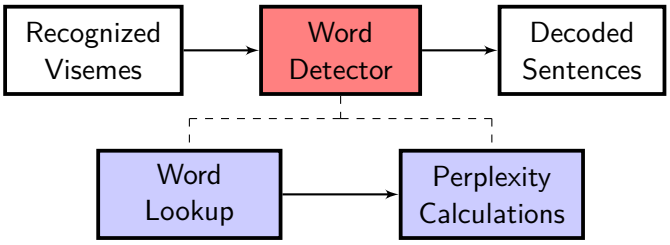


Figure 4.5: The components of the Word Detector.

The first stage of the Word Detection is the World Lookup stage. Every single cluster of visemes needs to be mapped to a set of words containing those visemes according to the mapping given by the Carnegie Mellon Pronouncing (CMU) Dictionary. However, if there are clusters where no match is found, a cluster in the dictionary that most closely resembles it is used instead and the words mapping to that cluster are used. The resemblance is determined using Levenshtein distance [206] and the cluster in the CMU dictionary with the smallest value is chosen.

Once the word lookup stage is performed, the next stage of Word Detection is the Perplexity Calculations. The different possible choices of words that map to the visemes are combined, and iterations are performed to determine which combination of words is most likely to correspond to the uttered sentence, given the visemes recognised. Naturally, the sentence that is most

grammatically correct will have the highest likelihood [207] and perplexity is one metric that can be used to compare sentences to determine which is most grammatically sound. The rationale behind perplexity is discussed later with an even more detailed description about how perplexity analysis is used to convert viseme to words. The following rules are used when predicting sentences and they are based on determining which combinations of words have the greatest likelihood according to probabilistic information theory:

1. If a viseme sequence has only 1 cluster matching to one word, that one word is selected as the output.
2. If a viseme sequence has only 1 cluster matching to several words, that word with largest expectation is selected as the output.
3. If a viseme sequence has more than 1 cluster, the words matching to the first two clusters are combined in every possible combination for the first iteration.
   (a) The combinations with the lowest 50 perplexity scores are kept.
   (b) These combinations are in turn combined with the words matching to the next viseme cluster.
   (c) The combinations with the lowest 50 perplexity scores are kept and the iterations continue for the remaining clusters of the sequence until the end of the sequence is reached.

The selection of the lowest 50 perplexity scores at each iteration is based on an implementation of a local beam search with width 50. In practice, it would be computationally expensive to do an exhaustive search so a beam search has been implemented to reduce the computational overhead, and the beam width is an arbitrary figure chosen as a compromise between accuracy and computational efficiency.

Eqs. 4.1 to 4.4 below describe the probabilistic relationship between the observed visemes and the words spoken; where V is the spoken sequence of viseme clusters, v_i corresponds to every ith cluster, W_C represents any given combination of words and w_i corresponds to every ith

word within the string of words. The string of words $\check{W}$ that is to be selected will be the combination that has the maximum likelihood given the identity of the viseme clusters for every combination C that falls within the set of combinations C^*. The sequence of visemes clusters given in Eq. 4.1 maps to any possible combination of words as given in Eq. 4.2, and the solution to predicting the sentence spoken is the combination of words given the recognised visemes which has the greatest probability as expressed in Eqs. 4.3 and 4.4.

$$V = (v_1, v_2, ..., v_N) = \sum_{i=1}^{N} v_i \tag{4.1}$$

$$W_C = (w_1, w_2, ..., w_N)_C = \sum_{i=1}^{N} w_i \tag{4.2}$$

$$\check{W} == \arg\max_{C \epsilon C^*} [P(W|V)]_C \tag{4.3}$$

$$\check{w}_1, \check{w}_2, ..., \check{w}_N = \arg\max_{C \epsilon C^*} [P(w_1, w_2, ..., w_N | v_1, v_2, ..., v_N)]_C \tag{4.4}$$

If the identity of observed visemes is known, the probability of the viseme sequence in Eq. 4.1 is equal to 1, resulting in the expression in Eq. 4.5. The choice of words predicted according to Eq. 4.4 gets reduced to the expression given in Eq. 4.6.

$$P(v_1, v_2, \ldots, v_N) = 1 \tag{4.5}$$

$$\check{w}_1, \check{w}_2, ..., \check{w}_N = \arg\max_{C \epsilon C^*} [P(w_1, w_2, ..., w_N)]_C \tag{4.6}$$

Eqs. 4.7 to 4.10 below describe the relationship between the perplexity PP, entropy H and probability $P(w_1, w_2, ..., w_N)$ of a particular sequence of N words $(w_1, w_2, ..., w_N)$. The word detector consists of a trained attention-based transformer for calculating PP expressed as the exponentiation of H in Eq. 4.7. The per-word entropy $\hat{H}$ is related to the probability $P(w_1, w_2, ..., w_N)$ of words $(w_1, w_2, ..., w_N)$ belonging to a vocabulary set W, and is calculated as a summation over all possible sequences of words. If the source is ergodic, the expression for $\hat{H}$ in Eq. 4.8 gets reduced to that in Eq. 4.9(ergodicity can be assumed on the basis that a language model can be used even if it has not been exposed to every single possible word

that has ever been spoken). The value of $P(w_1, w_2, ..., w_N)$ resulting in the choice of words selected as the output for Eq. 4.6 also results in the minimisation of entropy in Eq. 4.9, further resulting in the minimisation of perplexity given in Eq. 4.10.

$$PP = e^H \tag{4.7}$$

$$\hat{H} = -\lim_{N\to\infty} \frac{1}{N} \sum_{w_1,w_2,...w_N} P(w_1, w_2, ..., w_N) \ln P(w_1, w_2, ..., w_N) \tag{4.8}$$

$$\hat{H} = -\frac{1}{N} \ln P(w_1, w_2, ..., w_N) \tag{4.9}$$

$$PP = P(w_1, w_2, ..., w_N)^{-\frac{1}{N}} \tag{4.10}$$

A language model, i.e., a probability distribution over sequences of words, can be measured on the basis of the entropy of its output from the field of information theory [208]. Perplexity is a measure of the quality of a language model, because a good language model will generate sequences of words with a larger probability of occurrence resulting in a smaller perplexity.

The Transformer model used for the word detector is the pre-trained Generative Pre-Training (GPT) Transformer [183] - a multi-layer decoder and a variant of the transformer used in [137]. It consists of repeated blocks of multi-headed self-attention followed by position-wise feedforward layers. The architecture is typically used for sentence prediction; however, the architecture itself here is not used for direct classification, rather its purpose is for perplexity calculations that are required for word selection where visemes are converted to words. Visemes from the previous step are sequentially matched to words and the most probable sentence is chosen according to that with the minimum perplexity score. The perplexity score is calculated by taking the exponentiation of the cross-entropy loss when the GPT is evaluated on a sentence and like in [27], a beam width of 50 has been used.

4.2.7 Illumination

To test the proposed lip reading system's robustness to changes in lighting, the overall architecture, once trained, has been evaluated on videos from the testing set under levels of illumination. Illumination has been applied by varying the pixel brightness. It is after the video sampling stage of the pre-processing described in 4.2.3 that illumination is applied to the image frames. The overall process is described in Figure 4.6.

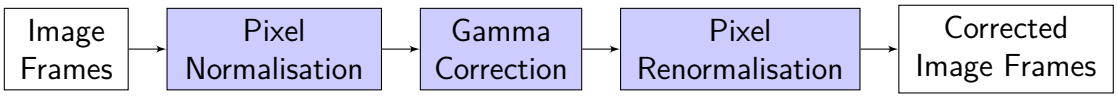


Figure 4.6: Stages for applying illumination.

Image frames of videos from the dataset consist of red, blue and green pixel components with numerical values ranging from minimum intensity 0 to maximum intensity 255. Pixel normalisation is the first stage of the procedure and this involves minimum-maximum normalisation of all pixel values where pixel values are mapped from the range [0,255] to [0,1]. Once this is done, a gamma correction is applied where pixel values are corrected according to Eq. 4.11, where I is a matrix of pixels, γ is scalar value and O is the resulting matrix of pixels after the gamma correction has been applied:

$$O = I^{1/\gamma} \tag{4.11}$$

Values of γ that are less than 1.0 will cause images to darken whereas values of γ that are greater than 1.0 cause images to brighten. Figure 4.7 gives examples of images with the standard image ($\gamma = 1.0$) on the left, the darkened image in the middle ($\gamma = 0.5$) and the brightened image on the right ($\gamma = 1.5$). The gamma corrections applied have utilised γ values ranging from 0.5 to 1.5.

After applying the gamma correction, pixels undergo re-normalisation where all pixels values are mapped back from from the range [0,1] to the range [0,255].

Figure 4.7: Images under varying illumination with standard image on the **left**, darkened image in the **middle** and brightened image on the **right**.

4.3 Experiments and Results

For training and evaluation of the viseme classifier, the BBC LRS2 dataset described in 4.2.2 has been used with 45839 sentences for training and 1243 sentences for testing. All components of the model are evaluated on the LRS2 test set. The metrics reported include VER, CER, WER, SAR and the total overall training time.

The viseme classifier was trained for a total of 2000 epochs and it was at the point that the validation loss(loss function evaluated on the test set) started to become saturated, and when no further convergence was recorded that the model was evaluated. Plots for the loss and VER for both training and validation are given in Figures 4.8 and 4.9.

The results are summarized in Table 4.4. As shown in the Table, the overall WER of 35.4% is a reduction of almost 15% compared to the 50% achieved in a previous state-of-the-art model trained and evaluated on the same dataset; and thus, improvement on the overall word accuracy to 64.6%. The accuracy by visemes was also very high with a VER of only 4.6%. The confusion matrices by both visemes and ASCII characters are given in Figures 4.10 and 4.11, respectively.

Table 4.5 gives the performance metrics for how the proposed lip reading system and Afouras et al's model [105] performed when videos in the validation set were subjected to different levels of illumination, applied to in accordance with 4.2.7. It can be seen that the proposed lip reading system is generally robust to varying levels of illumination, like that of Afouras et al [105] and this is expected given that videos in the BBC LRS2 corpus were recorded in varying lighting conditions.

In order to attain a good overall accuracy for classification of words, both the viseme classifica-

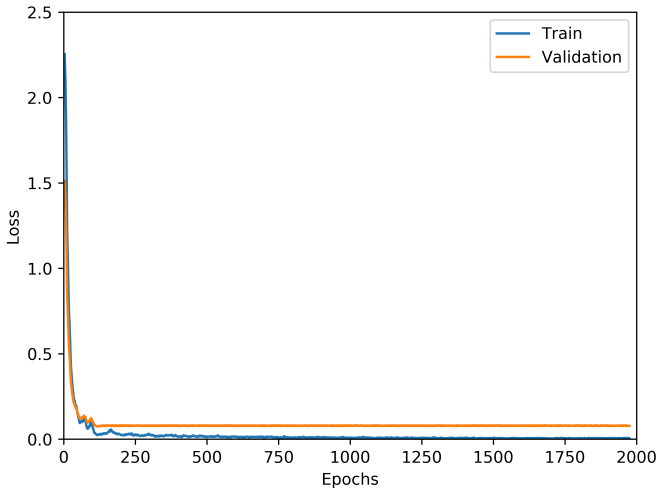


Figure 4.8: Loss curve for training and validation.

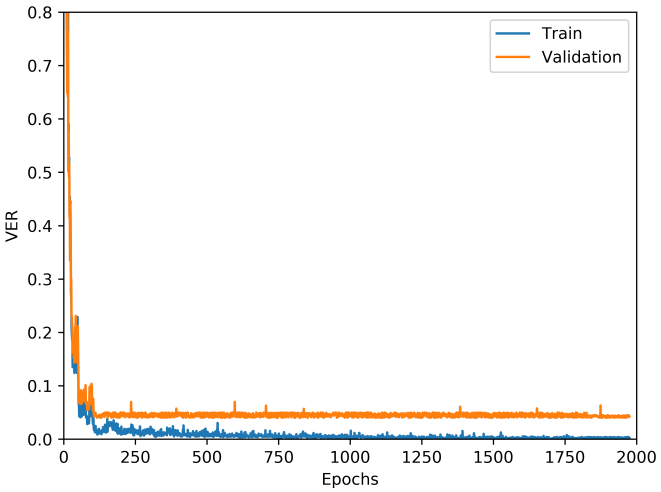


Figure 4.9: VER curve for training and validation.

Table 4.4: The performance results of lip reading sentences.

Validation Samples	**Parameters**	**VER(%)**	**CER(%)**	**WER(%)**	**SAR(%)**	**CPU Time**
1243	4,748,305	4.6	23.1	35.4	33.4	37 hours

Table 4.5: The performance of proposed system under varying illumination.

Gamma	**Visual Lip Reading System**			**Afouras et al.**		
	VER(%)	**WER(%)**	**SAR(%)**	**CER(%)**	**WER(%)**	**SAR(%)**
0.5	5.4	41.5	21.8	35.8	53.9	18.4
0.8	5.0	37.9	28.5	33.9	51.0	20.3
0.9	4.7	35.7	32.7	33.7	50.9	20.6
1	4.6	35.4	33.4	33.7	50.8	20.8
1.1	4.7	35.6	32.9	33.7	50.8	20.2
1.2	4.9	37.4	29.4	34.1	51.4	20.6
1.5	5.3	40.5	23.7	36.2	51.4	20.2

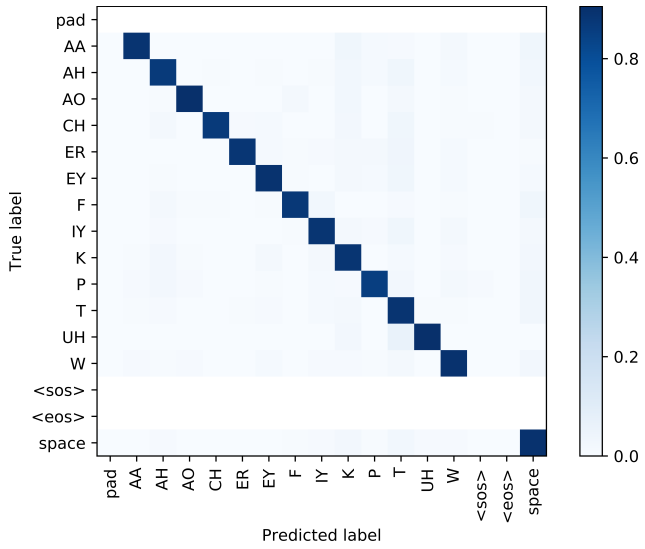


Figure 4.10: Confusion matrix for classification of visemes.

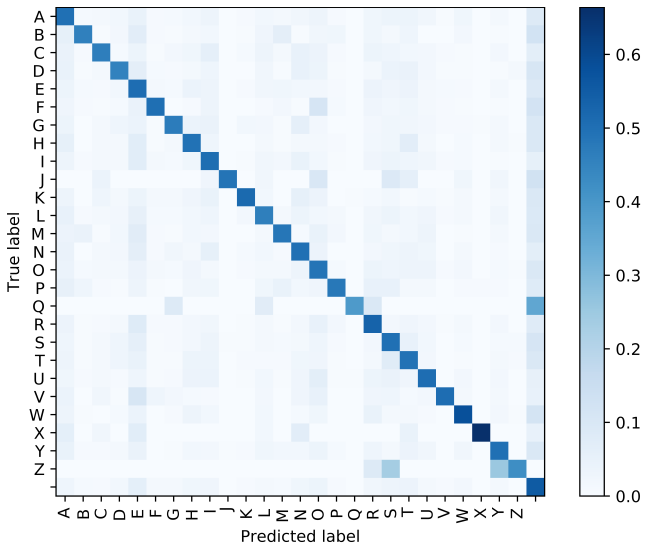


Figure 4.11: Confusion matrix for classification of ASCII characters.

tion performance and the viseme-to-word conversion performance need to be good. The VER is very low and any misclassifications that have occurred during the validation phase appeared to be influenced by the class imbalance of visemes present in the training data. When visemes are misclassified, they are most likely to be decoded as one of "AH", "K" or "T" because such visemes appear most frequently in training data and obscure classes such as "AA" and "CH" are the most likely to be misclassified.

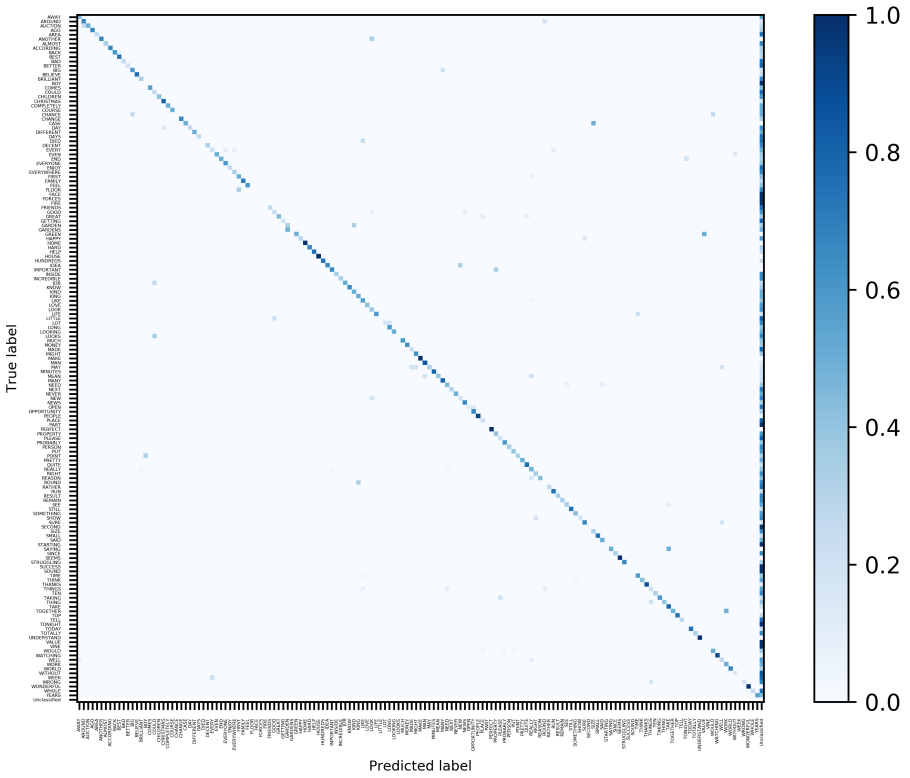


Figure 4.12: Word confusion matrix for Afouras et al's model.

Table 4.6 gives examples of sentences from the BBC LRS2 dataset along with the decoded visemes, the word combinations that were outputted at each iteration of the perplexity calculations, and the viseme clusters corresponding to each predicted word. Table 4.7 gives the full details of how those sentences were decoded by listing their corresponding visemes, the predicted visemes, the decoded sentences and their corresponding metric performance results.

A stratified sampling strategy was used to select the most frequently appearing 154 words in the BBC LRS2 training set that begin with each letter of the alphabet. For the selected 154 words, a comparison of the accuracy in terms of ratio of how many times a word was correctly

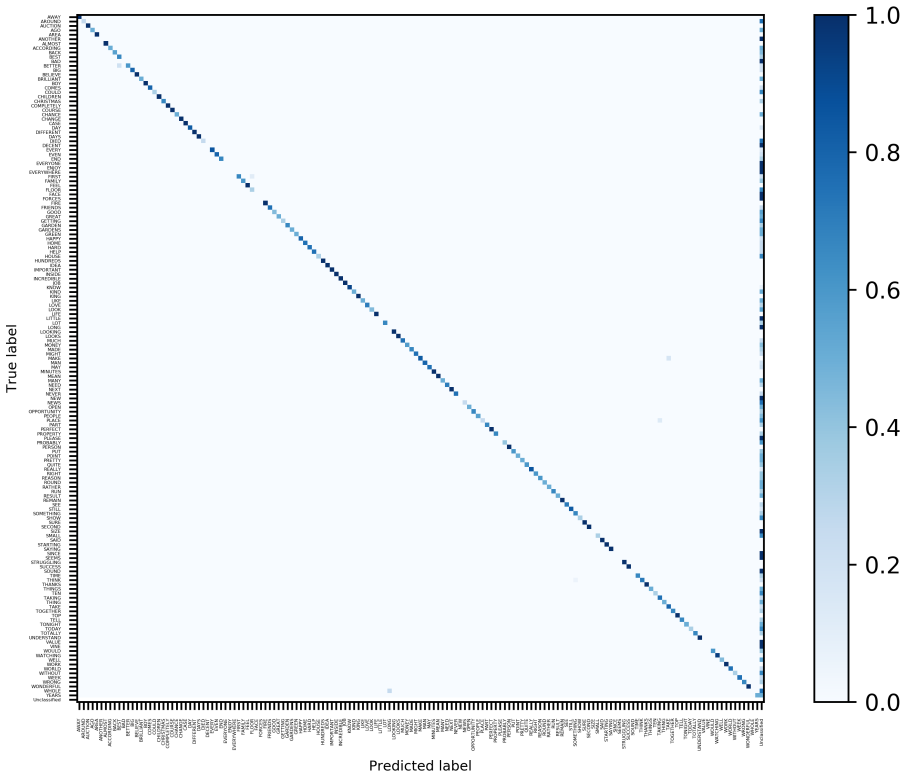


Figure 4.13: Word confusion matrix for the proposed lip-reading system.

decoded to how many times it appeared in the testing phase has been presented in Figures 4.12 and 4.13. Figure 4.12 shows the word accuracy for Afouras et al.'s model and Figure 4.13 shows the accuracy for this lip reading system. A better word precision is noticeable in Figure 4.13.

It should be noted that, whilst the VER was low, the WER was still high although it has been significantly improved compared to other existing works. To further reduce the error rate, the viseme-to-word conversion would need to be optimised. Many misclassifications have been caused by the presence of local optima during the implementation of the local beam search, whereby at each iteration of the viseme sequence during the perplexity calculation stage, the words that make up the ground truth are not included within the top 50 results. A large beam with would invariably result in a greater conversion rate, but at the expense of using more computational overhead and an exhaustive search would not even be viable. Further work needs to be done to ensure that the global optimum combinatorial solution is selected more frequently during the Perplexity Calculation stage to further improve on word accuracy.

Table 4.6: Examples of perplexity calculations for sentences from the test set.

Actual Subtitle	**Predicted Visemes**	**Decoded Subtitle Perplexity**	**Word Correspondence**
I	(’AH’),	(’a nouns’, 161.9), ("i can’t", 184.3), (’uh nouns’, 204.3), ...	(’AH’): i
CAN’T	(’K’, ’EY’,’K’,’T’),	(’a nouns but’, 182.5), (’uh nouns but’, 200.9), ("i can’t but", 223.3), ...	(’K’, ’EY’,’K’,’T’): can’t
PUT	(’P’,’AH’,’T’),	(’a nouns but it’, 120.6), (’uh nouns but it’, 125.0), ...	(’P’,’AH’,’T’): bite
IT	(’IY’,’T’),	("i can’t bite it any", 181.8), ("i can’t buss it any", 242.2), ...	(’IY’,’T’): it
ANY	(’EY’,’K’,’IY’),	("i can’t bite it any plainer", 130.8), ("i can’t buss it any plainer", 183.4), ...	(’EY’,’K’,’IY’): any
PLAINER	(’P’,’K’,’EY’,’K’,’ER’),	("i can’t bite it any plainer than", 87.4), ...	(’P’,’K’,’EY’,’K’,’ER’): plainer
THAN	(’T’,’EY’,’K’),	("i can’t bite it any plainer than that", 57.7), ...	(’T’,’EY’,’K’): than
THAT	(’T’,’EY’,’T’)	Result: i can’t bite it any plainer than that	(’T’,’EY’,’T’): that
WHEN	[’W’,’EY’,’K’),		[’W’,’EY’,’K’): when
THERE	(’T’,’EY’,’W’),	(’when there’, 121.0), ("when they’re", 216.4), (’whack their’, 220.6), ...	(’T’,’EY’,’W’): there
ISN’T	(’IY’,’T’,’AH’,’K’,’T’),	("when they’re isn’t", 69.9), ("wreck there isn’t", 88.9), ...	(’IY’,’T’,’AH’,’K’,’T’): isn’t
MUCH	(’P’,’AH’,’CH’),	("when there isn’t much else", 52.5), ...	(’P’,’AH’,’CH’): much
ELSE	(’EY’,’K’,’T’),	("when there isn’t much else in", 60.9), ...	(’EY’,’K’,’T’): else
IN	(’IY’,’K’),	("when there isn’t much else in the", 41.9), ...	(’IY’,’K’): in
THE	(’T’,’AH’),	("when there isn’t much else in the garden", 60.3), ...	(’T’,’AH’): the
GARDEN	(’K’,’AA’,’W’,’T’,’AH’,’K’)	Result: when there isn’t much else in the garden	(’K’,’AA’,’W’,’T’,’AH’,’K’):garden
SORT	(’T’,’AO’,’W’,’T’),	(’sort of’, 1.2), (’source of’, 1.5), (’doors of’, 25.0), (’sword of’, 28.3), ...	(’T’,’AO’,’W’,’T’): sort
OF	(’AH’,’F’),	(’sort of second’, 55.3), (’sort of talent’, 81.1), (’source of talent’, 89.3), ...	(’AH’,’F’): of
SECOND	(’T’,’EY’,’K’,’AH’,’K’,’T’),	(’sort of tennent naff’, 147.4), (’sort of second half’, 158.6), ...	(’T’,’EY’,’K’,’AH’,’K’,’T’): second
HALF	(’K’,’EY’,’F’),	(’sort of second half of’, 60.4), ("zorz i’ve tennent naff i’ve", 132.7), ...	(’K’,’EY’,’F’): half
OF	(’AH’,’F’),	(’sort of second half of october’, 229.1), ...	(’AH’,’F’): of
OCTOBER	(’AA’,’K’,’T’,’AO’,’P’,’ER’)	Result: sort of second half of october	(’AA’,’K’,’T’,’AO’,’P’,’ER’): october
BUT	(’W’,’AH’,’T’),	(’right before’, 188.2), (’ride before’, 309.3), (’rise before’, 319.8), ...	(’W’,’AH’,’T’): right
BEFORE	(’P’,’IY’,’F’,’AO’,’W’),	(’right before i’, 41.2), (’ride before i’, 69.1), (’ries before i’, 81.2), ...	(’P’,’IY’,’F’,’AO’,’W’): before
I	(’AH’),	(’right before i do’, 55.9), (’ride before i do’, 78.6), ...	(’AH’): i
DO	(’T’,’UH’)	Result: right before i do	(’T’,’UH’): do
AS	(’IY’,’T’),	(’is a’, 14.4), (’eat a’, 39.7), (’ease a’, 56.6), ("e.’s a", 132.8), ...	(’IY’,’T’): is
A	(’AH’),	("e’s a whittle’s", 157.6), ("e’s i. whittle’s", 191.8),...	(’AH’): a
RESULT	(’W’,’IY’,’T’,’AH’,’K’,’T’),	(’is a result of’, 40.0), ("e’s i. whittle’s i’ve", 106.4), ...	(’W’,’IY’,’T’,’AH’,’K’,’T’): result
OF	(’AH’,’F’),	(’is a result of smoking’, 135.4), ("e’s a whittle’s i’ve smolin", 190.9), ...	(’AH’,’F’): of
SMOKING	(’T’,’P’,’AO’,’K’,’IY’,’K’)	Result: is a result of smoking	(’T’,’P’,’AO’,’K’,’IY’,’K’): smoking
PRETTY	(’W’,’W’,’IY’,’T’,’IY’),	(’wheatie on’, 169.2), (’reidy on’, 296.3), (’riedy on’, 349.9), ...	(’W’,’W’,’IY’,’T’,’IY’): witty
ON	(’AA’,’K’),	(’weedy on the’, 29.4), (’reedy on the’, 31.8), (’witty on the’, 56.7), ...	(’AA’,’K’): on
THE	(’T’,’AH’),	(’witty on the outside’, 45.3), (’weedy on the outside’, 52.2), ...	(’T’,’AH’): the
OUTSIDE	(’EY’,’T’,’T’,’AH’,’T’)	Result: witty on the outside	(’EY’,’T’,’T’,’AH’,’T’): outside
EVEN	(’T’,’AH’,’IY’,’K’),	(’dying before’, 346.6), (’sighing before’, 368.7), ...	(’T’,’AH’,’IY’,’K’): dying
BEFORE	(’P’,’IY’,’F’,’AO’,’W’),	(’sighing before she’, 64.1), (’dying before she’, 77.1), ...	(’P’,’IY’,’F’,’AO’,’W’): before
SHE	(’CH’,’IY’),	(’sighing before she answered’, 35.3), ...	(’CH’,’IY’): she
ENTERED	(’EY’,’K’,’T’,’ER’,’T’),	(’sighing before she answered a’, 85.4), ...	(’EY’,’K’,’T’,’ER’,’T’): entered
THE	(’AH’),	(’dying before she entered a water’, 190.5), ...	(’AH’): a
WATER	(’W’,’AO’,’T’,’ER’)	Result: dying before she entered a water	(’W’,’AO’,’T’,’ER’): water
LIKE	(’K’,’AH’,’K’),	(’nine hundreds’, 1831.3), (’lysne hundreds’, 2486.6), ...	(’K’,’AH’,’K’): nine
HUNDREDS	(’K’,’AH’,’K’,’T’,’W’,’AH’,’T’,’T’),	(’nine hundreds of’, 62.7), (’cul hundreds of’, 113.9), ...	(’K’,’AH’,’K’,’T’,’W’,’AH’,’T’,’T’): hundreds
OF	(’AH’,’F’),	(’nine hundreds of thousands’, 49.2), ...	(’AH’,’F’): of
THOUSANDS	(’T’,’EY’,’T’,’AH’,’K’,’T’,’T’),	(’nine hundreds of thousands of’, 20.8), ...	(’T’,’EY’,’T’,’AH’,’K’,’T’,’T’): thousands
OF	(’AH’,’F’),	(’nine hundreds of thousands of peopled’, 72.3), ...	(’AH’,’F’): of
PEOPLE DO	(’P’,’IY’,’P’,’AH’,’K’,’K’,’T’,’UH’),	(’nine hundreds of thousands of peopled every’, 103.8), ...	(’P’,’IY’,’P’,’AH’,’K’,’K’,’T’,’UH’): peoples
EVERY	(’EY’,’F’,’ER’,’IY’),	(’nine hundreds of thousands of peoples every year’, 65.4), ...	(’EY’,’F’,’ER’,’IY’): every
YEAR	(’K’,’IY’,’W’)	Result: nine hundreds of thousands of peoples every year	(’K’,’IY’,’W’) year

Table 4.7: Examples of how sentences from the test set were decoded.

Actual Subtitle	Corresponding Visemes	Predicted Visemes	Decoded Subtitle	VER(%)	CER(%)	WER(%)	SAR(%)
I CAN'T PUT IT ANY PLAINER THAN THAT	[('AH'),('K','EY','K','T'), ('P','UH', 'T'),('IY','T'), ('EY','K','IY'), ('P','K','EY', 'K','ER'), ('T','EY','K'),('T','EY','T')]	('AH'),('K', 'EY','K','T'), ('P','AH','T'),('IY','T'), ('EY','K','IY'), ('P','K','EY','K','ER'), ('T','EY','K'),('T','EY','T')	I CAN'T BITE IT ANY PLAINER THAN THAT	3.1	8.3	12.5	0.0
WHEN THERE ISN'T MUCH ELSE IN THE GARDEN	('W','EY','K'),('T','EY','W'), ('IY','T','AH','K','T'), ('P','AH','CH'),('EY','K','T'), ('IY','K'),('T','AH'), ('K','AA','W','T','AH','K')	['W','EY','K'),('T','EY','W'), ('IY','T','AH','K','T'), ('P','AH','CH'),('EY','K','T'), ('IY','K'),('T','AH'), ('K','AA','W','T','AH','K')	WHEN THERE ISN'T MUCH ELSE IN THE GARDEN	0.0	0.0	0.0	100.0
SORT OF SECOND HALF OF OCTOBER	('T','AO','W','T'),('AH','F'), ('T','EY','K','AH','K','T'), ('K','EY','F'), ('AH','F'), ('AA','K','T','AO','P','ER')	('T','AO','W','T'),('AH','F'), ('T','EY','K','AH','K','T'), ('K','EY','F'), ('AH','F'), ('AA','K','T','AO','P','ER')	SORT OF SECOND HALF OF OCTOBER	0.0	0.0	0.0	100.0
BUT BEFORE I DO	[('P','AH','T'), ('P','IY','F','AO','W'), ('AH'),('T','UH')]	('W','AH','T'), ('P','IY','F','AO','W'), ('AH'),('T','UH')	RIGHT BEFORE I DO	6.7	26.7	25.0	0.0
AS A RESULT OF SMOKING	('EY','T'),('AH'), ('W','IY','T','AH','K','T'), ('AH','F'), ('T','P','AO','K','IY','K')	('IY','T'),('AH'), ('W','IY','T','AH','K','T'), ('AH','F'), ('T','P','AO','K','IY','K')	IS A RESULT OF SMOKING	4.5	4.5	20.0	0.0
PRETTY ON THE OUTSIDE	('P','W','IY','T','IY'),('AA','K'), ('T','AH'),('EY','T','T','AH','T')	('W','W','IY','T','IY'),('AA','K'), ('T','AH'),('EY','T','T','AH','T')	WITTY ON THE OUTSIDE	5.6	14.3	25.0	0.0
EVEN BEFORE SHE ENTERED THE WATER	('IY','F','IY','K') ('P','IY','F','AO','W'), ('CH','IY'),('EY','K','T','ER','T'), ('T','AH'),('W','AO','T','ER')	('T','AH','IY','K'), ('P','IY','F','AO','W'), ('CH','IY'),('EY','K','T','ER','T'), ('AH'),('W','AO','T','ER')	DYING BEFORE SHE ENTERED A WATER	10.7	21.2	33.3	0.0
LIKE HUNDREDS OF THOUSANDS OF PEOPLE DO EVERY YEAR	('K','AH','K'), ('K','AH','K','T','W','AH','T','T'), ('AH','F'), ('T','EY','T','AH','K','T','T'), ('AH','F'), ('P','IY','P','AH','K'),('T','UH'), ('EY','F','ER','IY'),('K','IY','W')	('K','AH','K'), ('K','AH','K','T','W','AH','T','T'), ('AH','F'), ('T','EY','T','AH','K','T','T'), ('AH','F'), ('P','IY','P','AH','K','K','T','UH'), ('EY','F','ER','IY'),('K','IY','W')	NINE HUNDREDS OF THOUSANDS OF PEOPLES EVERY YEAR	2.2	10.0	33.3	0.0

A viseme-based lip-reading system would be expected to predict words with unique visemes to near 100% accuracy if visemes were decoded correctly. Table 4.8 gives the distribution of words in the LRS2 test set that either have or do not have unique visemes, while Figure 4.14 below gives the percentage ratios of both viseme clusters and words from the LRS2 validation set being classified correctly for the cases of words with unique visemes and for homopheme words. Though approximately half of words in English are homopheme words, words with unique sets of visemes tend to be more obscure and this is why the relative distribution between words with unique visemes and words without visemes in the test set is imbalanced. Theoretically, eventual word classification is expected to be higher for words with unique visemes than homopheme words because there is no one-to-many mapping for the viseme-to-word conversion once visemes have been predicted correctly and this explains why word accuracy is higher for words with unique visemes as shown in Table 4.8

Table 4.8: Distribution of word with either unique or non-unique visemes in the LRS2 test.

Word Category	**Total Words**	**Viseme Clusters predicted correctly**	**Words predicted correctly**	**Viseme Cluster Accuracy(%)**	**Word** Accuracy(%)
Homopheme Words	3696	3235	2348	87.5	63.5
Words with Unique Visemes	570	449	471	78.8	82.6

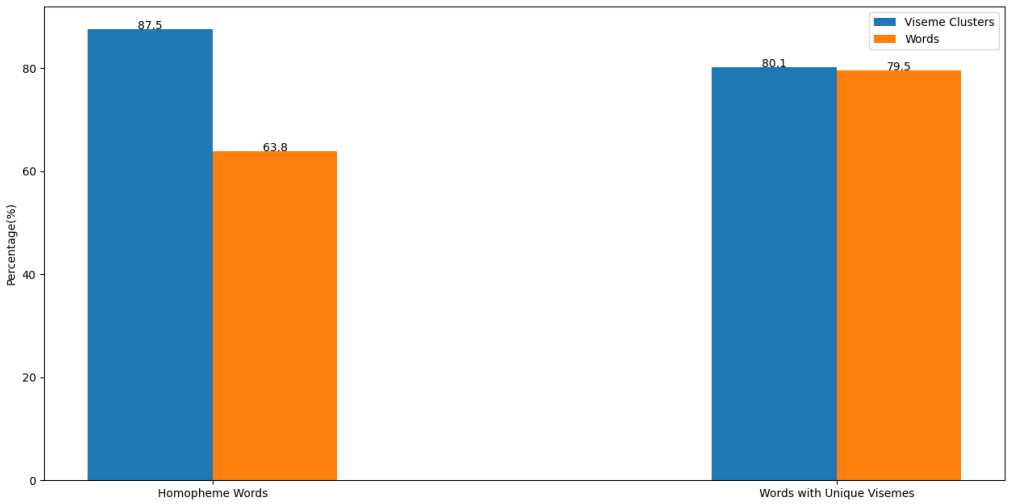


Figure 4.14: Accuracies for words and viseme clusters with unique and non-unique sets of visemes

For words in the validation set with unique visemes, the ratio of words being predicted correctly

is nearly identical to the the ratio viseme clusters being predicted correctly which is expected given that the set of visemes can only correspond to one possible word. The occurrence of unique sets of visemes being incorrectly matched is down to error propagation in the conversion model whereby incorrectly predicted words in the output sequence can cause other words in the sequence to be predicted incorrectly.

As well as being robust to varying levels of illumination, the viseme classifier has been trained and tested on samples with different ratios of testing samples to training samples. The majority of lip reading systems that were trained on the BBC-LRS2 dataset used a ratio of 2.71% but this is too small a ratio. The model was trained and tested on 4 different scenarios where by the ratio of test samples to train samples was increased from 10% to 40% keeping the total number of samples to the same figure of 47801. The best validation results attained have been shown in Table 4.9 and for each scenario, the viseme classifier was trained using 2000 iterations with learning curves shown in Figures 4.15 to 4.18.

Table 4.9: Performance of viseme classifier under different test to train ratios.

Ratio(%)	**Train Samples**	**Test Samples**	**Best validation VER(%)**	**VAR(%)**	**WER**
2.71	45839	1242	4.64	95.36	35.4
10.00	42373	4708	4.88	95.12	36.1
20.00	37665	9416	9.07	90.93	92.6
30.00	32957	14124	18.63	81.37	123.2
40.00	28249	18832	30.80	69.20	129.8

The best validation output for the viseme classifier at each ratio was used as the input to the Word Detector which is why a WER metric is also given in Table 4.9. The best validation accuracy for the viseme classifier does not change significantly when the the number of training samples is reduced. However a significant drop in the performance accuracy of the word detector is noticeable which further confirms that the word detector used is not robust to confused or incorrectly classified visemes.

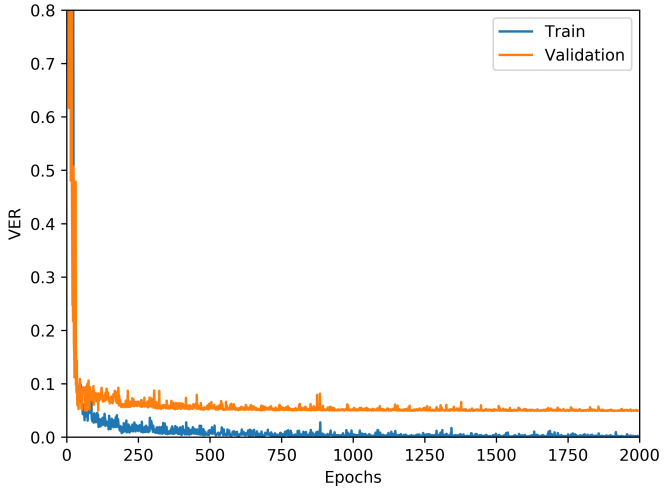


Figure 4.15: VER curve for training and validation for ratio 10%.

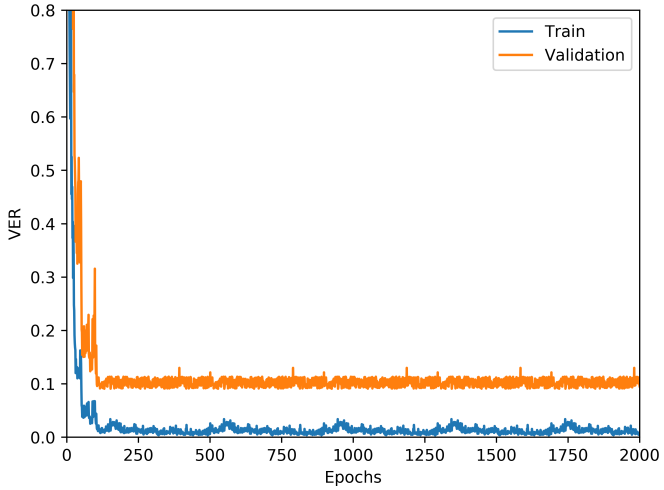


Figure 4.16: VER curve for training and validation for ratio 20%.

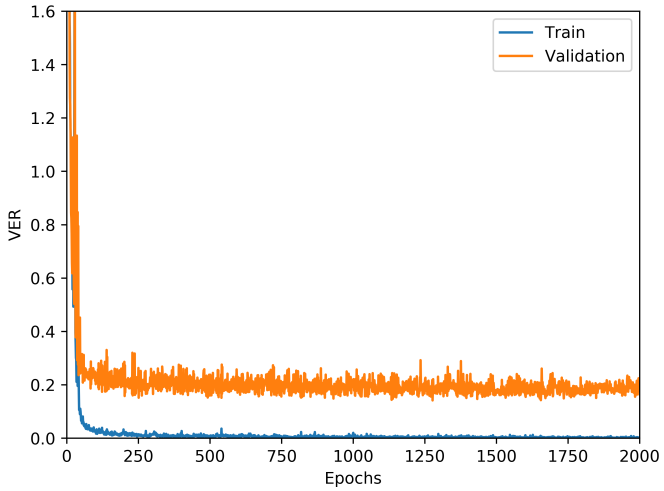


Figure 4.17: VER curve for training and validation for ratio 30%.

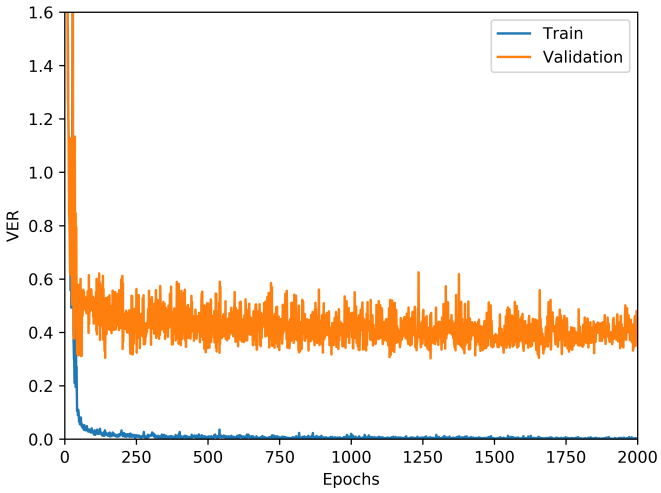


Figure 4.18: VER curve for training and validation for ratio 40%.

4.4 Summary

A neural network-based lip reading system has been developed to predict sentences covering a wide range of vocabulary in silent videos from people speaking. The system is lexicon-free, uses only visual cues represented by visemes of a limited number of distinct lip movements, and is robust to different levels of lighting. Verified on the BBC LRS2 data set, the system has demonstrated a significant improvement on classification accuracy of words compared to the state-of-the-art works.

In addition, an efficient conversion of visemes to words is crucial when using visemes as classification scheme for lip reading sentences. As shown in the experiments, although the classification accuracy of visemes achieved by the proposed system was very high (over 95%), the classification accuracy of words was significantly dropped after the conversion (65.5%). As such, it is important to explore any other possible approaches for the conversion. For perplexity analysis-based conversion, different global optimisation methods need to be considered while also limiting the computational overhead required.

Chapter 5

Viseme-to-Word Conversion with Robustness

This Chapter addresses the question *"Can a language model be implemented that is robust to confused visemes?"*. It presents a comparison of possible approaches used for viseme-to-word conversion and presents converter that is an improvement on the converter used in Chapter 4 in that it is more robust to misclassified visemes and quicker to execute. The converter described in Chapter 4 is robust to variations in lighting and pixel changes but it is not robust to the possibility of visemes being incorrectly decoded. The model proposed uses an attention-based language model that has been demonstrated to be effective at discriminating between words that are syntactically and semantically different compared with traditional language models.

A language model implemented in a viseme-based lip-reading system must be robust to the possibility of confused or incorrectly classified visemes. The language model used for predicting spoken words must be prone to not only the possibility of visemes not being classified correctly, but also that words at any point in sequence being predicted must prone to the possibility of earlier words in the sequence not being predicted correctly.

The rest of this Chapter is organised as follows: First in Section 5.1 an Introduction is given. Then in Section 5.2, all the distinct components that make up the viseme-to-word conversion process are described including: the principle of perplexity analysis, the neural network used for

performing viseme-to-word conversion, the data augmentation techniques used for modelling the converter's robustness to noise, and the accuracy metrics used to evaluate the performance of the word detector. In Section 5.3, the performance of the proposed word converter is discussed and compared with other approaches, followed by concluding remarks given in Section 5.4 along with suggestions for further research.

5.1 Introduction

The overall performance of a viseme-based lip-reading sentences system has been significantly affected by the efficiency of the conversion from visemes to words. A high accuracy for classification of visemes can be achieved such was reported in Chapter 4 where a viseme accuracy rate of more than 95.4% was recorded. However, the overall accuracy of word classification was dropped down to 64.6%. The underlying cause of this phenomenon is the existential problem whereby one set of visemes can map to multiple different sounds or phonemes resulting in a one-to-many relationship between sets of visemes and words. As such, it becomes critical to design efficient strategies for viseme-to-word conversion in order to develop practically useful viseme-based lip-reading systems.

In this chapter a viseme-to-word converter is proposed for effectively distinguishing between words sharing identical visemes and its performance is compared with three other approaches. Compared with other approaches [175] [173] [31] [182] [143], the proposed approach is more robust to the possibility of misclassified visemes in the input, and its robustness is demonstrated by adding perturbations to the input visemes and comparing the outputs to the ground-truth. Moreover, the converter is implemented in a deep learning network-based architecture for lip reading sentences from the BBC LRS dataset and it attained an improved performance of over 15% on other lip reading systems evaluated on the same corpus such as Ma et al. [104](who achieved a word accuracy of 62.1%) and Fenghour et al. [159](who achieved a word accuracy of 64.6%).

The rationale behind the comparison of four models is to compare conversion approaches from

three different categories of conversion implementation previously discussed in Section 3.4. One of the approaches uses a statistical language model with a fixed-context window while another of the approaches uses a feed-forward neural network with a fixed-context window. The other two approaches use neural language models but one of these approaches is known to perform poorly when there are incorrectly classified visemes as inputs because the predicted output is vulnerable to error propagation.

The best proposed approach has been theoretically and experimentally verified. It is shown to be more effective at discriminating between words sharing visemes that are either semantically and syntactically different because unlike other approaches that only use context from a fixed-window, the proposed approach uses unlimited context to detect semantic and syntactic information needed for the disambiguation. The proposed approach is also shown to be somewhat robust to incorrectly classified visemes due to its ability to model both long and short term dependencies.

5.2 Methodology

Given a sequence of visemes, the objective is to predict the possible sentence spoken by someone given that only the lip movements or visemes of the person speaking are known, and given that there is the possibility of an error or misclassification in the input viseme sequence. In this Section, an overall framework is proposed for the viseme-to-word conversion with a component for testing its robustness to misclassified viseme sequences. The entire process consists of two main components: a viseme-to-word conversion model for performing the viseme-to-word conversion and a noisifying component for performing data augmentation to vary the errancy of the input visemes. The performance of the system is evaluated by comparing sentences predicted by the viseme-to-word converter, to the ground truth of the actual sentences and then measuring the edit distance [39] (which is the minimum number of character-level operations required to correct the actual sentence to the ground truth).

One aim of this work is to model and improve upon the performance of viseme-to-word con-

version reported in another work [159] - particularly with attention to misclassified visemes because the word detector used relies on visemes being decoded with 100% precision. In addition to modelling the robustness of the viseme-to-word converter, the conversion performance attained is also sufficient to solve the problem of why visemes are not widely used as a classification schema in lip-reading, which is down the performance of viseme-to-word conversion being inadequate because of the failure to pick up on semantic and syntactic information needed to distinguish between words that share identical visemes (as discussed in Subsection 3.4.3). Four models have been utilised to convert visemes to words directly, and these include the following:

- An attention-based sequence model using GRUs
- A GPT-based iterator that uses perplexity scores
- A Feed-Forward Neural Network
- A Hidden Markov Model

Figure 5.1 outlines the framework of a lip reading system that uses solely visual cues. This framework comprises of a viseme classifier followed by a viseme-to-word converter where an image-based classification system takes image frames of a person's moving lips to classify visemes. Once the visemes have been recognised, a word detector uses the output of the viseme classifier as the inputs in order to determine which words were spoken. For the conversion of visemes to words, a variety of sequence modelling networks could be used to determine the most likely set of words to have been uttered given the visemes that had been decoded.

Some sequence-modelling approaches are prone to the possibility that one incorrectly decoded word causes other words in the rest of the outputted sequence to also be predicted incorrectly. This is why for the purpose of evaluating the robustness of a viseme-to-word converter, data augmentation techniques are used to add noise to viseme sequence using techniques that include deletion, insertion, substitution, and swapping so as to model the performance of the viseme-to-word converter under varying levels of noise. The augmentation or noisification techniques are described in more detail in Subsection 5.2.4. However, robustness performance results

have only been reported for the GRU model and GPT-based iterator because the priority of a viseme-to-word converter is its efficiency.

Overall, there are three instances that the viseme-to-word detection's performance is being reported. For all three of these instances, their respective performances will be compared with the perplexity-based viseme-to-word converter discussed in Subsection 5.2.3

1. Visemes with 100% accuracy where the identity of spoken visemes are known;
2. The outputs of the viseme classifier reported in [159];
3. Perturbed visemes with added noise whereby the errancy is varied.

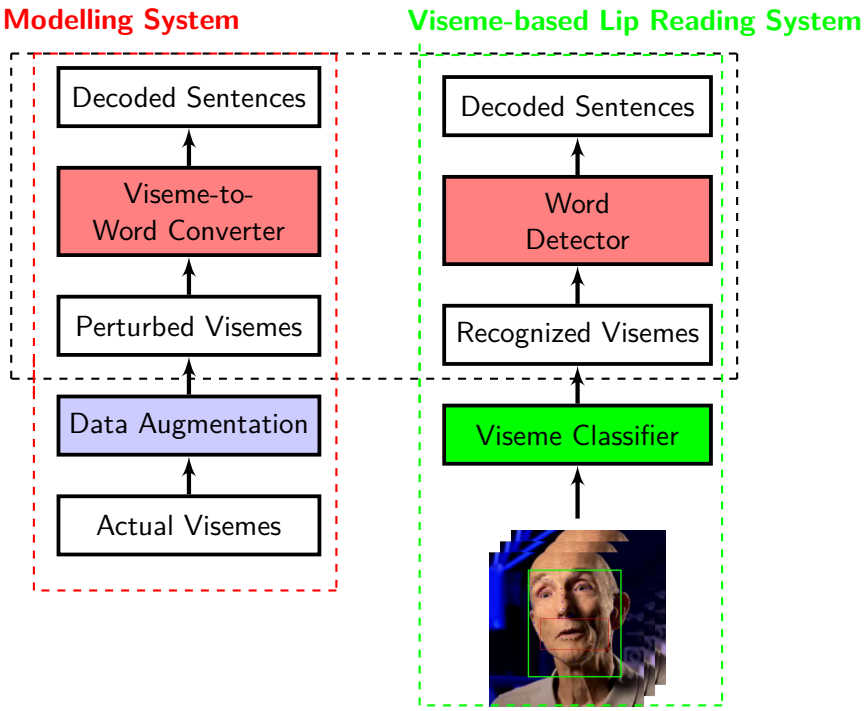


Figure 5.1: Modelling of viseme-to-word conversion.

5.2.1 Data

The dataset used is the BBC LRS2 dataset [17]. It consists of approximately 46,000 videos covering over 2 million word instances, a vocabulary range of over 40,000 words and sentences of up to 100 ASCII characters from BBC videos. This chapter is all about modelling how robust the viseme-to-word converter is to noise and misclassifications, so the details about the videos will not be discussed here. Additionally, videos from the LRS3-TED dataset [40] which is

similar to the LRS2 dataset has also been used for the scenario of visemes with 100% accuracy where the identity if spoken visemes are known. This dataset is more challenging because the sentences are on average longer in length and it consists of a vocabulary covering over 50,000 possible words.

Lip reading datasets like BBC LRS2 consist of labels in the form of subtitles. These subtitles are strings of words that need to be converted to sequences of visemes to provide labels for the viseme classifier. The conversion is performed in two stages: first, they are mapped to phonemes using the Carnegie Mellon Pronouncing Dictionary [204], and then the phonemes are mapped to visemes according to Lee and Yook's approach [35]. The GRU-based viseme-to-word converter uses 17 classes or input tokens in total; these include the 13 visemes, a space character, start of sentence (SoS), end of sentence (EoS) and a character for padding. All the defined classes are listed in Table 5.1. All viseme sequences are padded to 28 characters which is length of the longest viseme sequence.

There is no official standard convention for defining precise visemes or even the precise total number of visemes and different approaches to viseme classification have used varying numbers of visemes as part of their conventions with different phoneme-to-viseme mappings [35] [36] [32] [33] [30] [34]. All the different conventions consist of consonant visemes, vowel visemes and one silent viseme but Lee and Yook's [35] mapping convention appears to be the most favoured for speech classification and it is the one that has been widely utilised for this thesis. It is however accepted that there are multiple phonemes that are visually identical on any given speaker [23] [22].

For the classifier, an output token will be assigned for every single word that is contained within sentences that make up labels for the training data and there will be four additional tokens "<start>", "<end>", "<pad>" and "<unk>". The "<start>" token gets appended to the start of every sentence with the "<end>" token being appended to the end of every sentence, while "<unk>" is a token for modelling any words that do not appear in the training phase.

Table 5.1: Viseme Classes used for input to viseme-to-word converter [35].

[pad], AA, AH, AO, CH, ER, EY, F, IY, K, P, T, UH, W, <sos>, <eos>, [space]

5.2.2 Viseme Classifier

The Viseme Classifier used is identical to that used in [159] and it relies on the same Preprocessing and Visual Frontend. Videos consist of images with red, green and blue pixel values and a resolution of 160 pixels by 160 pixels; plus a frame rate of 25 frames/second. Identical pre-processing is used whereby videos go through the same stages of sampling, facial landmark extraction, grayscale conversion, and cropping around the boundary of the facial landmarks. Image data augmentation is then applied in the form horizontal flipping, random frame removal and pixel shifting; which is then followed by z-score normalisation. This results in reduced image dimensions of $112 \times 112 \times T$ dimensions (where T corresponds to the number of image frames).

The viseme classifier itself follows the transformer [137] architecture with an encoder-decoder structure using multi-head attention layers used as building blocks albeit with modifications to the decoder topology. The encoder used is a stack of self-attention layers and the decoder consists of 3 fully connected layer blocks. The viseme classifier takes pre-processed images of lips as input to predict sequences of visemes following the 17 classes referred to in Subsection 5.2.1. The network was trained on all 45839 training samples of the BBC LRS dataset with 1243 samples used for validation.

5.2.3 Viseme-to-Word Converters

To demonstrate the effectiveness of neural language models with unlimited context, the performance of an Attention-based GRU has been compared with two other conversion models, each of which are representative of viseme-to-word conversion models from the two categories listed in Figure 3.1 of Subsection 3.4.2, and these include a bigram Hidden Markov Model(to represent statistical language models) and a Feed-Forward Model(to represent neural language models with a fixed window). A third model in the form of a GPT-based iterator has also included for comparison in performance to demonstrate the Attention-based GRU's robustness to incorrectly classified visemes.

Hidden Markov Model

The Hidden Markov Model(HMM) is similiar to that used by Vogel et al. [209] which was used for a statistical machine translation task. Visemes can be modelled as individual visemes or as clusters where groups of visemes make up a word. Unlike the approaches of works reported in [181] and [159], this approach classifies words based on sequences of individual visemes. The HMM uses a bigram language model to predict words given the inputted visemes corresponding to one of 17 tokens, and the language model is accompanied with Laplace smoothing and a Katz backoff. The bigrams used to train the model are all extrapolated from one of either the LRS2 and LRS3 training sets.

Feed-Forward Neural Network

The feed-forward neural network used implements a language model similar to that of Bengio et al [179] and uses a context window of the same size as the HMM model. The networks consists of a dense layers with 1024 nodes plus a softmax layer with classes corresponding to all possible tokens contained within one of the LRS2 and LRS3 datasets. The network is trained using the cross-entropy loss function with the training set split into batches of 64 samples each. During the training phase, the Adam optimiser [205] is implemented with default parameters($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$) and initial learning rate 10^{-3}. A curriculum learning strategy is used to train the network much like the training performed for Chung et al's model [181]. For the first iteration, sentences are clipped to one word followed by two words in the next iteration, then three words, then four, and finally the full length of sentences. The rationale behind this is to better learn the grammatical structure of word combinations found in natural language by being able to learn N-grams of variable lengths.

GPT-based Iterator

For a spoken sequence of visemes $V = (v_1, v_2, \ldots, v_N)$ where v_i corresponds to every ith viseme, $W = (w_1, w_2, \ldots, w_N)$ represents any given combination of words that map to those visemes

and w_i corresponds to every ith word within the string of words. Given that visemes have a one-to-many mapping relationship with phonemes which results in a situation of a cluster of visemes that map to several different words, it is expected that the combination of words that are most likely to have been uttered would be the combination that is most grammatically correct and thus the combination with the greatest likelihood of occurrence. The string of words $\check{W}$ that is expected to have been uttered for a set of visemes would be the combination that has the greatest likelihood. A set of visemes V can map to any combination of words W_C for a combination C that falls within the overall set of combinations C^*. The solution to predicting the sentence spoken is the combination of words given the recognised visemes which has the greatest probability as expressed in Eq. 5.1.

$$\check{W} = \arg\max_{C \epsilon C^*} P(W_C|V) \tag{5.1}$$

Perplexity is a measure of the quality of a language model, because a good language model will generate sequences of words with a larger probability of occurrence resulting in a smaller perplexity. The Perplexity-based Word Detector of [159] maps cluster of visemes to words through an iterative procedure.

The word matching is performed in different stages shown in Figure 5.2 and the World Lookup stage is the very first stage. This is where every single cluster of visemes needs to be mapped to a set of words containing those visemes according to the mapping given by the Carnegie Mellon Pronouncing (CMU) Dictionary [204]. Once the word lookup stage is performed, the next stage of Word Detection is the Perplexity Calculations. The numerous possible choices of words that map to the visemes are combined, and perplexity iterations are performed to determine which combination of words is most likely to correspond to the uttered sentence, given the visemes recognised.

The word detector uses the GPT to calculate perplexity by taking the exponential of the cross-entropy loss for a particular combination of words. Eqs. 5.2 to 5.4 below describe the relationship between the perplexity PP, entropy H and the probability $P(w_1, w_2, ..., w_N)$ of a particular sequence of N words $(w_1, w_2, ..., w_N)$ [208]. PP can expressed as the exponentiation of

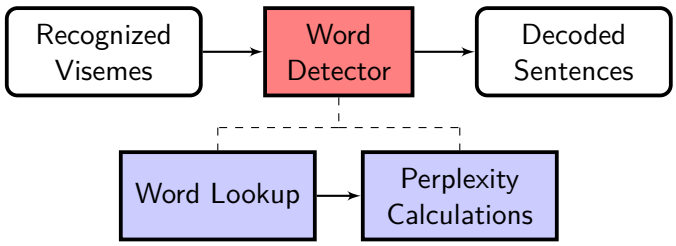


Figure 5.2: Processes of word detector.

entropy H in Eq. 5.2. The per-word entropy $\hat{H}$ is related to the probability $P(w_1, w_2, ..., w_N)$ of words $(w_1, w_2, ..., w_N)$. The value of $P(w_1, w_2, ..., w_N)$ that results in the choice of words selected as the output is that which results in the minimisation of entropy in Eq. 5.3, further resulting in the minimisation of perplexity given in Eq. 5.4 [208].

$$PP = e^{H} \tag{5.2}$$

$$\hat{H} = -\frac{1}{N} \ln P(w_1, w_2, ..., w_N) \tag{5.3}$$

$$PP = P(w_1, w_2, ..., w_N)^{-\frac{1}{N}} \tag{5.4}$$

When performing a conversion of visemes to words, some selection rules are implemented shown in Algorithm 1. If a viseme sequence has only 1 cluster matching to one word, that one word is selected as the output; whereas if a viseme sequence has only 1 cluster matching to several words, the word with largest expectation is selected as the output. This is determined by word rankings found in the Corpus of Contemporary American English(COCA) [35]. If a viseme sequence has more than 1 cluster, the words matching to the first two clusters are combined in every possible combination for the first iteration and the combinations with the lowest 50 perplexity scores are kept. If there are more clusters in the sequence to be matched, then these combinations are in turn combined with the words matching to the next viseme cluster keeping combinations with the lowest 50 perplexity scores at each iteration until the end of the sequence is reached. The selection of the lowest 50 perplexity scores at each iteration is based on an implementation of a local beam search with width 50.

One advantage of the language model used by GPT-based iterator is that when predicting a

Algorithm 1 Rules for Sentence Prediction

Require: Viseme Clusters V, Beam With B, Coca Rankings C, Word Lexicon mapping L, Predicted Output O, Perplexity scores for sentences p_s

if $V.length = 1$ and $L_V.length = 1$ **then**
  Select 1 Word Match
  $O \leftarrow L_V$
if $V.length = 1$ and $L_V.length > 1$ **then**
  Select Highest ranked word according to COCA
  $O \leftarrow C^{-1}(\max\{C_L\} : w)$
if $V.length > 1$ **then**
  Exhaustively combine words matching to $V_{n=0:1}$
  Select Combinations with lowest B Perplexity scores for $V_{n=0:1}$
  $p_s \leftarrow \min_{s \in B}\{s : PP(s)\}$
  $sents \leftarrow p_s^{-1}(PP(s) : s)$
  for For $n = 2$, $n < V.length$, $n++$ **do**
    $L_V \leftarrow$ Perform word matches for V_n
    Combine sentences from $sents$ with words from L_V
    Select Combinations with lowest B Perplexity scores
    $p_s \leftarrow \min_{s \in B}\{s : PP(s)\}$
    $sents \leftarrow p_s^{-1}(PP(s) : s)$
  $p_s \leftarrow \min\{s : PP(s)\}$
  $O \leftarrow p_s^{-1}(PP(s) : s)$

word at a particular timestep, it is able to base the prediction on all previous words predicted in the sentence. For a sentence of K words, the choice of the K'th word can be conditioned on all the previous $K-1$ words as a context which makes it a better implementation of Markov chains(Eq. 5.5). One disadvantage of this is that for long sentences, it would create more computational overhead but it also make the model more prone to errors if one word in the sentences is predicted incorrectly. The GPT-iterator calculates perplexity scores of words in combination so one incorrect word causes a cascading of errors.

$$P(w_1, w_2, ..., w_N) = P(w_1)P(w_2|w_1)...P(w_i|w_1, w_2, ..., w_i - 1) \tag{5.5}$$

Attention-GRU

Like [181], the neural network architecture used for word detection follows a Recurrent Neural Network(RNN) Encoder-Decoder structure modelled according to neural machine translation whereby for a given input sequence of visemes x, a sequence of words y(Eq. 5.6) is outputted. However, the RNN here is in the form of a GRU not an LSTM; also, the input here takes the form a sequence of individual visemes as opposed to clusters of visemes and so only requires 17 tokens given in Table 5.1 to be encoded.

$$y = \arg\max_{I \epsilon I^*}(y|x) \tag{5.6}$$

An encoder-decoder framework(Figure 5.3) takes an input sequence of vectors $x = x_1, \ldots, x_t$ where x_t corresponds to a vector, and inputs into a vector c with hidden state h_t at time t. The vector c is generated from the sequence of hidden states while f and q are non-linear variables. Vectors h_t and c are given in Eqs. 5.7 and 5.8 [124]. For this network, the encoder and decoder each consist of a GRU with 1024 nodes and a softmax layer with each possible word from the two corpuses LRS2 and LRS3 encoded as as as class. Sequences of visemes are the sequence inputs and they consist of 17 input tokens.

$$h_t = f\left(x_t, h_{t-1}\right) \tag{5.7}$$

$$c = q\left(\{h_1, ..., h_{t-1}\}\right) \tag{5.8}$$

The decoder is trained to predict the next word y_t in a sequence given the context vector c and all the previously predicted words $y_1, ..., y_t$. The decoder defines a probability $p(y)$ given in Eq. 5.9 over the prediction probability $p(y)$ by considering the joint conditional probability of all other previous words. A sentence predicted at time t follows with probability $p(y_t)$ follows the expression given in Eq. 5.10 where g is a nonlinear and s_t is a the hidden state of the GRU.

$$p(y) = \prod_{t=1}^{T} p(y_t|y_1, \ldots, y_t, c) \tag{5.9}$$

$$p(y_t|y_1, \ldots, y_t, c) = g\left(y_{t-1}, s_t, c\right) \tag{5.10}$$

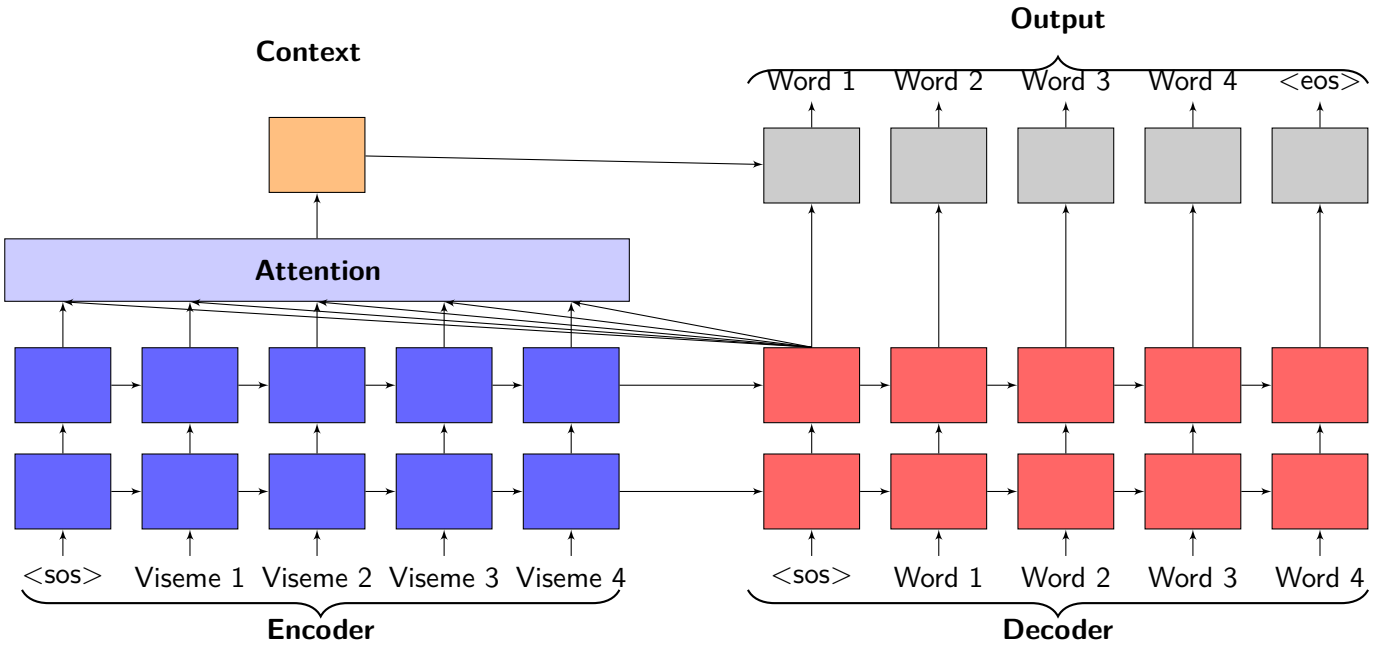


Figure 5.3: Components of Attention-based GRU architecture

Sequences of visemes are inputted into the encoder, while teacher forcing is used to provide the inputs for the decoder(as seen in Figure 5.3). During training, the ground truth for the previous timesteps would be used as the decoder inputs, whereas for validation, the predicted outputs of the previous timesteps provide the inputs to the decoder.

The neural network architecture uses Bahdanau's attention mechanism [124] for learning to align and predict sequences. The mechanism consist of components which include an alignment score, attention weights and a context vector. The alignment score is a component for learning the mapping relationship between different inputs and outputs. The network is trained using identical hyper-parameters to the feed-forward neural networks and the same curriculum learning strategy.

One obvious advantage of this architecture is that it allows the overall speech recognition system to use fewer parameters (roughly 16 million parameters) in comparison to other lip reading systems like the Transformer based network of Afouras et el. [105] used for decoding sentences from the LRS2 which used roughly 100 million parameters. Moreover, the GPT-absed iterator for the viseme-to-word converter uses the GPT to calculate perplexity for every word combination made at each stage of the iterative procedure, meaning that the number of times the model will have to be evaluated will increase exponentially with the number of words contained in an uttered sentence. The iterator uses a beam search width of 50 so a minimum of 50^{n-1} perplexity iterations would need to be performed for a sentence with n words.

5.2.4 Data Noisification

Data noisification [210] is implemented for the purpose of evaluating how robust the viseme-to-word classifier is to errancy in the inputted visemes by adding noise in the form of misclassification to the inputs. Noisification is implemented by adding small perturbations to the input visemes and there are four different techniques being implemented. These four techniques include random deletion, insertion, substitution and swapping [210].

Random Deletion [210]. is a technique where random visemes are deleted according to a probabilistic metric α_{rd}. The total number of visemes n_rd that gets deleted for a sequence with n_v total visemes is equivalent product of α_{rd} and n_v rounded to the nearest integer given in Eq. 5.11.

$$n_{rd} = \alpha_{rd} n_v \tag{5.11}$$

Random Swapping [210] involves the swapping of random visemes implemented according to a probabilistic metric α_{rs} and the total number of visemes n_v. The number of swap operations $n_r s$ that takes place is governed by the outcome of Eq. 5.12. This is simply the product of α_{rs} and n_v rounded to the nearest integer. The two visemes that get swapped are chosen by generating two random numbers to determine the positions of those two respective visemes to be swapped.

$$n_{rs} = \alpha_{rs} n_v \tag{5.12}$$

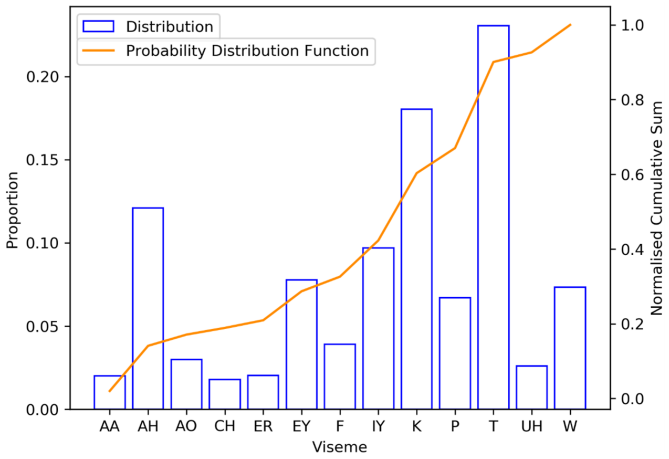


Figure 5.4: Probability distribution for generating visemes.

Random Insertion [210] is a process where random visemes are inserted along parts of the viseme sequences according to probabilistic metric α_{rs} and the number of visemes n_v. Like random swapping, the number of insertion operations to be performed is calculated using a similar equation. The number of insertions $n_r i$ that occurs is governed by the outcome of Eq. 5.13 which is the product of α_{ri} and n_v rounded to the nearest integer.

$$n_{ri} = \alpha_{ri} n_v \tag{5.13}$$

The choice of viseme that does get inserted for the random insertion operation is determined by a random number operation, such that the identity of the viseme will be generated according to a probability distribution matching the viseme distribution of BBC LRS2 training set. The rationale behind this is that when visemes are misclassified, they are most likely to be classified as any of the most frequently appearing visemes found in the training set. Figure 5.4 shows the cumulative probability distribution for visemes contained within the LRS2 training set.

The other technique called Random Substitution [210]. is where random positions along the viseme sequence are chosen, and the viseme corresponding to that position gets substituted for another viseme. The number of substitutions that takes place is set by Eq. 5.14 where for n_v total visemes and a probabilistic metric α_{sr} for substitution, a number of substitutions operations n_{sr} take place. Like the random insertion operation, the new viseme being substituted will be generated according to a probability distribution matching the viseme distribution of BBC LRS2 training set.

$$n_{sr} = \alpha_{sr} n_v \tag{5.14}$$

5.3 Experiment and Results

For the training and evaluation of the viseme-to-word converters mentioned in Section 5.2 excluding the GPT-based interator, BBC LRS2 sentence data described in 5.2.1 has been used with 80% of all sentences used for training(37666 samples) and 20% sentences being utilised for testing(9416 samples). k-Fold cross validation has been used with a fold value for k=5 and for each fold, a different set of 9416 samples were used. Viseme-to-word conversion has also been performed for sentences from the LRS3 corpus with 26588 samples for training, 6477 samples for testing and k=5 for k-Fold cross validation.

The metrics reported include CER, WER, SAR and the word accuracies(WAR). Performance results for word prediction are given for the three situations:

1. Correct Visemes,

2. Visemes classified as outputs of the viseme classifier reported in [159]; and

3. Perturbed Visemes with added noise to vary the errancy.

For Situation 1, the final performance results reported are averaged over each of the folds for the k-fold cross validation. Because the standard deviations of the WAR were small in comparison to the WAR values themselves, the decision was taken to only use models trained on Fold 1 for Situations 2 and 3. The rationale behind this decision was purely for ease of reporting and because the priority was to test the robustness of the viseme-to-word converter itself. Therefore, results were reported for that particular fold.

Situation 2 is significant because it is identical to substituting the viseme-to-word converter used in [159] with the GRU-based converter proposed in this paper whilst using the same viseme classifier for classifying visemes.

For the third situation, the accuracy of incident visemes is altered using the noisification process described in Section 5.2.4 where all probabilistic indicators are modified to vary the viseme accuracy. The probabilistic indicators α_{rd}, α_{ri}, α_{rs} and α_{sr}; for deletion, insertion, swapping and substitution respectively, are all set to the same value α_{mod} and incremented to vary the noise level on the visemes being inputted. Once the viseme-to-word detector has been trained, the trained network is evaluated on different incident viseme accuracies ranging from 70% to 100% to examine its robustness to noise.

The GRU architecture, feed-forward network and HMM were trained for several epochs until no improvement in either the training or validation losses were observed. It was at the point that the validation loss stopped converging that the performance of the model was evaluated. As well as modelling the GRU network's performance under different levels of viseme noise, it has also been compared with the performance of the GPT-based iterator.

Tables 5.2 and 5.3 lists the performance metrics of all four models for Situation 1 for the LRS2 and LRS3 corpuses when inputted visemes are known to be 100% correct. It it is noticeable

that the performances of all four models when decoding sentences from the LRS3 set were not as good as those for LRS2 and this can be explained by the fact that the LRS3 corpus consists of longer sentences and a deviation between the predicted sentence and ground-truth is more likely as the sentence lengths increase.

Table 5.4 gives the performance metrics of the four architectures for Situation 2 using the output of the viseme classifier in [159](results reported in Chapter 4), where $VER \approx 4$ % and it is clear that the Attention-based GRU and GPT-based iterator are significantly more effective in their conversion compares with the feed-forward network and HMM because they are able to exploit larger context windows.

The Attention-based GRU outperforms the GPT-based iterator for Situation 1 where the identity of visemes is known with 100% accuracy. But even with the smallest noise added to the viseme inputs, the difference between the performances of the two models diverge and the GRU network is clearly more resilient to perturbations in the input viseme sequences. Tables 5.7 and 5.8 both give samples of how some sentences are predicted by all four models along with time elapsed for execution. Confusion matrices have been plotted in Figures 5.5, 5.10, 5.11 and 5.12 for the GPT-based iterator, GRU network, Feed-forward network and HMM correspondingly.

Additionally, the resilience of the attention-based GRU to perturbations compared with the GPT-transformer based iterator is further observed when more noise is added to the input viseme sequences by analysing the performance results of Situation 3. The difference in character and word error rates recorded by both models grows even further apart with the increase in errancy of visemes as shown in Table 5.5 and Figures 5.6 and 5.7(for LRS2) or Table 5.6 and Figures 5.8 and 5.9(for LRS3).

The improvement in performance of predicting sentences with the GRU network especially with perturbed inputs can be explained by two main factors. The first is that word matching is done on an individual viseme level rather than being done on a cluster level like for the perplexity-based iterator; so if there is a word with one viseme being decoded incorrectly, the word it is contained with can still be identified correctly because the network is designed to classify visemes in combination.

This is not the case for GPT-based iterator which maps clusters to words, meaning that one viseme being decoded incorrectly would cause the entire cluster to be matched to the wrong words and an example of this can be seen with the sentence "for a brief time" being decoded as "or a brief time" by the GPT-based iterator. The reason for this incorrect prediction is because the first viseme "F" has been incorrectly decoded as "AO", yet the attention-based GRU is able to predict the spoken sentence correctly.

The second reason for there being a better resilience is that the GRU network is better at modelling shorter groups of words [211]. It does not suffer from the problem posed in the mapping of viseme clusters to words using the GPT-based iterator whereby compound errors occur in the combination of words during the iterations and in which the sentence being decoded is based on the conditional dependence of word combinations.

The GPT-iterator model uses the GPT to calculate perplexity scores of word combinations matching to viseme clusters in an iterative manner starting from the beginning of the sentence as opposed to being used for word prediction. If one viseme is misclassified, the input cluster would then be wrong leading to not only incorrect word matches for that one cluster but would also cause words further along the sequence to be incorrectly predicted because the words in the rest of the sentence are all dependant on words that have previously predicted. Moreover due to the curriculum learning strategy deployed for training the GRU network, it is better at recognising shorter N-grams [169] [212] [213].

When looking at the differences in how some sentences were decoded by both systems, it is clear to see that the system with the GRU network is less affected by compound errors in the prediction, because when one word has been predicted incorrectly, it will be less likely that other words in the outputted sentence would also be classified incorrectly too.

As well as the GRU network being more robust to noisy inputs than the GPT-based iterator, it also more efficient and requires less overhead, which is why it takes significantly less time to execute than the GPT-based iterator. The GPT-based iterator uses approximately 11 times the number of parameters as the GRU network does and as seen in Table 5.7, it takes significantly more time when decoding visemes.

When comparing how the conversion of sequences of visemes to words for all four models for some samples in Tables 5.7 and 5.8, it is noticeable that the accuracy of the two models utilising unlimited context, namely the GPT-based Iterator and Attention-based GRU are significantly more accurate in their conversions compared with both the Feed-Forward network and Hidden Markov Model which utilise fixed context windows. For instance for the sequence of visemes that corresponds to the sentence "for a brief time", the last viseme cluster corresponding to the word "time" was actually predicted by both the Feed-Forward Network and Hidden Markov Model as "type". The words "time" and "type" are both homopheme words, yet they are both semantically different and a longer context window is needed to be able to exploit semantic information to predict the correct word.

Table 5.2: Performance of viseme-to-word converters for Situation 1 on the LRS2 dataset.

Converter	**Fold No.**	**CER(%)**	**WER(%)**	**SAR(%)**	**WAR(%)**
GPT-based Iterator	Fold 1	10.7	18.0	56.8	82.0
GPT-based Iterator	Fold 2	11.4	19.5	55.1	80.5
GPT-based Iterator	Fold 3	11.2	19.1	54.8	80.9
GPT-based Iterator	Fold 4	12.0	20.3	54.2	79.7
GPT-based Iterator	Fold 5	11.0	18.6	54.9	81.4
GPT-based Iterator	**Average**	**11.3±0.5**	**19.1±0.9**	**55.2±1.0**	**80.9±0.9**
Attention-based GRU	Fold 1	6.2	8.8	74.9	91.2
Attention-based GRU	Fold 2	6.9	9.7	74.2	90.3
Attention-based GRU	Fold 3	7.5	10.6	73.4	89.4
Attention-based GRU	Fold 4	7.4	10.4	73.4	89.6
Attention-based GRU	Fold 5	7.1	10.2	73.8	89.8
Attention-based GRU	**Average**	**7.0±0.5**	**9.9±0.7**	**73.9±0.6**	**90.1±0.7**
Feed-Forward Network	Fold 1	31.7	42.7	9.4	57.3
Feed-Forward Network	Fold 2	32.4	43.4	8.6	56.6
Feed-Forward Network	Fold 3	33.0	44.1	7.8	55.9
Feed-Forward Network	Fold 4	32.8	43.9	8.1	56.1
Feed-Forward Network	Fold 5	32.6	43.5	8.1	56.5
Feed-Forward Network	**Average**	**32.5±0.5**	**43.5±0.5**	**8.4±0.6**	**56.5±0.5**
Hidden Markov Model	Fold 1	34.0	44.5	9.0	55.5
Hidden Markov Model	Fold 2	35.3	46.2	7.8	53.8
Hidden Markov Model	Fold 3	36.1	49.8	6.5	50.2
Hidden Markov Model	Fold 4	35.8	48.0	8.0	52.0
Hidden Markov Model	Fold 5	35.2	45.9	7.4	54.1
Hidden Markov Model	**Average**	**35.3±0.8**	**46.9±2.1**	**7.7±0.9**	**53.1±2.1**

Table 5.3: Performance of viseme-to-word converters for Situation 1 on the LRS3 dataset.

Converter	**Fold No.**	**CER(%)**	**WER(%)**	**SAR(%)**	**WAR(%)**
GPT-based Iterator	Fold 1	18.8	31.7	36.2	68.3
GPT-based Iterator	Fold 2	19.7	32.5	34.8	67.5
GPT-based Iterator	Fold 3	20.3	33.3	33.7	66.7
GPT-based Iterator	Fold 4	19.4	32.2	35.3	67.8
GPT-based Iterator	Fold 5	18.6	31.3	36.3	68.7
GPT-based Iterator	**Average**	**19.4±0.7**	**32.2±0.8**	**35.3±1.1**	**67.8±0.8**
Attention-based GRU	Fold 1	10.2	15.0	59.2	85.0
Attention-based GRU	Fold 2	10.5	15.4	59.0	84.6
Attention-based GRU	Fold 3	11.2	16.1	58.2	83.9
Attention-based GRU	Fold 4	10.9	15.8	58.2	84.2
Attention-based GRU	Fold 5	11.5	16.8	57.6	83.2
Attention-based GRU	**Average**	**10.9±0.5**	**15.8±0.7**	**58.4±0.7**	**84.2±0.7**
Feed-Forward Network	Fold 1	38.5	49.9	7.1	50.1
Feed-Forward Network	Fold 2	39.4	51.3	6.3	48.7
Feed-Forward Network	Fold 3	41.1	52.1	5.4	47.9
Feed-Forward Network	Fold 4	39.6	51.6	6.2	48.4
Feed-Forward Network	Fold 5	39.3	51.3	6.3	48.7
Feed-Forward Network	**Average**	**39.6±0.9**	**51.2±0.8**	**6.3±0.6**	**48.8±0.8**
Hidden Markov Model	Fold 1	41.3	52.1	7.0	47.9
Hidden Markov Model	Fold 2	42.5	54.2	6.1	45.8
Hidden Markov Model	Fold 3	43.3	54.9	5.4	45.1
Hidden Markov Model	Fold 4	42.6	54.4	5.8	45.6
Hidden Markov Model	Fold 5	42.2	53.8	6.3	46.2
Hidden Markov Model	**Average**	**42.4±0.7**	**53.9±1.1**	**6.1±0.6**	**46.1±1.1**

Table 5.4: Performance of viseme-to-word converters for Situation 2.

Viseme-to-Word Converter	**CER(%)**	**WER(%)**	**SAR(%)**	**WAR(%)**
GPT-based iterator	23.1	35.4	33.4	64.6
Attention-based GRU	14.0	20.4	49.8	79.6
Feed-Forward Network	67.2	78.7	2.9	21.3
Hidden Markov Model	71.4	81.7	2.8	18.3

Table 5.5: Performance of viseme-to-word converters under varying noise levels on the LRS2 dataset.

α_{mod}	VER(%)	Attention-based GRU		GPT-based iterator	
		CER(%)	WER(%)	CER(%)	WER(%)
0	0.0	5.8	8.6	10.7	18.0
5	3.1	12.9	18.4	21.2	35.7
6	3.7	14.8	20.9	21.9	37.0
7	4.4	17.1	24.1	22.3	37.5
8	5.3	18.8	26.5	26.3	44.3
10	7.3	24.9	33.7	32.5	54.8
15	11.1	31.7	43.0	40.4	68.1
20	16.2	40.9	54.4	43.5	73.4
25	20.3	48.2	63.0	59.2	100.0
30	23.9	53.4	68.4	67.5	113.9
35	27.7	57.2	74.5	72.7	122.7

Table 5.6: Performance of viseme-to-word converters under varying noise levels on the LRS3 dataset.

α_{mod}	VER(%)	Attention-based GRU		GPT-based iterator	
		CER(%)	WER(%)	CER(%)	WER(%)
0	0.0	10.2	15.0	18.8	31.7
5	2.8	22.5	31.5	35.5	62.4
6	3.5	25.0	35.8	37.6	64.6
7	4.5	29.8	41.7	38.0	63.2
8	5.2	32.8	45.8	45.6	77.1
10	7.6	41.5	56.8	56.9	93.7
15	11.0	53.6	72.1	70.6	118.2
20	16.5	70.8	90.7	76.1	127.3
25	20.1	80.8	108.3	102.0	169.8
30	23.9	92.9	117.1	116.4	199.8
35	27.5	99.3	127.9	125.7	205.5

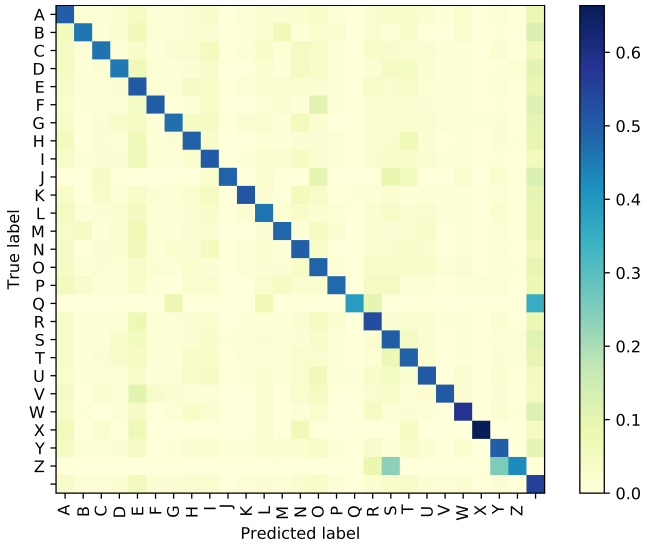


Figure 5.5: Confusion Matrix for GPT-based Iterator.

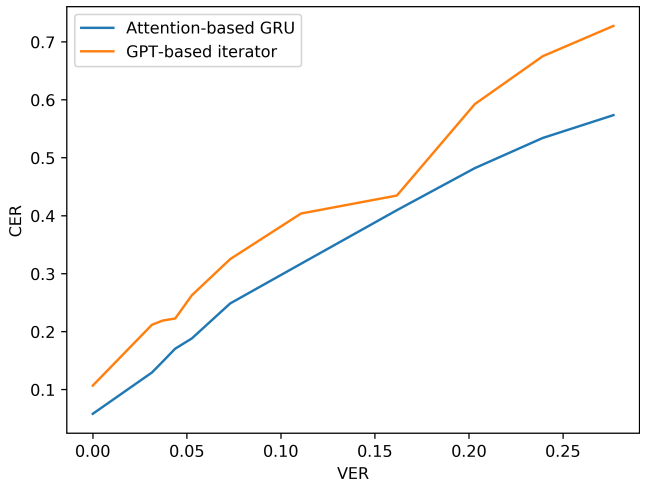


Figure 5.6: CER performance under varying noise levels(evaluation on LRS2 corpus).

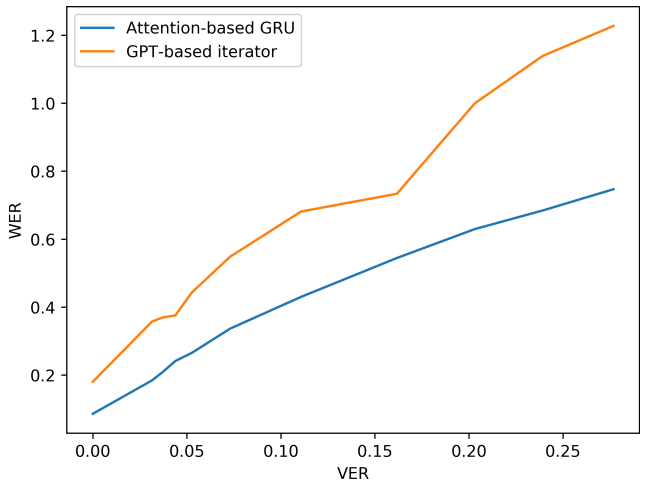


Figure 5.7: WER performance under varying noise levels(evaluation on LRS2 corpus).

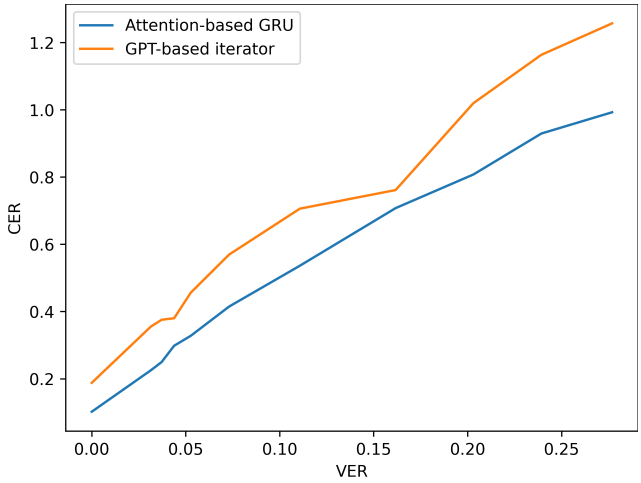


Figure 5.8: CER performance under varying noise levels(evaluation on LRS3 corpus).

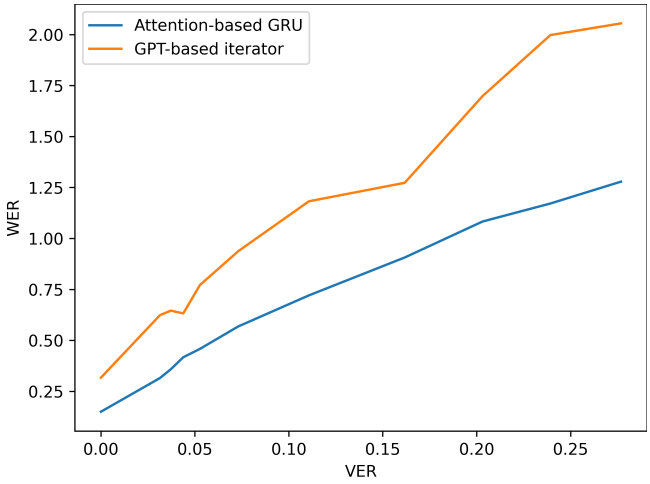


Figure 5.9: WER performance under varying noise levels(evaluation on LRS3 corpus).

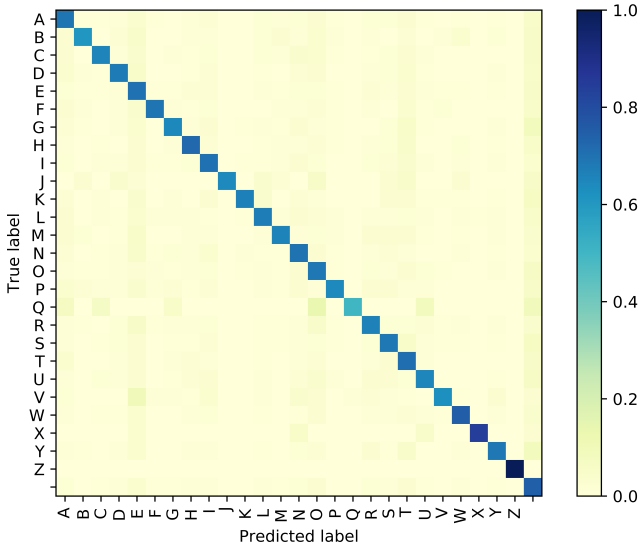


Figure 5.10: Confusion Matrix for Attention-based GRU.

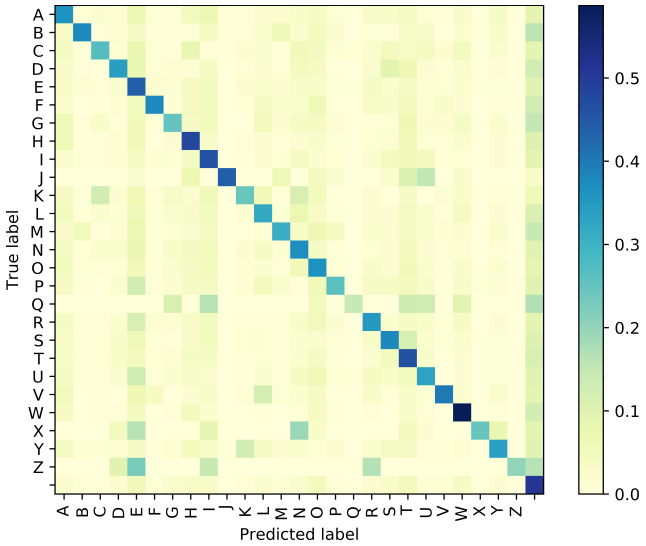


Figure 5.11: Confusion Matrix for Feed-Forward Network.

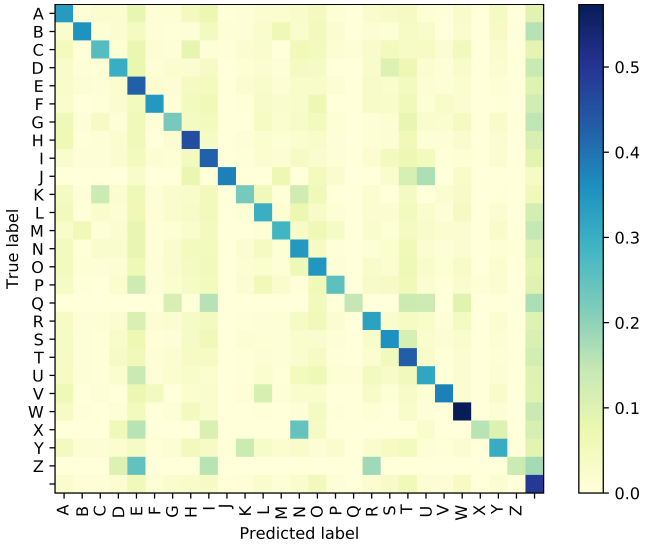


Figure 5.12: Confusion Matrix for Hidden Markov Model.

Table 5.7: Examples of decoded sentences from the two viseme-to-word converters.

Actual Subtitle	**Actual Visemes**	**Predicted Visemes**	**GPT-based iterator**		**Attention-based GRU**	
			Decoded Subtitle	**Execution Time(s)**	**Decoded Subtitle**	**Execution Time(s)**
WHEN THERE ISN'T MUCH ELSE IN THE GARDEN	('W', 'EY', 'K'), ('T', 'EY', 'W'), ('IY', 'T', 'AH', 'K', 'T'), ('P', 'AH', 'CH'), ('EY', 'K', 'T'), ('IY', 'K'), ('T', 'AH'), ('K', 'AA', 'W', 'T', 'AH', 'K')	('W', 'EY', 'K'), ('T', 'EY', 'W'), ('IY', 'T', 'AH', 'K', 'T'), ('P', 'AH', 'CH'), ('EY', 'K', 'T'), ('IY', 'K'), ('T', 'AH'), ('K', 'AA', 'W', 'T', 'AH', 'K')	WHEN THERE ISN'T MUCH ELSE IN THE GARDEN	147.35	WHEN THEY'RE ISN'T MUCH ELSE IN THE GARDEN	0.08
SORT OF SECOND HALF OF OCTOBER	('T', 'AO', 'W', 'T'), ('AH', 'F'), ('T', 'EY', 'K', 'AH', 'K', 'T'), ('K', 'EY', 'F'), ('AH', 'F'), ('AA', 'K', 'T', 'AO', 'P', 'ER')	('T', 'AO', 'W', 'T'), ('AH', 'F'), ('T', 'EY', 'K', 'AH', 'K', 'T'), ('K', 'EY', 'F'), ('AH', 'F'), ('AA', 'K', 'T', 'AO', 'P', 'ER')	SORT OF SECOND HALF OF OCTOBER	32.86	SORT OF SECOND HALF OF OCTOBER	0.05
WELL INTO NOVEMBER	('W', 'EY', 'K'), ('IY', 'K', 'T', 'UH'), ('K', 'AO', 'F', 'EY', 'P', 'P', 'ER')	('W', 'EY', 'K'), ('IY', 'K', 'T', 'UH'), ('K', 'AO', 'F', 'EY', 'P', 'P', 'ER')	RAN INTO NOVEMBER	2.83	WELL INTO NOVEMBER	0.03
WE CAN JUST ABOUT GET AWAY WITH IT NOW	('W', 'IY'), ('K', 'EY', 'K'), ('CH', 'AH', 'T', 'T'), ('AH', 'P', 'EY', 'T'), ('K', 'EY', 'T'), ('AH', 'W', 'EY'), ('W', 'IY', 'T'), ('IY', 'T',) ('K', 'EY')	('W', 'IY'), ('K', 'EY', 'K'), ('CH', 'AH', 'T', 'T'), ('AH', 'P', 'EY', 'T'), ('K', 'EY', 'T'), ('AH', 'W', 'EY'), ('W', 'IY', 'T'), ('IY', 'IY'), ('K', 'EY')	WE CAN JUST ABOUT GET AWAY WITH IIE KAYE	323.84	WE CAN JUST ABOUT GET AWAY WITH IT NOW	0.07
AND IF YOU WANT WONDERFUL	('AH', 'K', 'T'), ('IY', 'F'), ('K', 'UH'), ('W', 'AA', 'K', 'T'), ('W', 'AH', 'K', 'T', 'ER', 'F', 'AH', 'K')	('AH', 'K', 'T'), ('IY', 'F'), ('K', 'UH'), ('W', 'AA', 'K', 'T'), ('W', 'AH', 'K', 'T', 'ER', 'F', 'AH', 'K')	AND IF YOU WANT WONDERFUL	35.19	AND IF YOU WANT WONDERFUL	0.05
FOR A BRIEF TIME	('F', 'AO', 'W'), ('AH'), ('P', 'W', 'IY', 'F'), ('T', 'AH', 'P')	('AO', 'AO', 'W'), ('AH'), ('P', 'W', 'IY', 'F'), ('T', 'AH', 'P')	OR A BRIEF TIME	21.95	FOR A BRIEF TIME	0.05
IT WILL CHANGE LIVES	('IY', 'T'), ('W', 'IY', 'K'), ('CH', 'EY', 'K', 'CH'), ('K', 'IY', 'F', 'T')	('T', 'T'), ('W', 'IY', 'K'), ('CH', 'EY', 'K', 'CH'), ('K', 'IY', 'F', 'T')	THS WE'LL CHANGE LIFFE'S	27.53	THIS WILL CHANGE LIVES	0.05
I THINK IT'S BRILLIANT	('AH'), ('T', 'IY', 'K', 'K'), ('IY', 'T', 'T'), ('P', 'W', 'IY', 'K', 'K', 'AH', 'K', 'T')	('AH'), ('T', 'IY', 'K'), ('IY', 'T', 'T'), ('P', 'W', 'IY', 'K', 'K', 'AH', 'K', 'T')	EYE 'TIL IT'S PRINGLE'S	13.24	I THING IT'S BRILLIANT	0.05
BUT IT'S A DECENT SIZE	('P', 'AH', 'T'), ('IY', 'T', 'T'), ('AH'), ('T', 'IY', 'T', 'AH', 'K', 'T'), ('T', 'AH', 'T')	('P', 'AH', 'T'), ('IY', 'T', 'T'), ('AH'), ('T', 'IY', 'T', 'AH', 'K', 'T'), ('T', 'AH', 'T')	BUT IT'S I DIDN'T SUSS	79.88	BUT IT'S A DECENT SIZE	0.06

Table 5.8: Examples of decoded sentences from the two viseme-to-word converters.

Actual Subtitle	**Actual Visemes**	**Predicted Visemes**	**Hidden Markov Model**		**Feed-Forward Network**	
			Decoded Subtitle	**Execution Time(s)**	**Decoded Subtitle**	**Execution Time(s)**
WHEN THERE ISN'T MUCH ELSE IN THE GARDEN	('W', 'EY', 'K'), ('T', 'EY', 'W'), ('IY', 'T', 'AH', 'K', 'T'), ('P', 'AH', 'CH'), ('EY', 'K', 'T'), ('IY', 'K'), ('T', 'AH'), ('K', 'AA', 'W', 'T', 'AH', 'K')	('W', 'EY', 'K'), ('T', 'EY', 'W'), ('IY', 'T', 'AH', 'K', 'T'), ('P', 'AH', 'CH'), ('EY', 'K', 'T'), ('IY', 'K'), ('T', 'AH'), ('K', 'AA', 'W', 'T', 'AH', 'K')	WHEN THEY'RE ISN'T BE JUST IN THE GARDEN	0.07	WHEN THEY'RE ISN'T JUST BE IN THE GARDEN	0.08
SORT OF SECOND HALF OF OCTOBER	('T', 'AO', 'W', 'T'), ('AH', 'F'), ('T', 'EY', 'K', 'AH', 'K', 'T'), ('K', 'EY', 'F'), ('AH', 'F'), ('AA', 'K', 'T', 'AO', 'P', 'ER')	('T', 'AO', 'W', 'T'), ('AH', 'F'), ('T', 'EY', 'K', 'AH', 'K', 'T'), ('K', 'EY', 'F'), ('AH', 'F'), ('AA', 'K', 'T', 'AO', 'P', 'ER')	SORT I'VE SECOND HALF I'VE WHICH	0.04	SORT I SECOND HALF OF OCTOBER	0.05
WELL INTO NOVEMBER	('W', 'EY', 'K'), ('IY', 'K', 'T', 'UH'), ('K', 'AO', 'F', 'EY', 'P', 'P', 'ER')	('W', 'EY', 'K'), ('IY', 'K', 'T', 'UH'), ('K', 'AO', 'F', 'EY', 'P', 'P', 'ER')	WHEN INTO NOVEMBER	0.03	WHEN INTO NOVEMBER	0.04
WE CAN JUST ABOUT GET AWAY WITH IT NOW	('W', 'IY'), ('K', 'EY', 'K'), ('CH', 'AH', 'T', 'T'), ('AH', 'P', 'EY', 'T'), ('K', 'EY', 'T'), ('AH', 'W', 'EY'), ('W', 'IY', 'T'), ('IY', 'T',) ('K', 'EY')	('W', 'IY'), ('K', 'EY', 'K'), ('CH', 'AH', 'T', 'T'), ('AH', 'P', 'EY', 'T'), ('K', 'EY', 'T'), ('AH', 'W', 'EY'), ('W', 'IY', 'T'), ('IY', 'IY'), ('K', 'EY')	WE CAN JUST ABOUT GET AWAY WITH IT HOW	0.07	WE CAN JUST ABOUT GET AWAY WITH IT NOW	0.07
AND IF YOU WANT WONDERFUL	('AH', 'K', 'T'), ('IY', 'F'), ('K', 'UH'), ('W', 'AA', 'K', 'T'), ('W', 'AH', 'K', 'T', 'ER', 'F', 'AH', 'K')	('AH', 'K', 'T'), ('IY', 'F'), ('K', 'UH'), ('W', 'AA', 'K', 'T'), ('W', 'AH', 'K', 'T', 'ER', 'F', 'AH', 'K')	AND IF KNEW WANT WONDERFUL	0.04	AND IF YOU WANT WONDERFUL	0.05
FOR A BRIEF TIME	('F', 'AO', 'W'), ('AH'), ('P', 'W', 'IY', 'F'), ('T', 'AH', 'P')	('AO', 'AO', 'W'), ('AH'), ('P', 'W', 'IY', 'F'), ('T', 'AH', 'P')	FOR I THIS TYPE	0.04	FOR A BIG TYPE	0.05
IT WILL CHANGE LIVES	('IY', 'T'), ('W', 'IY', 'K'), ('CH', 'EY', 'K', 'CH'), ('K', 'IY', 'F', 'T')	('T', 'T'), ('W', 'IY', 'K'), ('CH', 'EY', 'K', 'CH'), ('K', 'IY', 'F', 'T')	THIS WILL CHANGE LIVES	0.04	THIS WILL CHANGE LIVES	0.05
I THINK IT'S BRILLIANT	('AH'), ('T', 'IY', 'K', 'K'), ('IY', 'T', 'T'), ('P', 'W', 'IY', 'K', 'K', 'AH', 'K', 'T')	('AH'), ('T', 'IY', 'K'), ('IY', 'T', 'T'), ('P', 'W', 'IY', 'K', 'K', 'AH', 'K', 'T')	I THINK IT'S BRILLIANT	0.04	I THINK IT'S BRILLIANT	0.04
BUT IT'S A DECENT SIZE	('P', 'AH', 'T'), ('IY', 'T', 'T'), ('AH'), ('T', 'IY', 'T', 'AH', 'K', 'T'), ('T', 'AH', 'T')	('P', 'AH', 'T'), ('IY', 'T', 'T'), ('AH'), ('T', 'IY', 'T', 'AH', 'K', 'T'), ('T', 'AH', 'T')	BUT IT'S A DECENT SUSS	0.05	BUT IT'S A DECENT SIZE	0.05

5.4 Summary

A viseme-to-word conversion model has been proposed that is robust, quick to execute and effective at discriminating between words that share identical visemes. Its performance has been compared with three other conversion model approaches. The model has been proven to be effective at disambiguating between words that are semantically and syntactically different as well as being able to model long and short term dependencies to make it robust to incorrectly classified visemes. The converter's robustness has been verified on the LRS2 and LRS3 corpuses; and when implemented in a neural network-based architecture for lip reading sentences from the LRS2 dataset, a 79.6% word accuracy rate is recorded - an improvement of 15.0% from the previous-state of art.

Future research includes improving the robustness of viseme-to-word conversion further by using techniques like augmentation in the training phase. Moreover, there are other types of networks that could be used to enhance the overall word accuracy further such as bidirectional RNNs as these can exploit right-to-left context, in addition to left-to-right context for word prediction. There is also merit in considering the use of either Attention-Transformers or Temporal Convolutional networks as conversion models because they can process inputs in parallels as opposed to RNNs which process inputs sequentially.

It would also be ideal if it were possible to exploit knowledge regarding words that either consist or do not consist of unique visemes sequences as has been done for the case of viseme-to-word conversion when the identity of the inputted visemes are known with absolute precision.

Chapter 6

Conclusions and Future Work

6.1 Conclusion of Thesis Achievements

Automated Lip Reading is a broad field with many components and branches, and it is a domain that has seen lots of interest and progress in recent years. The work documented in the thesis contributes towards the different classification schema that can be used in visual speech recognition by developing an machined-based lip reading system that predicts sentences in continuous speech by classifying visemes. The trends in automated lip-reading and work reported in this have helped to opened up a whole a new line of research and entirely new way of doing lip-reading has been explored.

A neural network-based lip reading system has been developed to predict sentences covering a wide range of vocabulary in silent videos from people speaking. The system is lexicon-free, uses only visual cues represented by visemes of a limited number of distinct lip movements, and is robust to different levels of lighting. The system was verified on the BBC LRS2 data set(an initial word accuracy rate of 64.6% had been been achieved which was further improved to 80%). The system has demonstrated a significant improvement on classification accuracy of words compared to the state-of-the-art works.

The proposed lip-reading system is not only effective decoding visemes in continuous speech

but it uses a language model that is effective discriminating between words that share identical visemes and is been proven to be theoretically effective as disambiguating between words that are semantically and syntactically different as well as being able to model long and short term dependencies to make it robust to incorrectly classified visemes. The viseme-to-word converter is not only effective at distinguishing between homopheme words and relatively robust to misclassified visemes, but it also quick to execute.

Chapter 2 reviewed all of the different components that make up automated lip-reading systems including visemes and phonemes, performance metrics, audio-visual databases, pre-processing feature extraction and classification networks and classification. Lip-reading systems have evolved significantly because of both the advances of neural networks in performing feature extraction and classification and because of the emergence of large-scale databases which means that there is the possibility to cover vocabularies with thousands of different words.

Chapter 3 started off with a summary of all of the trends in the evolution in automated lip-reading systems based on analysis of Chapter 2. Lip-reading systems have attained very good accuracies for predicting words with each words but the achievement of good accuracies when prediction entire sentences covering thousands of words has been more a challenge. The majority of lip-reading systems that are tasked to predict sentences use ASCII characters as a classification schema and this indicate a gap in lip-reading research that there are alternative schema to be considered including visemes. In the discussion about classification schema, one of the advantages of using visemes is fewer classes can be used compared with words, phonemes or ASCII character and viseme-based lip-reading system can also be lexicon-free and be generalised to be implemented on people speaking in different languages. There is even merit to decoding speech in two stages with a language model being used in the second stage regardless of the intermediate class in that the grammatical correctness of the predicted sentence can be enhanced. Language models for performing the viseme-to-word require context in being effective when predicting words and they must also be robust to the possibility of incorrectly classified visemes.

Chapter 4 proposed a lip-reading system that is lexicon-free, uses only visual cues represented

by visemes of a limited number of distinct lip movements, and is robust to different levels of lighting. Verified on the BBC LRS2 data set, the system has demonstrated a significant improvement on classification accuracy of words compared to the state-of-the-art works. The question of "Can a good classification performance of individual visemes be attained" was addressed for visemes in continuous speech from both frontal and profile viewpoints and where temporal alignments also needed to be performed as viseme boundaries were unknown. The viseme-to-word conversion model performed to near-perfect accuracy for words that have a unique set of visemes which was in theory to be expect; but for homopheme words, the conversion model was shown to be relatively effective at disambiguating between words that share identical visemes. The proposed lip-reading system was also demonstrated to have good generalisation capabilities when retrained on samples of data from the LRS2 data set whereby the ratio of testing to training samples was increased.

Chapter 5 proposed a viseme-to-word conversion model to address the question of "Can a language model be implemented that is robust to confused visemes?". Chapter 5 builds on the work reported in Chapter 4 by acknowledging the possibility of incorrectly classified visemes. The classification performance of the system in predicting sentences from videos from the benchmark BBC-LRS2 is constrained by the viseme-to-word conversion bottleneck with its sensitivity to incorrectly classified visemes. A 95% viseme classification accuracy only yields a 65% word classification accuracy though Chapter 5 proposes a conversion approach that is both more efficient in its conversion and more robust to the possibility of confused visemes.

The proposed model was an Attention-based GRU model shown to be effective a discriminating between words that share visemes compared with other possible approaches such as Hidden Markov Models and Feed-Forward Neural Networks due to its ability to exploit a larger context window in utilising syntactic and semantic information to distinguish between words. The Attention-based GRU model outperformed the model used in Chapter 4 in that is was shown to be more robust to incorrectly classified visemes. The conversion model in Chapter 5 is more effective a predicting spoken words correctly given that the recognised visemes may have been incorrectly classified and it is less vulnerable to the possibility that other words in the outputted sentence will have been predicted incorrectly given that previous words were misclassified.

The work reported in this thesis has addressed each one of the research questions posed as follows:

(**Q1**): What are the benefits of using visemes to lip-read?

- Visemes are the most fundamental units of visual speech and visual speech recognition is a task of significant importance when audio is unavailable or when there is noise.
- There are benefits to using visemes as classes for visual speech recognition including the opportunity to use fewer classes, the possibility of predicting words by classifying image-based classes without lexicons and the possibility to predict speech from people speaking in different languages.

(**Q2**): Can a lip-reading system accommodate for classification of visemes?

- The proposed lip-reading system consists of two components to classify speech in two stages.
- The first component classifies visemes, then the second component which is a viseme-to-word converter uses a trained language model to convert classified visemes to words.

(**Q3**): Can a good classification performance of individual visemes be attained?

- The viseme classifier that forms part of the lip-reading system proposed, performs classification for individual visemes in continuous speech to very good accuracy.
- It has shown to be robust to lighting variations as well as having a good generalisation capability.

(**Q4**): What are the different language models available?

- Different language models were explored and they can be grouped into three main categories: statistical language models, neural language models with fixed-context windows and neural language models with unlimited context.

- Neural language models with unlimited context are the most effective language models that can be used for viseme-to-word conversion because the exploitation of as much as context as possible is necessary to disambiguate between words sharing visemes.

(**Q5**): Can a language model be implemented that is effective at converting visemes to words?

- The language model proposed in the final chapter of this thesis performs viseme-to-word conversion with very good word classification performance.
- It is not only effective in its conversion of visemes to words for words that have a unique set of visemes but also for homopheme words too.

(**Q6**): Can a language model be implemented that is robust to confused visemes?

- The language model used of part of the proposed lip-reading system has been demonstrated to be somewhat robust to misclassified visemes as indicated by its the performance of its response to visemes of varying accuracies.

(**Q7**): Can a good overall performance be attained for word classification when predicting sentences?

- The overall word classification accuracy attained on the benchmark LRS2 corpus is superior to a previous state-of-the-art.

6.2 Future Work

Further work can be done to build on the lip reading system discussed in this thesis and a lip-reading system that uses solely visual cues and is lexicon-free would be convenient for speech decoding in real time.

The lip reading system is lexicon-free but the samples that were used did not consist of test words that were note present among the training samples. One limitation of lip reading system that use words or ASCII characters as a classification schema is their inability to predict words

that the system has not observed in training. A viseme-based lip reading system in theory could predict words that the lip-reading system did not encounter in the training phase by simply matching clusters of visemes to potential words, however, the system's ability and precision in accurately decoding such words has not been verified.

Because many language share identical visemes, there is a multitude of further work that can be done in using a viseme-based lip reading system which can be implemented to decoded people uttering words and sentences in other languages. A further feature for viseme-to-word conversion could be to have one lip reading system that could be applied to people speaking in more than one language.

The question over which intermediate classification schema is the best to use is itself an area that could involve more inquiry because even a lip-reading system that uses ASCII characters as an intermediate class can benefit from the use of language model so an ablation study could be conducted to compare visemes, phonemes and ASCII characters.

In addition to comparing classification scheme, there is also the possibility of developing a speech recognition system that combines visemes, phonemes and ASCII characters enhance through ensemble modelling to enhance the performance accuracy of a lip-reading system.

Finally, another trend that is being seen in automated lip reading is the emergence of end-to-end system where visual feature extraction can be applied with the need to physically located the region-of-interest because at present, the lip reading system proposed in the thesis requires to region-of-interest to be at the centre of every image frame within the video.

References

[1] T. Mohammed, R. Campbell, M. Macsweeney, F. Barry and M. Coleman. (2006). Speechreading and its association with reading among deaf, hearing and dyslexic individuals. Clinical Linguistics and Phonetics.

[2] A. Gabbay, A. Ephrat, T. Halperin, S. Peleg. (2017). Seeing through noise: speaker separation and enhancement using visually-derived speech. Proc. International Workshop on Computer Vision for Audio-Visual Media.

[3] D. Stewart, R. Seymour, A. Pass and J. Ming. (2014). Robust audio-visual speech recognition under noisy audio-video conditions, IEEE Trans. Cybern. 44 (2). 175–184.

[4] F.S. Lesani, F.F. Ghazvini, R. Dianat. (2015). Mobile phone security using automatic lip reading. Proc. International Conference on E-Commerce in Developing Countries: With Focus on e-Business pp. 1–5.

[5] E. T. Auer and L. E. Bernstein. (2007). Enhanced Visual Speech Perception in Individuals with Early-Onset Hearing Impairment. Journal of Speech, Language, and Hearing Research.

[6] R. Campbell and T. E. Mohammed. (2010). Speechreading for information gathering: a survey of scientific sources. Deafness Cognition and Language Research Centre.

[7] R. Bowden et al. (2013). Recent developments in automated lip-reading. Proceedings of SPIE - The International Society for Optical Engineering.

[8] M. Bohning et al. (2002). Audiovisual speech perception in Williams syndrome. Neuropsychologia.

[9] J. Leybaert *et al.* (2014). Atypical audio-visual speech perception and McGurk effects in children with specific language impairment. Front Psychol.

[10] W. H. Sumby and I. Pollack, (1954). Visual contribution to speech intelligibility in noise. Journal of the Acoustical Society of America. vol. 26, no. 2.

[11] E. D. Petajan (1984). Automatic lipreading to enhance speech recognition - PhD Dissertation. University of Illinois at Urbana-Champaign.

[12] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. (2011). Multimodal deep learning. Proceedings of the 28th International Conference on Machine Learning, ICML.

[13] H. Lee, C. Ekanadham, and A. Y. Ng. (2008). Sparse deep belief net model for visual area V2. Proceedings of Advances in Neural Information Processing Systems

[14] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno and T. Ogata. (2014). Lipreading using convolutional neural network. In Interspeech.

[15] M. Wand, J. Koutnik and J. Schmidhuber. (2016). Lipreading with long short term memory. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.

[16] Y. M. Assael, B. Shillingford, S. Whiteson and N. de Freitas. (2016). LipNet: End-to-End sentence Level Lipreading. ICLR Conference.

[17] J. S. Chung, A. Zisserman, A. Senior and O. Vinyals. (2016). Lip Reading Sentences in the Wild. IEEE Conference on Computer Vision and Pattern Recognition.

[18] A. B. Mattos, D. Oliveira and E. Morais. (2018). Improving Viseme Recognition Using GAN-Based Frontal View Mapping. Analysis and Modeling of Faces and Gestures (CVPR).

[19] A. J. Goldschen, O. N. Garcia and E. D. Petajan. (1997). Continuous automatic speech recognition by lipreading. In Motion-Based recognition.

[20] L. Cappelletta and N. Harte. (2012). Phoneme-to-viseme mapping forvisual speech recognition. In International Conference on Pattern Recognition Applications and Methods (ICPRAM), pages 322-329..

[21] B-J. Theobald. (2003). Visual Speech Synthesis Using Shape and Appearance Models. PhD thesis. University of East Anglia.

[22] F. DeLand. (1931). The story of lip-reading, its genesis and development.

[23] C. G. Fisher. (1968). Confusions among visually perceived consonants. Journal of Speech, Language, and Hearing Research.

[24] M. F. Woodward and C. G. Barber. (1960). Phoneme perception in lip-reading. Journal of Speech, Language, and Hearing Research, 3(3):212–222.

[25] J. C. Catford. (1977). Fundamental Problems in Phonetics. Bloomington. Indiana University.

[26] Amazon Polly. Phoneme/viseme tables for supported languages. https://docs.aws.amazon.com/polly/latest/dg/refphoneme-tables-shell.html. Accessed: 2018-02.

[27] P. Hanavan. (2020). Audiovisual Speech Perception.

[28] G. Potamianos, C. Neti, I. Matthews. (2004). Audio-visual automatic speech recognition: an overview, in: G. Bailly, E. Vatikiotis-Bateson, P. Perrier (Eds.). Issues in audio-visual speech processing, MIT Press.

[29] K. S. Talha et al. (2013). Speech Analysis Based On Image Information from Lip Movement. IOP Conference Series: Materials Science and Engineering.

[30] T. J. Hazen, K. Saenko, C. La and J. R. Glass. (2004). A segment-based audio-visual speech recognizer: data collection, development, and initial experiments. Proceedings of the 6th International Conference on Multimodal Interfaces.

[31] H.L. Bear, R. Harvey. (2016). Decoding visemes: improving machine lip-reading, Proceedings of International Conference on Acoustics. Speech and Signal Processing.

[32] J. Jeffers and M. Barley (1971). Speechreading (Lipreading). Charles C Thomas Publisher Limited.

[33] C. Neti et al. (2000). Audio visual speech recognition. Technical report IDIAP.

[34] E. Bozkurt, C. E. Erdem, E. Erzin, T. Erdem and M. Ozkan. (2007). Comparison of phoneme and viseme based acoustic units for speech driven realistic lip animation. In 3DTV Conference.

[35] S. Lee and D. Yook. (2002). Audio-to-Visual Conversion Using Hidden Markov Models. In Proceedings of the 7th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence.

[36] A. Botev, B. Zheng and D.Barber. (2017). Complementary sum sampling for likelihood approximation in large scale classification.

[37] M. Collins. (2011). Course Notes for COMS w4705: Language Modeling. Collins 2011 Course NF.

[38] J. R. Firth. (1957). A Synopsis of Linguistic Theory. Studies in Linguistic Analysis.

[39] G. Kondrak. (2000). A new algorithm for the alignment of phonetic sequences. Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference. Association for Computational Linguistics.

[40] T. Afouras, J. S. Chung and A. Zisserman. (2018). LRS3-TED: a large-scale dataset for visual speech recognition.

[41] B. Shillingford et al. (2018). Large-Scale Visual Speech Recognition.

[42] J. S. Chung and A. Zisserman. (2015). Lip Reading in the Wild. Asian Conference on Computer Vision.

[43] M. Igras, B. Ziolko and T. Jadczyk. (2012). Audiovisual database of Polish speech recordings. Studia Informatica.

[44] D. Estival, S. Cassidy, F. Cox and D. Burnham. (2014). AusTalk: an audio-visual corpus of Australian English, Proceedings of the International Conference on Language Resources and Evaluation.

[45] Y. Lu, H. Li. (2019). Automatic Lip-Reading System Based on Deep Convolutional Neural Network and Attention-Based Long Short-Term Memory. Applied Sciences.

[46] A. Ortega, F. Sukno, E. Lleida, A.F. Frangi, A. Miguel, L. Buera and E. Zacur. (2004). AV@CAR: a Spanish multichannel multimodal corpus for in-vehicle automatic audio-visual speech recognition. Proc. International Conference on Language Resources and Evaluation.

[47] S. Antar and A. Sagheer. (2013). Audio Visual Arabic Speech (AVAS) Database for Human-Computer Interaction Applications. The International Journal of Advanced Research in Computer Science and Software Engineering.

[48] B. Lee, M. Hasegawa-Johnson, C. Goudeseune, S. Kamdar, S. Borys, M. Liu and T.S. Huang. (2004). AVICAR: audio-visual speech corpus in a car environment. Proceedings of Interspeech.

[49] I. Matthews, T.F. Cootes, J.A. Bangham, S. Cox and R. Harvey. (2002). Extraction of visual features for lipreading. IEEE Transactions in Pattern Analysis and Machine Intelligence.

[50] S.J. Cox, R. Harvey, Y. Lan, J.L. Newman and B.J. Theobald. (2008). The challenge of multispeaker lip-reading. Proceedings of the International Conference on Auditory-Visual Speech Processing

[51] L. A. Elrefaei, T. Q. Alhassan and S. S. Omar. (2019). An Arabic Visual Dataset for Visual Speech Recognition. Procedia Computer Science.

[52] E. Bailly-Bailliere, et al. (2003). The BANCA database and evaluation protocol. Proc. International Conference on Audio- and Video-based Biometric Person Authentication.

[53] Y. Benezeth, G. Bachman, G. Le-Jan, N. Souviraà-Labastie, F. Bimbot. (2011). BL-Database: A French Audiovisual Database for Speech Driven Lip Animation Systems INRIA. Ph.D. Thesis.

[54] X. Yanjun, D. limin, L. guoqiang, Z. xin, and Z. zhi. (2000). Chinese audiovisual bimodal speeeh database CAVSR1.0. Acta Acustica.

[55] S. Tamura et al. (2010). CENSREC-1-AV: an audio-visual corpus for noisy bimodal speech recognition. Proc. International Conference on Auditory-Visual Speech Processing.

[56] K. Kumar, T. Chen and R. M. Stern. (2007). Profile View Lip Reading. IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP.

[57] E.K. Patterson, S. Gurbuz, Z. Tufekci, J.N. Gowdy. (2002). CUAVE: a new audio-visual database for multimodal human-computer interface research. Proceedings of International Conference on Acoustics, Speech, and Signal Processing.

[58] C.C. Chibelushi, F. Deravi and J.S. Mason. (1996). BT DAVID Database-Internal Rep. Speech and Image Processing Research Group, Dept. of Electrical and Electronic Engineering, University of Swansea.

[59] M. Cooke, J. Barker, S. Cunningham and X. Shao (2006). An audio-visual corpus for speech perception and automatic speech recognition. The Journal of the Acoustical Society of America.

[60] N. Alghamdi, S. Maddock, R. Marxer, J. Barker and G. J. Brown. (2018). A corpus of audio-visual Lombard speech with frontal and profile views. The Journal of the Acoustical Society of America.

[61] V. Verkhodanova, A. Ronzhin, I. Kipyatkova, D. Ivanko, A. Karpov and M. Zelezny. (2016). HAVRUS corpus: high-speed recordings of audio-visual Russian speech. Proc. International Conference on Speech and Computer.

[62] X. Line, H. Yao, X. Hong and Q. Wang. (2008). HIT-AVDB-II: A New Multi-view and Extreme Feature Cases Contained Audio-Visual Database for Biometrics. 10.2991/jcis.2008.61.

[63] Y. Mroueh, E. Marcheret and V. Goel. (2015). Deep multimodal learning for audio-visual speech recognition. Proceddings of the International Conference on Acoustics, Speech and Signal Processing.

[64] J. Huang, G. Potamianos, J. Connell, C. Neti. (2004). Audio-visual speech recognition using an infrared headset. Speech Comm.

[65] P.J. Lucey, G. Potamianos and S. Sridharan. (2008). Patch-based analysis of visual speech from multiple views. Proc. International Conference on Auditory-Visual Speech Processing.

[66] D. Petrovska-Delacretaz et al. (2008). The IV 2 multimodal biometric database (including iris, 2D, 3D, stereoscopic, and talking face data), and the IV 2-2007 evaluation campaign. Proc. International Conference on Biometrics: Theory, Applications and Systems.

[67] Y. Lan, B.-J. Theobald, R. Harvey, E.-J. Ong, R. Bowden. (2010). Improving visual features for lip-reading. Proceedings of International Conference on Auditory-Visual Speech Processing.

[68] S. Yang, Y. Zhang, D. Feng, M. Yang, C.Wang, J. Xiao, K. Long, S. Shan and X. Chen. (2019). LRW-1000: A naturally-distributed large-scale benchmark for lip reading in the wild. Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition.

[69] V. Estellers and J.P. Thiran. (2011). Multipose audio-visual speech recognition. Proceedings of 19th European Signal Processing Conference.

[70] O. Vanegas, K. Tokuda, and T. Kitamura. (1999). Location normalization of HMM-based lip-reading: Experiments for the M2 VTS database. Proceedings of International Conference of Image Processing.

[71] A. Rekik, A. Ben-Hamadou and W. Mahdi. (2014). A new visual speech recognition approach for RGB-D cameras. Proceedings of the International Conference on Image Analysis and Recognition.

[72] C. McCool et al. (2012). Bi-modal person recognition on a mobile phone: using mobile phone data, Proc. International Workshop on Multimedia and Expo.

[73] A. Czyzewski, B. Kostek, P. Bratoszewski, J. Kotus and M. Szykulski. (2017). An audio-visual corpus for multimodal automatic speech recognition. Journal of Intelligent Information Systems.

[74] J.S. Chung and A. Zisserman. (2017). Lip reading in profile. Proceedings of the British Machine Vision Conference.

[75] A.G. Chitu, K. Driel, L.J. Rothkrantz. (2010). Automatic lip reading in the Dutch language using active appearance models on high speed recordings, Proc. International Conference on Text, Speech and Dialogue.

[76] G. Zhao, M. Barnard, M. Pietikainen. (2009). Lipreading with local spatiotemporal descriptors. IEEE Trans. Multimedia.

[77] I. Anina, Z. Zhou, G. Zhao and M. Pietikainen. (2015). OuluVS2: a multi-view audiovisual database for non-rigid mouth motion analysis. Proc. International Conference on Automatic Face and Gesture Recognition.

[78] A. Pass, J. Zhang and D. Stewart. (2010). An investigation into features for multi-view lipreading, Proceedings of International Conference on Image Processing.

[79] D.L. Howell. (2015). Confusion Modelling for Lip-reading, University of East Anglia. Ph.D. Thesis.

[80] N. Harte and E. Gillen. (2015). TCD-TIMIT: an audio-visual corpus of continuous speech. IEEE Transactions in Multimedia.

[81] J. R. Movellan. (1994). Visual speech recognition with stochastic networks. Proceedings of Advances in Neural Information Processing Systems.

[82] Y.W. Wong, S.I. Chng, K.P. Seng, L.-M. Ang, S.W. Chin, W.J. Chew and K.H. Lim. (2011). A new multi-purpose audio-visual UNMC-VIER database with multiple variabilities. Pattern Recognition

[83] P. Cisar, M. Zelezny, Z. Krnoul, J. Kanis, J. Zelinka and L. Muller. (2005). Design and recording of Czech speech corpus for audio-visual continuous speech recognition. Proceedings of Auditory-Visual Speech Process International Conference.

[84] J. Trojanova, M. Hruz, P. Campr and M. Zelezny. (2008). Design and recording of Czech audio-visual database with impaired conditions for continuous speech recognition. Proc. International Conference on Language Resources and Evaluation.

[85] N.A. Fox, B.A. O'Mullane, R.B. Reilly. (2005). VALID: a new practical audio-visual database, and comparative results, Proc. International Conference on Audio and Video-based Biometric Person Authentication.

[86] C. Sanderson. (2002). The VidTIMIT database. IDIAP.

[87] A. Fernandez-Lopez, O. Martinez and F.M. Sukno. (2017). Towards estimating the upper bound of visual-speech recognition: the visual lip-reading feasibility database. Proceedings of International Conference on Automatic Face and Gesture Recognition.

[88] A. Vorwerk X. Wang, D. Kolossa, S. Zeiler and R. Orglmeister. (2010). WAPUSK20 - a database for robust audiovisual speech recognition. Proceedings of the International Conference on Language Resources and Evaluation.

[89] K. Messer, J. Matas, J. Kittler, J. Luettin and G. Maitre. (1999). XM2VTSDB: the extended M2VTS database, Proc. International Conference on Audio and Video-based Biometric Person Authentication.

[90] X. Chen, J. Du, H. Zhang. (2020). Lipreading with DenseNet and resBi-LSTM. Signal Image and Video Processing. 981-989.

[91] S. Petridis, J. Shen, D. Cetin and M. Pantic. (2018). Visual-only recognition of normal, whispered and silent speech. Proceedings of the International Conference on Acoustics, Speech and Signal Processing.

[92] H. Lane, B. Tranel. (1971). The Lombard sign and the role of hearing in speech. Journal of Speech Language Hearing.

[93] V. Zue, S. Sene and J. Glass. (1990). Speech database development: TIMIT and beyond, Speech Communications.

[94] R. Goecke and J.B. Millar. (2004). The audio-video Australian English speech data corpus AVOZES. Proc. International Conference on Spoken Language Processing.

[95] G. Papandreou, A. Katsamanis, V. Pitsikalis and P. Maragos. (2008). Adaptive multimodal fusion by uncertainty compensation with application to audio-visual speech recognition. Proceedings of International Conference on Multimodal Processing and Interaction.

[96] Y. Lan, B. J. Theobald and R. Harvey. (2012). View independent computer lip-reading. Proceedings of IEEE International Conference Multimedia Expo.

[97] M. Hao, M. Mamut, N. Yadikar, A. Aysa and K. Ubul. (2020). A Survey of Research on Lipreading Technology. IEEE Access, vol. 8.

[98] S. Dupont and J. Luettin. (2000). Audio-visual speech modeling for continuous speech recognition. IEEE Transactions on Multimedia.

[99] Z. Zhou, G. Zhao, X. Hong and M. Pietikainen. (2014). A review of recent advances in visual speech decoding. Image and vision computing.

[100] S.L. Phung, A. Bouzerdoum, D. Chai and A. Watson. (2004). Naive Bayes face-nonface classifier: a study of preprocessing and feature extraction techniques. Proceedings of International Conference on Image Processing.

[101] M. Saaidia, A. Chaari,L. Sylvie, V. Vigneron and M. Bedda. (2007). Face localization by neural networks trained with Zernike moments and Eigenfaces feature vectors - A comparison. IEEE International Conference on Advanced Video and Signal based Surveillance.

[102] Q.M. Rizvi. (2011). A Review on Face Detection Methods. Journal of Management Development and Information Technology.

[103] SJ. Lee, SB. Jung, JW. Kwon and SH. Hong. (1999). Face detection and recognition using PCA. TENCON 99: Proceedings of the IEEE Region 10 Conference.

[104] P. Ma, S. Petridis and Maja Pantic. (2021). End-to-End Audio-visual Speech Recognition with Conformers. ICASSP.

[105] T. Afouras, J. S. Chung and A. Zisserman. (2018). Deep lip reading: a comparison of models and an online application. Proceedings of Interspeech.

[106] X. Li, T. Zhang, X. Zhao et al. (2020). Guided autoencoder for dimensionality reduction of pedestrian features. Applied Intelligence 50.

[107] S. Petridis, Z. Li and M. Pantic. (2017). End-to-end visual speech recognition with LSTMs. Proceedings of the International Conference on Acoustics, Speech and Signal Processing.

[108] S. Petridis, Y. Wang, Z. Li and M. Pantic. (2017). End-to-end audiovisual fusion with LSTMs. Proceedings of International Conference on Auditory-Visual Speech Processing.

[109] S. Petridis, Y. Wang, Z. Li and M. Pantic. (2017). End-to-end multi-view lipreading. Proceedings of the British Machine Vision Conference.

[110] J. Masci, U. Meier, D. Ciresan and J. Schmidhuber. (2011). Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction. 52-59.

[111] A. Krizhevsky, I. Sutskever, and G. E. Hinton. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, pp. 1097–1105.

[112] J. S. Chung and A. Zisserman. (2016). Out of time: Automated lip sync in the wild. in Proc. Asian Conf. Comput. Vis. Cham, Switzerland: Springer,

[113] D. Lee, J. Lee and K.-E. Kim. (2016). Multi-view automatic lip-reading using neural network. Proceedings of Asian Conf. Comput. Vis. Cham, Switzerland: Springer.

[114] X. Zhang, H. Gong, X. Dai, F. Yang, N. Liu, and M. Liu, (2019). Understanding pictograph with facial features: End-to-end sentence-level lip reading of Chinese. Proceedings of AAAI Conference on Artificial Intelligence.

[115] Y. Lu, S. Yang, Z. Xu and J. Wang. (2020). Speech Training System for Hearing Impaired Individuals Based on Automatic Lip-Reading Recognition. Proceedings of the AHFE 2020 Virtual Conference on Human Factors and Systems Interaction.

[116] A. Garg, J. Noyola, and S. Bagadia. (2016). Lip reading using CNN and LSTM. Technical report Stanford University - CS231n project report.

[117] T. Saitoh, Z. Zhou, G. Zhao and M. Pietikainen. (2016). Concatenated frame image based cnn for visual speech recognition. Proc. Asian Conf. Comput. Vis. Cham, Switzerland: Springer.

[118] M. Lin, Q. Chen, and S. Yan, (2013). Network in network. arXiv:1312.4400. Available: http://arxiv.org/abs/1312.4400

[119] A. Mesbah, A. Berrahou, H. Hammouchi, H. Berbia, H. Qjidaa and M. Daoudi, (2019). Lip reading with hahn convolutional neural networks. Image and Vision Computing.

[120] Y. Li, Y. Takashima, T. Takiguchi and Y. Ariki. (2016). Lip reading using a dynamic feature of lip images and convolutional neural networks. Proceedings of IEEE/ACIS 15th Int. Conf. Comput. Inf. Sci. (ICIS).

[121] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei. (2014). Large-scale video classification with convolutional neural networks. Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 1725–1732.

[122] S. Ji, W. Xu, M. Yang and K. Yu. 3d convolutional neural networks for human action recognition. (2013). IEEE transactions on pattern analysis and machine intelligence, 35(1):221–231.

[123] A. Torfi, S. M. Iranmanesh, N. Nasrabadi, and J. Dawson. (2017). 3D convolutional neural networks for cross audio-visual matching recognition. IEEE Access, vol. 5.

[124] D. Bahdanau, K. Cho and Y. Bengio. (2014). Neural machine translation by jointly learning to align and translate. arXiv:1409.0473.

[125] K. Xu, D. Li, N. Cassimatis and X. Wang. (2018). LCANet: end-to-end lipreading with cascaded attention-CTC. Proceedings of the International Conference on Automatic Face and Gesture Recognition.

[126] R. K. Srivastava, K. Greff and J. Schmidhuber. (2015). Training very deep networks. Advances in Neural Information Processing Systems.

[127] T. Stafylakis and G. Tzimiropoulos, (2017). Combining residual networks with LSTMs for lipreading. Proceedings of Interspeech.

[128] T. Stafylakis and G. Tzimiropoulos. (2018). Deep word embeddings for visual speech recognition. IEEE International Conference on Acoustic Speech Signal Processing (ICASSP).

[129] D. Kumar Margam, R. Aralikatti, T. Sharma, A. Thanda, P. A K, S. Roy and S. M. Venkatesan. (2019). Lip Reading with 3D-2D-CNN BLSTM-HMM and word-CTC models. arXiv:1906.12170.

[130] A. Fernandez-Lopez and F. Sukno. (2018). Survey on Automatic Lip-Reading in the Era of Deep Learning. Image and Vision Computing. 78.

[131] S. Hochreiter and J. Schmidhuber (1997). Long short-term memory. Neural Comput. vol. 9.

[132] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. Proceedings of EMNLP.

[133] S. Fenghour, D. Chen, K. Guo, B. Li and P. Xiao. (2021). Deep Learning-Based Automated Lip-Reading: A Survey," in IEEE Access, vol. 9, pp. 121184-121205, doi: 10.1109/ACCESS.2021.3107946.

[134] Lipton, Zachary. (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning.

[135] J. Chung, C. Gulcehre, K. Cho and Y. Bengio. (2014). Empirical evaluation of gated recurrent neural networks on sequence modelling. arXiv preprint arXiv:1412.3555.

[136] A. Graves, S. Fernandez, F. Gomez and J. Schmidhuber. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. International Conference on Machine Learning pages 369-376.

[137] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. (2017). Attention Is All You Need. NIPS.

[138] B. Martinez, P. Ma, S. Petridis and M. Pantic. (2020). Lipreading using Temporal Convolutional Networks.

[139] S. Petridis, T. Stafylakis, P. Ma, F. Cai, G. Tzimiropoulos and M. Pantic. (2018). End-to-End audiovisual speech recognition. IEEE International Conference on Acoustic Speech Signal Processing.

[140] J. Huang and B. Kingsbury. (2013). Audio-visual deep learning for noise robust speech recognition. Proc. International Conference on Acoustics, Speech and Signal Processing.

[141] S. Moon, S. Kim, H. Wang. (2015). Multimodal transfer deep learning with applications in audio-visual recognition, MMML Workshop at Neural Information Processing Systems.

[142] K. Thangthai, R. Harvey, S. Cox and B. J. Theobald. (2015). Improving Lip-reading Performance for Robust Audiovisual Speech Recognition using DNNs. Proceedings of the International Conference on Auditory-Visual Speech Processing.

[143] I. Almajai, S. Cox, R. Harvey and Y. Lan. (2016). Improved speaker independent lip reading using speaker adaptive training and deep neural networks. Proceedings of International Conference on Acoustics, Speech and Signal Processing.

[144] S. Petridis, M. Pantic. (2016). Deep complementary bottleneck features for visual speech recognition, Proc. International Conference on Acoustics, Speech and Signal Processing.

[145] M. Wand, J. Schmidhuber. (2017). Improving speaker-independent lipreading with domain-adversarial training, Proceedings of Interspeech.

[146] S. NadeemHashmi, H. Gupta, D. Mittal, K. Kumar, A. Nanda and S. Gupta. (2018). A Lip Reading Model Using CNN with Batch Normalization. 11th International Conference on Contemporary Computing.

[147] M. Wand, N.T. Vu and J. Schmidhuber. (2018). Investigations on end-to-end Audiovisual fusion. Proceedings of the International Conference on Acoustics, Speech and Signal Processing.

[148] T. Afouras, J. S. Chung, A. Senior, O. Vinyals and A. Zisserman. (2018). Deep audio-visual speech recognition. IEEE Transactions in Pattern Analysis and Machine Intelligence.

[149] A. B. Mattos, D. Oliveira and E. Morais. (2018). Improving CNN-based Viseme Recognition Using Synthetic Data. 10.1109/ICME.2018.8486470.

[150] L. Courtney and R. Sreenivas. (2019). Learning from videos with deep convolutional LSTM networks. arXiv:1904.04817.

[151] D.-W. Jang, H.-I. Kim, C. Je, R.-H. Park, and H.-M. Park. (2019). Lip reading using committee networks with two different types of concatenated frame images. IEEE Access, vol. 7.

[152] P. Zhou, W. Yang, W. Chen, Y. Wang, J. Jia. (2019). Modality attention for end-to-end audio-visual speech recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

[153] X. Weng and K. Kitani. (2019). Learning spatio-temporal features with two-stream deep 3D CNNs for lip-reading. British Machine Vision Conference.

[154] C. Wang. (2019). Multi-grained spatio-temporal modelling for lip-reading. British Machine Vision Conference.

[155] Y. Zhang, S. Yang, J. Xiao, S. Shan, and X. Chen. (2020). Can we read speech beyond the lips? Rethinking RoI selection for deep visual speech recognition. Proceedings of 15th IEEE International Conference on Automatic Face Gesture Recognition (FG).

[156] J. Xiao, S. Yang, Y. Zhang, S. Shan, and X. Chen. (2020). Deformation flow based two-stream network for lip reading. Proceedings of 15th IEEE International Conference on Automatic Face Gesture Recognition (FG).

[157] M. Luo, S. Yang, S. Shan, and X. Chen. (2020). Pseudo-convolutional policy gradient for sequence-to-sequence lip-reading. Proceedings of 15th IEEE International Conference on Automatic Face Gesture Recognition.

[158] X. Zhao, S. Yang, S. Shan, and X. Chen. (2020). Mutual information maximization for effective lip reading. Proceedings of 15th IEEE International Conference on Automatic Face Gesture Recognition (FG).

[159] S. Fenghour, D. Chen, K. Guo and P. Xiao, (2020). Lip Reading Sentences Using Deep Learning With Only Visual Cues. IEEE Access, vol. 8.

[160] P Ma, Y Wang, J Shen, S Petridis, M Pantic. (2021). Lip-reading with Densely Connected Temporal Convolutional Networks. WACV.

[161] P Ma, B Martinez, S Petridis, M Pantic. (2021). Towards Practical Lipreading with Distilled and Efficient Models. ICASSP.

[162] K. R. Prajwal, T. Afouras and A. Zisserman. (2021). Sub-word Level Lip Reading With Visual Attention. ArXiv abs/2110.07603

[163] A. Piktus, N. B. Edizel, P. Bojanowski, E. Grave, R. Ferreira and F. Silvestri. (2019). Misspelling Oblivious Word Embeddings. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 3226–3234.

[164] Y. Pei, T.-K. Kim and H. Zha. (2013) Unsupervised random forest manifold alignment for lipreading. Proceedings of IEEE International Conference on Computer Vision.

[165] D. Hu et al. (2016). Temporal multimodal learning in audiovisual speech recognition, Proc. Conference on Computer Vision and Pattern Recognition.

[166] G. Papandreou, A. Katsamanis, V. Pitsikalis and P. Maragos, (2009). Adaptive multimodal fusion by uncertainty compensation with application to audiovisual speech recognition, IEEE-ACM Transactions in Audio, Speech and Language Processing

[167] M.H. Rahmani, F. Almasganj. (2017). Lip-reading via a DNN-HMM hybrid system using combination of the image-based and model-based features. Proceedings of International Conference on Pattern Recognition and Image Analysis.

[168] P. Wu, H. Liu, X. Li, T. Fan and X. Zhang. (2016). A novel lip descriptor for audio-visual keyword spotting based on adaptive decision fusion. IEEE Transactions on Multimedia.

[169] Y. Bengio et al. (2009). Curriculum learning. Proceedings of the 26th annual international conference on machine learning.

[170] L. Elman. (1993). Learning and development in neural networks : the importance of starting small. In: 48, pp. 71-99.

[171] K. Saenko, K. Livescu, J. Glass, and T. Darrell. (2005). Production Domain Modeling Of Pronunciation For Visual Speech Recognition. ICASSP.

[172] O. Koller, H. Ney, and R. Bowden. (2015). Deep learning of mouth shapes for sign language. IEEE International Conference on Computer Vision Workshop (ICCVW).

[173] K. Thangthai, H. L. Bear and R. Harvey. (2017). Comparing phonemes and visemes with DNN-based lipreading. 28th British Machine Vision Conference.

[174] H. L. Bear, R. Harvey, B. J. Theobald, and Y. Lan. (2104). Resolution limits on visual speech recognition. In 2014 IEEE International Conference on Image Processing (ICIP).

[175] D. Howell, S. Cox and B. Theobald. (2016). Visual Units and Confusion Modelling for Automatic Lip-reading. Image and Vision Computing. 51. 10.1016/j.imavis.2016.03.003.

[176] K. Thangthai and R. Harvey. (2017). Improving Computer Lipreading via DNN Sequence Discriminative Training Techniques. 10.21437/Interspeech.2017-106.

[177] M. Mohri, F. Pereira and M. Riley. (2002) Weighted finite-state transducers in speech recognition. Computer Speech and Language.

[178] J. Kun, J. Xu, and B. He. (2019). A Survey on Neural Network Language Models.

[179] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. (2003). A neural probabilistic language model. Journal of Machine Learning Research.

[180] F. Rahutomo, T. Kitasuka, M. Aritsugi. (2012). Semantic Cosine Similarity. The 7th International Student Conference on Advanced Science and Technology ICAST.

[181] S. Fenghour, D. Chen and P. Xiao. (2019). Decoder-Encoder LSTM for Lip Reading. Conference: 8th International Conference on Software and Information Engineering (ICSIE).

[182] Y. Lan, R. Harvey and B. J. Theobald. (1988). Insights into machine lip reading. International Conference on Acoustics, Speech and Signal Processing.

[183] A. Radford, K. Narasimhan, T. Salimans and I. Sutskever. (2018). Improving Language Understanding by Generative Pre-Training.

[184] G. Sterpu and N. Harte. (2018). Towards Lipreading Sentences with Active Appearance Models.

[185] J. Peymanfard, M. R. Mohammadi, H. Zeinali and N. Mozayani. (2021). Lip reading using external viseme decoding. arXiv preprint arXiv:2104.04784.

[186] L. Lamel, R, H. Kassel and S. Seneff. (1989). Speech database development: Design and analysis of the acoustic-phonetic corpus. Proceedings of the DARPA Speech Recognition Workshop.

[187] B. Zhang, D. Xiong, and J. Su. (2017). A GRU-Gated Attention Model for Neural Machine Translation. IEEE Transactions on Neural Networks and Learning Systems.

[188] H. Schwenk. (2007). Continuous space language models. Computer Speech and Language. 21(3).

[189] A. Kuncoro, C. Dyer, J. Hale, D. Yogatama, S. Clark and P. Blunsom. (2018). LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modeling Structure Makes Them Better.. 10.18653/v1/P18-1132.

[190] T. Linzen, E. Dupoux and Y. Goldberg. (2016). Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. Transactions of the Association for Computational Linguistics.

[191] A. Handler, M. Denny, H. Wallach and B. O'Connor. (2016). Bag of What? Simple Noun Phrase Extraction for Text Analysis. 114-124. 10.18653/v1/W16-5615.

[192] P. F. Brown et al. (1992). An Estimate of an Upper Bound for the Entropy of English. Computational Linguistics.

[193] A. J. Goldschen, O. N. Garcia and E. D. Petajan. (1996). Rationale for phoneme-viseme mapping and feature selection in visual speech recognition. Speechreading by Humans and Machines.

[194] Y. Fu, X. Zhou, M. Liu, M. Hasegawa-Johnson and T.S. Huang. (2007). Lipreading by locality discriminant graph. Proceedings of the International Conference on Image Processing.

[195] P.J. Lucey, G. Potamianos and S. Sridharan. (2007). A unified approach to multi-pose audio-visual ASR, Proceedings of Interspeech.

[196] E. Marcheret, V. Libal and G. Potamianos. (2007). Dynamic stream weight modeling for audio-visual speech recognition. Proceedings of the International Conference on Acoustics, Speech and Signal Processing.

[197] S.J. Cox, R. Harvey, Y. Lan, J.L. Newman and B.J. Theobald. (2008). The challenge of multispeaker lip-reading. Proceedings of the International Conference on Auditory-Visual Speech Processing.

[198] H. Hofmann, S. Sakti, R. Isotani, H. Kawai, S. Nakamura and W. Minker. (2010). Improving spontaneous English ASR using a joint-sequence pronunciation model. 4th International Universal Communication Symposium, Beijing.

[199] K. Saenko, K. Livescu, M. Siracusa, K. Wilson, J. Glass, and T. Darrell. (2005). Visual speech recognition with loosely synchronized feature streams. In Tenth IEEE International Conference on Computer Vision (ICCV).

[200] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu and A.C. Berg. (2016). Ssd: Single shot multibox detector. Proceedings of ECCV. pp. 21-37. Springer.

[201] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou and M. Pantic. (2013). 300 faces in-the-wild challenge: The first facial landmark localization challenge. In IEEE International Conference on Computer Vision Workshops.

[202] W. B. Dolan and C. Brockett. (2005). Automatically constructing a corpus of sentential paraphrases. In Proceedings of the Third International Workshop on Paraphrasing.

[203] S. Gray, A. Radford, and K. P. Diederik. (2017). Gpu kernels for block-sparse weights.

[204] R. Treiman, B. Kessler and S. Bick. (2001). Context sensitivity in the spelling of English vowels. Journal of Memory and Language.

[205] D. P. Kingma and J. Ba. (2015). Adam: A method for stochastic optimization. Proceedings of ICLR.

[206] V. I. Levenshtein. (1966). Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady.

[207] S. Fenghour, D. Chen, P. Xiao and K. Guo. (2020). Disentangling Homophemes in Lip Reading using Perplexity Analysis.

[208] C.E. Shannon. (1948). A Mathematical Theory of Communication. The Bell System Technical Journal.

[209] S. Vogel, H. Ney and C. Tillmann. (1996). HMM-Based Word Alignment in Statistical Translation.. 836-841. 10.3115/993268.993313.

[210] J. Wei and K. Zou. (2019). EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).

[211] D. Ataman, O. Firat, M. Gangi, M. Federico and A. Birch. (2019). On the Importance of Word Boundaries in Character-level Neural Machine Translation. 187-193. 10.18653/v1/D19-5619.

[212] V. I. Spitkovsky, H. Alshawi, and D. Jurafsky. 2010. From baby steps to leapfrog: How less is more in unsupervised dependency parsing. In Proceedings of NAACL, pages 751-759.

[213] Y. Tsvetkov, M. Faruqui, W. Ling, B. Macwhinney and C. Dyer. (2016). Learning the Curriculum with Bayesian Optimization for Task-Specific Word Representation Learning. 130-139. 10.18653/v1/P16-1013.