Computer lipreading via hybrid deep neural network hidden Markov models

Kwanchiva Thangthai

This thesis is submitted for the
Degree of Doctor of Philosophy

School of Computing Science
University of East Anglia

November 2018

Abstract

Constructing a viable lipreading system is a challenge because it is claimed that only 30% of information of speech production is visible on the lips. Nevertheless, in small vocabulary tasks, there have been several reports of high accuracies. However, investigation of larger vocabulary tasks is rare.

This work examines constructing a large vocabulary lipreading system using an approach based-on Deep Neural Network Hidden Markov Models (DNN-HMMs). We present the historical development of computer lipreading technology and the state-of-the-art results in small and large vocabulary tasks. In preliminary experiments, we evaluate the performance of lipreading and audiovisual speech recognition in small vocabulary data sets. We then concentrate on the improvement of lipreading systems in a more substantial vocabulary size with a multi-speaker data set. We tackle the problem of lipreading an unseen speaker. We investigate the effect of employing several steps to pre-process visual features. Moreover, we examine the contribution of language modelling in a lipreading system where we use longer n-grams to recognise visual speech. Our lipreading system is constructed on the 6000-word vocabulary TCD-TIMIT audiovisual speech corpus. The results show that visual-only speech recognition can definitely reach about 60% word accuracy on large vocabularies. We actually achieved a mean of 59.42% measured via three-fold cross-validation on the speaker independent setting of the TCD-TIMIT corpus using Deep autoencoder features and DNN-HMM models. This is the best word accuracy of a lipreading system in a large vocabulary task reported on the TCD-TIMIT corpus. In the final part of the thesis, we examine how the DNN-HMM model improves lipreading performance. We also give an insight into lipreading by providing a feature visualisation. Finally, we present an analysis of lipreading results and suggestions for future development.

Table of contents

List of abbreviations

Acronyms

AAM	Active appearance model
ASR	Automatic speech recognition
AVSR	Audio visual speech recognition
CCA	Canonical correlation analysis
CD	Contrastive divergence
CE	Cross-entropy
CNN	Convolutional neural network
CTC	Connectionist temporal classification
DAE	Deep autoencoders
DBMs	Deep Boltzmann machines
DBN	Deep belief network
DCT	Discrete cosine transform
DNN	Deep neural network
DNN-HMMs	Deep neural network hidden Markov models
DTCWT	Dual-tree complex wavelet transform
DWT	Discrete wavelet transform
EI	Early integration

EM	Expectation-maximisation
FDCT	Fast discrete curvelet transforms
fMLLR	Feature space maximum likelihood linear regression
GANs	Generative adversarial networks
GMM	Gaussian mixture model
HiLDA	Hierarchical linear discriminant analysis
HMM	Hidden Markov model
ICA	Independent component analysis
IPA	International phonetic alphabet
LDA	Linear discriminant analysis
LI	Late integration
LM	Language model
LSTM	Long short-term memory
LVCSR	Large vocabulary continuous speech recognition
MFCC	Mel-frequency cepstral coefficients
MLLR	Maximum likelihood linear regression
MLLT	Maximum likelihood linear transform
MMI	Maximum mutual information
MP	Matched pair sentence segment error
MPE	Minimum phone error
MSE	Mean square error
PCA	Principal components analysis
PDF	Probability density function

PER	Phoneme error rate
RNN	Recurrent neural networks
ROI	Region-of-interest
ROVER	Recogniser output voting error reduction
SAT	Speaker adaptive training
SGD	Stochastic gradient descent
sMBR	State-level minimum Bayes Risk
SNR	Signal-to-noise ratio
SP	Signed paired comparison
t-SNE	t-Distributed Stochastic Neighbour Embedding
TDNN	Time delay neural network
WER	Word error rate
WFST	Weighted finite-state transducer
WI	Wilcoxon signed-rank test

List of figures

List of tables

Acknowledgements

I would like to thank my advisor, Prof. Richard Harvey, for his invaluable guidance. I am grateful to all of my supervisory team, Dr. Ben Milner, Dr. Barry-John Theobald, Prof. Stephen Cox. Without you, this thesis would not have been possible.

A very special gratitude to the Office of the Civil Service Commission (OCSC) and the Royal Thai government for funding my PhD.

I would like to thank my examiners Assoc.Prof. Gerasimos Potamianos, and Dr. Gavin Cawley, for your comments, suggestions and the once-in-a-lifetime joy moment during my Viva. With a special mention to Dr. Dominic Howell, Dr. Helen L. Bear, Dr. Sarah Taylor, Dr. Yuxuan Lan, all CMP and HPC staff for all your help. Thank you to all my friends at UEA, Dr. Aki, Danny, Seth, Dr. Tom, Zhuoyi, and everyone in the UEA Thai-Soc 2014-2018. I am grateful to 5clubs (Aor, Fai, Nan, Yui) and my colleague at NECTEC. Special thanks to my boss, Dr. Chai Wuttiwiwatchai, you are the best boss, work-coach, and life-coach ever.

Finally, I would like to thank you to my family members, my mom, Sommai Saykham, my dad, Berm Saykham, my husband Dr. Ausdang Thangthai and his family, for your unconditional love, support, and encouragement. Thanks to my younger brother Aof and his wife, Tidtee, you are amazing.

Thanks for all your encouragement!

Chapter 1

Introduction

Automatic speech recognition (ASR) systems are ubiquitous. Voice has become the primary input method in many existing products, especially in a group of smart devices and smart assistants such as Google Assistant, Amazon Alexa, Microsoft Cortana and Apple Siri. At Google I/O 2018, Google announced a new product called Google Duplex which is a human-like smart assistant that can book a restaurant or hair salon by making a phone call and talking to staff like a human. This product involves many complex systems such as text-to-speech (TTS), natural language understanding, natural conversation and, of course, speech recognition. After the first ASR system was announced in 1952, it took more than half a century of intensive research and development to bring ASR to face the real market. Nowadays, with the power of deep learning and massive datasets, voice input exceeds human-level performance in various benchmarks [5, 159, 136]. In English, speech input is three times faster than typing. It is very remarkable that ASR beats human performance in a natural conversational speech task such as CallHome and Switchboard. However, the recording conditions of these corpora are quite ideal: close-talking and hence a high signal-to-noise ratio (SNR). Speech input is still challenging in some situations in real-world scenarios such as speech in the far field, and under cocktail party noise. It is also challenging to use ASR in a place where the SNR is extremely low such as music concerts, football games and so on. Plus, ASR needs to be a real-time recogniser in those examples. Indeed, a human is still superior to a machine in those circumstances.

Human speech perception is bimodal. That visual information affects human speech perception has been known for several decades. An excellent illustration of this effect is provided in a study by McGurk and MacDonald [87] who reported that people tend to hear the sound /da/ when the video of the lip movements of the syllable /ga/ is played alongside the speech sound /ba/ (The McGurk Effect). Moreover, research from Summerfield [142], shows that visual information contains three important clues for human speech perception: speaker localisation, speech segmental information and the place of articulation for certain phonemes. For instance, visual speech, which provides information about lip movements can help to distinguish between the acoustic confusion of consonant /m/ and /n/ [83]. There is a great deal of additional evidence about the use of visual speech in human speech perception. We cite, for example, supporting sign language in the hearing-impaired [82, 14] and obtaining better understanding in second language listeners [94]. Experiments over many years have shown that intelligibility scores are higher in noisy conditions if visual information, as well as audio information, are available [141, 142, 86]. Consequently, visual information has also been integrated into ASR systems to improve the robustness to acoustic noise [114, 52]. However, current commercial speech recognition systems use only acoustic speech. Thus, it would be useful to study the visual speech modality for increasing speech recognition system performance or as a replacement for situations when there is no acoustic signal available: this is known as computer lipreading.

Computer lipreading is a speech recognition system that extends the ability to work without speech sounds by using a visual form of speech such as the visibility of lips, teeth, tongue and lower face. It enables a normal speech recogniser to work even without a perfect speech sound. There are applications that could potentially benefit from a lipreading system. The first example is to use computer lipreading to aid forensic lipreading. In forensic lipreading, an expert speechreader who has intensive training in a particular language analyses output from a silent CCTV camera and produces a transcription of what was said. The second example of lipreading application is to use as a silent speech interface. This lipreading interface allows a user to command and access devices using their lip movements without making any audible sounds. In a situation where we want to interact with a device with confidential information such as password, personal information, sensitive information, a private method is needed. This silent input method could be used as for hands-free privacy and security information transmission in a military setting. The third example is to use as an assessment for lipreading learners. This type of application can help students learn and practice their lipreading and talking skills. The final example is to use as an audiovisual interface to enhance speech recognition systems. However, these applications require an accurate lipreading system that works well for a large vocabulary continuous speech recognition (LVCSR) task and real environments.

The challenges of lipreading

While modern-day speech recognition systems are nowadays commercially utilised, lipreading systems face some challenges that make them less reliable compared to the acoustic counterpart as listed below.

- **Visual speech provides less information than the acoustic signal.**

  Speechreading may be a natural way of silent speech communication between humans but, compared to the audio signal, the video signal is impoverished. Various works including [99, 102, 10, 134] estimate that only about 30% of speech production information is visible on the lips as the vocal cords, nasal cavity, and oral cavity are mostly hidden. This leads to *homopheneous* words which look the same on the lips but sound different (words such as /b/ bat and /m/ mat are often perceived to be identical by lip readers). This becomes the main limitation of computer lipreading. Visual speech captures only visible speech articulators such as an appearance of lip shape, teeth and tongue. It misses information from the vocal cords which is a main source of speech, and many more articulators are not visible. A study by Newman et al. [97] noted that electromagnetic articulography (EMA) signals illustrate that removing signals generated at the back of the mouth such as the velum (soft palate), tongue dorsum, and tongue blade decreases speech recognition performance significantly. In fact, the missing information leads to confusion in humans. For example, perceiving a similar group of sound such as /b/ bat and /m/ mat with no sound can look the same. Therefore, to decode lipreading the context of speech including a conversation topic, and the background knowledge for the topic is far more important than it is for acoustic speech.

- **Lipreading has impoverished data sets.** A modern machine learning technique requires a massive dataset to train an accurate model. Furthermore, a study by Howell [60] shows evidence that visual speech modelling needs more training data than acoustic modelling to achieve optimum performance in the same task. Unfortunately, the size of most available audiovisual speech corpora is small and may be unsuitable to train a sophisticated machine learning.

- **Lipreading performs poorly on LVCSR.** In the early day of lipreading as in 2004 [63], computer lipreading for LVCSR was believed to produce meaningless results because it is a highly confusable task. Recent advances in computer vision, speech processing and machine learning ought to feed-in to better lipreading systems. Of these advances, deep-learning is the most prominent. In a small vocabulary task, for example, lipreading systems via convolutional neural network (CNN) features and attention-based encoder-decoders achieved word accuracies of 97% [26] on the GRID dataset [30]. An end-to-end lipreading system using Long Short-Term Memory (LSTM) networks on the OuluVS2 [7] dataset achieved 84.5% phrase accuracy [111]. However, in larger vocabulary tasks, the performance of lipreading is much lower even if a complex deep learning approach has been employed. In the MV-LRS task [28], the word accuracy of lipreading is reported as 43.6% in frontal view and 37.2% in profile view using sequence-to-sequence LSTMs. In the LRS task [26], a lipreading system achieved 49.8% word accuracy using a system called Watch Attend and Spell (WAS) which involves CNNs and multiple LSTMs.

- **Lipreading is sensitive to speaker variation and identity** which brings difficulties in multi-speaker, speaker-independent scenarios. Cox et al. [32] reveal the great challenge in lipreading unseen speakers in the isolated alphabet recognition task on the AVletters 2 corpus. They illustrate that visual speech features are highly sensitive to the identity of the speaker. In a situation where some of the data from a test speaker were seen (speaker dependent and multi-speaker), the word accuracy of lipreading reaches more than 70%. Conversely, the mean of word accuracy in the unseen scenario is only 21%. Several techniques in ASR have been explored to improve lipreading visual speech features. In [32], a speaker normalisation technique based on maximum likelihood linear regression (MLLR) has been applied to enhance lipreading accuracy. Recently, Almajai et al. [4] employ several Gaussian Mixture Model hidden Markov model (GMM-HMM) training and feature transformation steps to pre-possess visual features including linear discriminant analysis (LDA) and feature space MLLR. They use DNN-HMMs as a visual speech model. The results show that these feature normalisation and pre-processing steps increase the performance of lipreading unseen speakers significantly.

- **Lipreading requires that the front-end system handle visual variations.** Compared to the acoustic signal, videos of lips have a number of sources of variability: environment, angle, lighting, distance, etc.

1.1 Motivation and aims

This study aims to tackle the problem of large-vocabulary continuous speech recognition in lipreading, which is not yet a solved problem; partly because of the lack of a visual speech corpus for this task and partly because machine learning methods for visual speech modelling are usually based on HMM. In 2015, Howell [60] proposed incorporating a phoneme confusion model to enhance lipreading output and achieved 76.14% word accuracy in single speaker 1000-vocabulary continuous speech task on the RM-3000 dataset. Although acoustic ASR achieved over 90% word accuracy in the resource management (RM) which is the same task since 1991, this level of achievement might make lipreading a practical reality in the future.

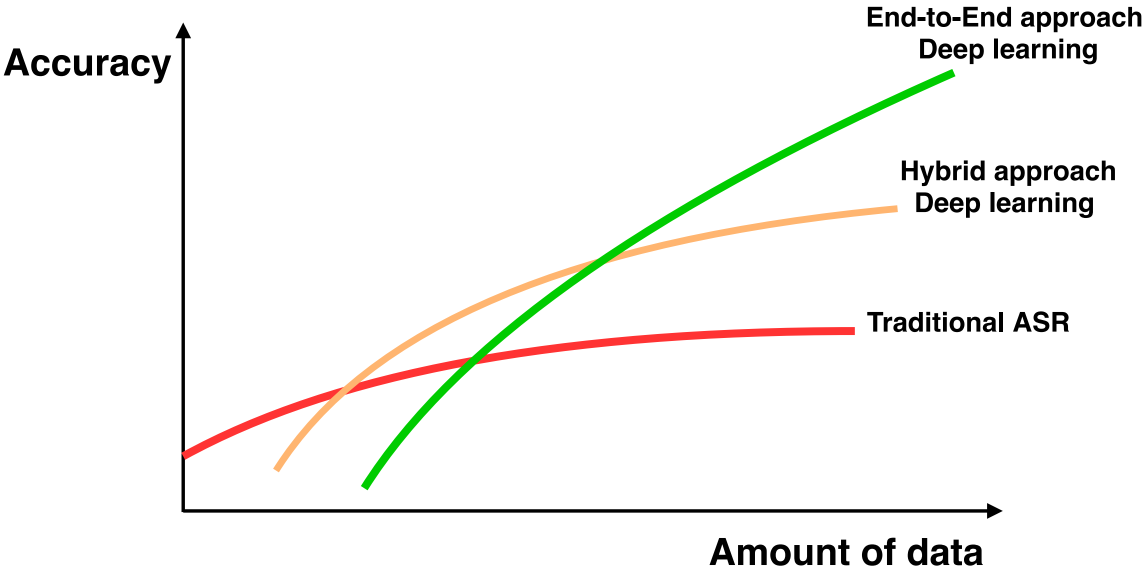


Fig. 1.1 Performance of automatic speech recognition system (ASR) comparing between deep learning approachs and conventional approach.

Another important motivation is the development of ASR systems. Significant progress in ASR has resulted from the introduction of deep learning in the form of DNNs. Figure 1.1 sketdus that the word accuracy of ASR tends to increase via a deep learning approach more than the traditional method. This has improved the accuracy achieved by conventional HMM-based ASR systems in both clean and noisy conditions [131]. The speech parameters learned in the deeper layers of the network are believed to be less influenced by noise, and the trained network can be used instead of the traditional GMM to estimate the likelihoods in the HMM. This notion can be extended to the visual modality of speech by incorporating visual information into deep learning.

In this work, we combine the DNN with HMMs to the, so called, hybrid DNN-HMM configuration which we train using a variety of sequence discriminative training methods. This is then followed by a weighted finite-state transducer. We consider the same feature processing methods as in [32, 4] such as LDA, and fMLLR. These methods are useful and general in ASR. This study focuses on using a type of DNNs model in lipreading. More specifically we use deep belief network (DBN) proposed by Mohamed et al. [89] to model HMM state probabilities instead of the conventional GMM. Regarding the use of deep learning with visual information for noise robust speech recognition, [62] constructed a noise robust audiovisual speech recogniser using deep belief network (DBNs) to recognise connected digits. Their result showed that using mid-level feature fusion DBNs reduced the word error rate by 21% relative to the baseline multi-stream GMM-HMM in noisy conditions (average 7dB). Deep learning techniques in the form of Deep Autoencoders (DAE) [98] and Deep Boltzmann Machines (DBMs) [135] have also been used in cross-modality unsupervised feature learning for improving the classification performance on the AVLetters and CUAVE databases.

1.2 Research question

In this thesis, we ask 'Can we simply employ the DNN-HMM hybrid approach, which is used successfully in ASR, to improve computer lipreading in the LVCSR task?'

There are several relevant points why we are interested in this topic as listed below.

1. Many authors believe that lipreading machine learning needs to be bespoke to lipreading.
2. Any bespoke system will have considerable more difficulty in using the existing language models (developed for acoustic recognition).
3. There is therefore a very strong practical and theoretic justification for trying carry-over techniques from the acoustic domain to the visual.

1.3 Statement of originality

Unless otherwise noted or referenced in the text, the work described in this thesis is that of the author. Novel contributions of this thesis can be summarised as follows:

- A graph summarising the developement in computer lipreading (Chapter 2).
- The use of existing techniques in ASR – the DNN-HMM approach– to make viable computer lipreading in an LVCSR task.
- Evidence that AVSR is more robust than audio-only ASR in any SNR in matched-conditions by simply using DNN-HMMs and feature fusion (Chapter 5).
- The achievement of best results of lipreading system with around 60% word accuracy in a 6000-vocabulary TCD-TIMIT corpus (Chapter 8) and 85% word accuracy in a 1000-vocabulary RM-3000 corpus (Chapter 5).
- A comparison of unit accuracy and word accuracy particularlly in LVCSR task between a phoneme recogniser and a viseme recogniser (Chapter 6.5.1).
- A full benchmarking of lipreading on four feature types: Discrete Cosine Transform (DCT), Eigenlips, Dual-tree complex wavelet transform (DTCWT), and DAE (Chapter 7).
- Evidence to explain why DNN improves lipreading (Chapter 7.2.2).
- An insight into visual features and the importance of feature transformations (Chapter 7.3.1).
- An analysis of lipreading results regarding speaker dependency (Chapter 8).
- A new evidence regarding the complexity of visual silence that caused errors at the start and end of sentences (Chapter 8).

1.4 Contributing publications

The following publications have been produced by the work in this thesis:

- Thangthai, K., Harvey, R. W., Cox, S. J., Theobald, B. J., **Improving lipreading performance for robust audiovisual speech recognition using DNNs**. – In *Proceedings of The 1st Joint Conference on Facial Analysis, Animation and Auditory-Visual Speech Processing (FAAVSP)*, Vienna, Austria, pp 127–131, 2015.

- Thangthai, K., Harvey, R., **Improving Computer Lipreading via DNN Sequence Discriminative Training Techniques**. – In *Proceedings of the Annual Conference of the International Speech Communication Association INTERSPEECH 2017*, Stockholm, Sweden, pp 3657–3661, 2017.

- Thangthai, K., Bear, H. L., Harvey, R., **Comparing phonemes and visemes with DNN-based lipreading**. – In *LRDLM Workshop on Lip-reading using Deep Learning Methods (at BMVC 2017)*, London, UK, 2017.

- Thangthai, K., Harvey, R., **Building large-vocabulary speaker-independent lipreading systems**. – In *Proceedings of the Annual Conference of the International Speech Communication Association INTERSPEECH 2018*, Hyderabad, India, pp 2648–2652, 2018.

Chapter 2

Introduction to computer lipreading

This section provides an overview of lipreading systems including background, recent developments, and the state-of-the-art.

2.1 Development of lipreading technology

Lipreading technology was first formulated in the 1980s. The first lipreading system was created by Petajan [110] in 1984; which was launched three decades later after the AT&T Bell Laboratories had built “Audre” the first speech recognition system in the 1950s. Brooke and Scott [19] summarised an early development of lipreading technology. In the early days, lipreading systems were targeted at simple tasks such as recognising isolated digits and letters. A typical system worked by capturing binary black and white images of lip regions in real time via special-purpose hardware. The captured images were then decoded using a template matching method to compare to the stored templates of different vocabularies. At that time, the main challenge was to make a system that was capable of processing sequences of facial images within a reasonable time [19], since there were very serious limitation in the hardware.

Visual speech processing via an Artificial Neural Network (ANN) approach had been explored since the late 1980s. An ANN is a computer model which is inspired by the human brain, that can learn a mapping function between input and output. In 1986, Peeling et al. [108] used the Multi-Layer Perceptron (MLP), a form of ANN, with three layers and two units per layer to classify vowels from 16 × 16 gray-scale lip image. In 1990, Brooke and Templeton [20] reported 91% classification accuracy identifying

11 vowels from 16 × 12 monochrome lip images via a six-unit MLP-based model. This technique reveals a possibility of representing visual cues from a multi-dimensional lip image using a few MLP parameters. However, MLPs were found to be sensitive to small changes in the image space, so other feature dimensionality reduction techniques such as Principal Components Analysis (PCA) [156] and DCT [2] were proposed in lipreading later on.

The statistical approach based on HMMs has been adopted in lipreading systems in the mid-1990s. HMMs are a mathematical model based-on a statistical approach using Markov chains and GMMs that have strong abilities to model time sequence data with a complex distribution such as speech input. The HMM-based approach for speech recognition has been very popular in the ASR and AVSR research areas, where a well-known tutorial of using HMMs in speech recognition was published by Rabiner [122] in 1989. Since then, HMMs have become a primary technology for speech recognition, audiovisual speech recognition and lipreading systems over the last two decades.

2.1.1 Lipreading tasks

Most of the current benchmarking audiovisual speech corpora such as CUAVE [107], AVLetters [85], and GRID [30], contain only 10 to 50 word vocabularies and are also constructed on a simple task such as digits and letters. Figure 2.1 shows some development of lipreading technology, which we have gathered from reports and publications spanning the last 20 years. The graph reveals the development of lipreading performance over time on several audiovisual speech corpora and tasks where difference colours represent a task, and each line represents the performance on each corpus. The tasks are isolated words and phrase recognition (shown in blue); connected digits and letters recognition (shown in green); restricted grammar recognition (shown in orange); small vocabulary continuous speech recognition task (shown in magenta); and large vocabulary continuous speech recognition (shown in red).

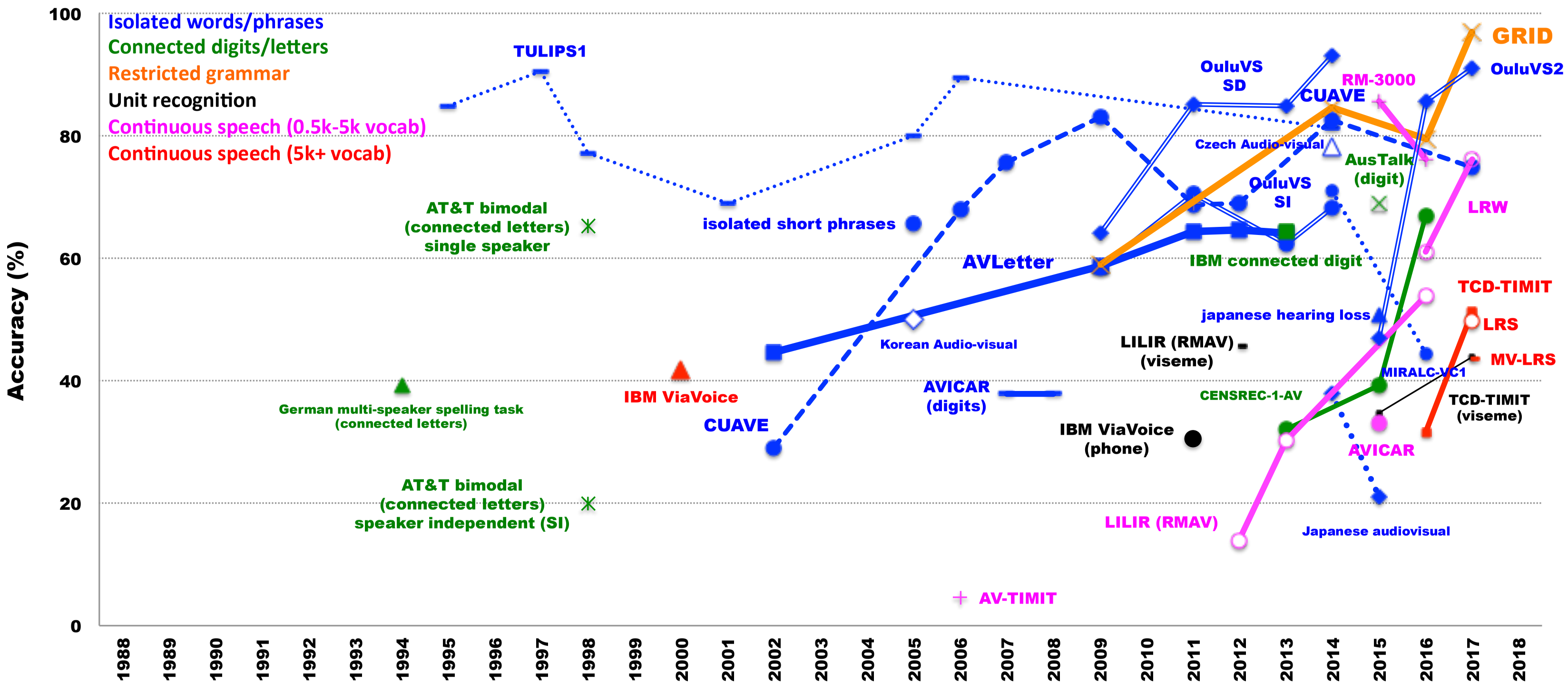


Fig. 2.1 History of computer lipreading showing the accuracy of various experimental systems versus time. Each database is shown as a different line and the colours refer to tasks where blue denotes isolated words and phase recognition; green denotes continuous digits and letters; orange denotes restricted-grammar; black denotes unit recognition; magenta denotes small vocabulary continous speech recognition; and red denotes large vocabulary continuous speech recognition.

According to Figure 2.1, lipreading performance has increased over time. In the beginning, the accuracy of computer lipreading continuously improves which we can see on three main corpora: CUAVE, AV-Letter, and TULIPS. Since 2010, the results of these three corpora have saturated while results of many other corpora started to develop. Here we can see a sharp improvement again after a new wave of deep learning technology after 2012. In this graph, the most challenging task is the large vocabulary continuous speech shown in the red lines. The large vocabulary defines the vocabulary size roughly between 5,000 to 60,000 words [66]. The word accuracy of this task rarely reaches 50% even with extensive training.

In lipreading, the simplest task is isolated word and phrase recognition which achieves good performance as can be seen from those blue lines. The isolated word task means that there is a pause between words. TULIPS1 [91] is an example of an isolated word task. There are four digits captured from 12 speakers. The state-of-the-art word accuracy of visual-only speech recognition is 90.60%, reported by Luettin and Thacker [81] in 1997. Luettin and Thacker [81] also reported that a professional lipreader gets 95.59%, and a normal hearing person gets 89.93%. This is a pattern repeated in future systems: automatic systems give performances inferior to trained human lipreaders but superior to untrained ones. More well-known datasets such as CUAVE [107], AVLetter [85], OuluVS [164], OuluVS2 [7], and a part of AVICAR [78] also fall into this simple task. Initial works report low performance, but later systems quickly reach more than 60% performance within a few years. AVICAR [78] is one of the more challenging audiovisual speech corpora. It was built to tackle challenges in both computer vision and acoustic conditions. The acoustic challenge is a recording in a car with various driving speeds, while the computer vision challenge combines various camera angles, moving speaker, multi-speakers in a video frame and so on. These conditions make the AVICAR performance drop below the results of other corpora in this category.

The second category is connected digits and letters. The performance is shown by the green lines. This task has the same vocabulary size compared to the isolated word task. The difference is that the speech signal of this task is more natural since there is no pause between words. Most of the corpora in this task are not publicly open. We can see that there is no continuous development over time and the performance still fluctuates.

The third category is continuous speech with restricted grammar. The primary dataset is the GRID corpus [30]. The script of GRID corpus can be viewed as a command and control speech style for example “SET WHITE WITH P TWO SOON”, "PLACE RED IN A ZERO NOW". This task has a 51-word vocabulary. The state-of-the-art lipreading result is 97% word accuracy which was reported recently by Chung et al. [26]. This is the highest word accuracy of any visual only speech recognition system albeit on a very restricted task.

The last category is continuous speech with larger vocabularies. We separate the results of this task into three levels: unit recognisers, medium vocabulary recognisers, and large vocabulary recognisers. A unit recogniser (black lines) evaluates the visual speech model performance rather than the transcription performance. They measure the success rate in terms of phoneme accuracy or viseme accuracy. This is based on an assumption that the improvement of unit accuracy may directly translate to the performance in word level.

Word accuracy of the medium vocabulary lipreading task is shown in magenta. Performance of this task varies depending on its complexity. For example, the word accuracy of a single speaker RM-3000 [60] is higher than 70%, but the performance is much lower in multi-speaker (as in RMAV [77] and LRW [27]) and multi-angle camera (as in AVICAR).

Small vocabulary tasks

Table 2.1 shows the state-of-the-art results for small vocabulary lipreading tasks. Also shown are the number of talkers and the number of utterances.

Table 2.1 Small vocabulary lipreading datasets and state-of-the-art performance

Corpus	ASR task	Speaking style	Talkers	Vocab size	Utt	Accuracy (%)
CUAVE [107]	Isolated digits	Read speech	36	10	7k	83.00 [104]
OuluVS [164]	phrases	Read speech	20	10	1k	70.60 [165]
OuluVS2 [7]	phrases	Read speech	52	20	3.6k	91.10 [28]
AVLetters [85]	Isolated letters	Read speech	10	26	0.7k	69.60 [109]
AVLetters2 [32]	Isolated letters	Read speech	5	26	0.9k	91.80 [109]
GRID [30]	Restricted grammar	Command and control	34	51	34k	97.00 [26]

Medium and large vocabulary tasks

Table 2.2 shows the best performances on larger vocabulary tasks, measured as word accuracy, for isolated (I) and continuous speech recognition (C) tasks.

The first system to report on a large vocabulary task, IBM ViaVoice [95], was devised in 2000 and reports a word accuracy of 48.92% on a 10,400 word vocabulary. Unfortunately, the first system was not a full lipreading system since it used the visual

Table 2.2 Medium-sized lipreading databases and state-of-the-art performance

Corpus	ASR task	Speaking style	Talkers	Vocab size	Utt	Word accuracy (%)
LRW [27]	C	Spontaneous speech	1,000+	500	500k	84.50 [26]
RM-3000 [60]	C	Read speech	1	1,000	3k	84.67 [149]
LiLiR (RMAV) [77]	C	Read speech	12	1,000	2.4k	∼53.00 [4]
AVICAR [78]	I and C	Read speech	100	1,356	59k	∼33.00 [15]
AV-TIMIT [51]	C	Read speech	223	1,793	4.6k	3.7 [50]
TCD-TIMIT [49]	C	Read speech	62	6,019	5.7k	51.29 [148]
IBM ViaVoice [95]	C	Read speech	290	10,400	18k	48.92 [95, 113]
MV-LRS [28]	C	Spontaneous speech	1,000+	14,960	74.5k	37.2 [28]
LRS [26]	C	Spontaneous speech	1,000+	17,428	118k	49.80 [26]

model to re-score a lattice produced from noisy audio. More recently, using data recorded from the BBC news, the LRW task, the best result was 84.5% accuracy but was achieved on a small vocabulary. For larger vocabularies the word accuracy drops as does, often, the number of talkers. For example in RM-3000 and LiLiR, a continuous lipreading task with a DNN-HMM hybrid architecture, for the single-speaker 1000-vocabulary, 3000-word-utterance database, RM-3000 accuracies of 76.14% [60, 61] and 85.67% [149] are reported. For the AVICAR data [15], which consists of isolated-words, connected-digit and continuous speech tasks, the word accuracies range between 24.53% and 33% on combined 4-camera using multi-stream HMM. Recently in 2017, [26] reported using an end-to-end deep learning system to obtain a 49.8% word accuracy on 4960 hours of BBC news audiovisual speech data (the LRS task). The data contain 118k utterances recorded from thousands of speakers with a vocabulary size of 17,428 words. Notably, on a a similar task, a professional lipreader achieved only 26.2% word accuracy[1].

2.2 Visual features

A feature extraction method aims to extract static and dynamic (or temporal features) of visual representations from a video. A static feature is a set of numbers extracted from a video frame. It involves techniques based on image transformations to compress image pixels of the lip region. A dynamic feature extracts a visual representation by considering lip movements from multiple frames.

Visual speech features can be grouped into shape-based (lip-contours) and appearance-based (pixels). Here we focus on the appearance-based derived from pixel values. Considering the mouth image as the region-of-interest (ROI), many image-based com-

[1] This disparity between human and automatic systems has also been reported in [77].

pression algorithms were proposed to deal with the high dimensional space. The purpose is to compress the raw pixels into a feature vector while retaining the most relevant speech feature from the visible articulators such as the lips, teeth, and the tongue tip. Feature compression is a key to extract informative representations from shigh-dimensional data. It becomes a common technique in speech recognition, since most machine learning models including the conventional HMM need a compact feature. Table 2.3 illustrates feature compression techniques that have been tried to extract a static feature. We group these techniques into four categories based on transform functions. The columns separate features into linear and non-linear transform functions. The rows separate features into fixed transform and learnt transform approaches.

Table 2.3 Transform functions used in lipreading.

	Linear	Non-linear
Fixed transforms	Cosine transform Wavelet transform Other transform	Sieve
Learnt transforms	Principal component analysis (PCA) Independent component analysis (ICA) Linear discriminant ananlysis (LDA)	Non-linear PCA Restricted Boltzmann machines (RBM) Deep autoencoder (DAE) Convolutional neural network (CNN)

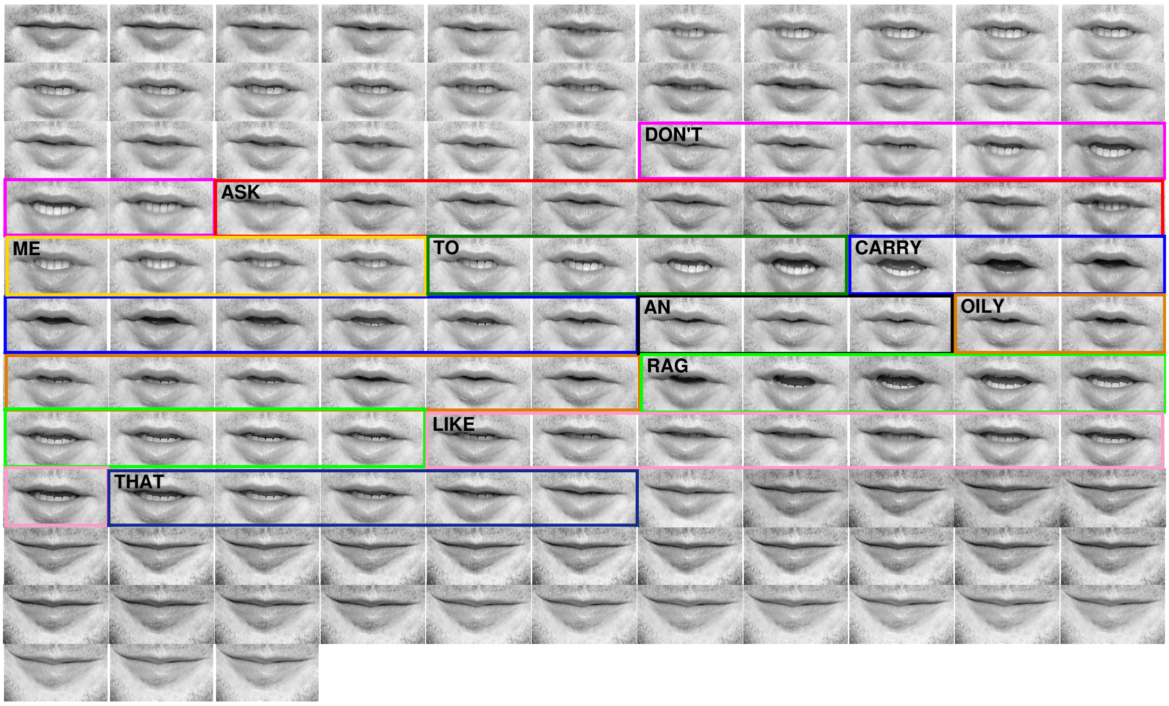


Fig. 2.2 TCD-TIMIT utterence, *"Don't ask me to carry an oily rag like that"*. There are ten words in this utterance. Colour-boxes indicate the boundary between words. No-box indicates a silence area. The boundary comes from acoustic force alignment provided in the corpus.

The temporal properties of a speech signal play an important role in identifying a linguistic unit [124]. More specifically different phone categories change at different temporal scales. To provide enough information to decode visual speech, a temporal feature computed from the changes between video frames is needed. Figure 2.2 is an example of a sequence of lip ROIs for the sentence *"Don't ask me to carry an oily rag like that"*. Figure 2.2 illustrates how tricky it is to define a speech class by providing only a single image since its mouth shape is similar to other images. The temporal transition of video frames between one another is a very useful source to predict a word. Therefore it is essential to provide a static feature that preserves detail such as the lip shape, the visibility of teeth and tongue as much as possible, along with the temporal coherence information which helps to identify a speech class.

Fixed transforms

We separate the static transform features into two types: linear and non-linear transforms.

Here are examples of linear visual features transforms: DCT as in [96, 53], discrete wavelet transform (DWT) as in [121], and DTCWT as in [38]. The DCT feature is the most common method used to extract visual features (as in [96, 53]). Neti et al. [96] used the score from the visual speech model to re-score noisy audio lattices. They found that DCT-based features achieved a better result than the active appearance model based features (AAM) discussed later. Seymour et al. [133] compared lipreading performance on corrupted videos among four features: DCT, fast discrete curvelet transforms (FDCT), PCA, and LDA. The results showed that DCT provided the best result in blurring conditions but poor results on the jitter condition. An alternative approach, the wavelet transform, does a multi-resolution analysis at different scale and resolution. All of these transform techniques help reduce image dimension using fewer selected parameters as a feature.

The only example of visual feature based on a static non-linear transform is the sieve feature proposed by Matthews et al. [84] (also used in [32]). The sieve feature contains extra information such as scale, amplitude and position. The scale parameter is robust to intensity and translation, so they extracted the scale histogram sieve feature to build lipreading system. They reported 50% accuracy visual for only speech recognition on the AVLetter task using 30 coefficients of the scale histogram.

Learnt transforms

Feature transformation via a data-driven approach can extract both static and dynamic features. There are two broad types of data-driven approach: unsupervised (data without labels) and supervised (data with labels).

In the unsupervised method, PCA is the most well known dimensionality reduction technique in lipreading. PCA projects the data in such a way that the largest variation of the data has been captured. PCA can be applied directly to lip ROIs to yield Eigenlips features [17]. Moreover, it can also be used to reduce the feature dimension of other features such as AAM [85] and DCT [24, 59].

Another well-known method is linear LDA. LDA is a supervised method which reduces feature dimension to preserve the class separation ability rather than the variation of the data. Thus, the speech class labels are necessary to determine the LDA transform. Potamianos and Graf [112] apply the LDA method on a stack of lip ROIs using the labels generated from an acoustic alignment. LDA obtained the highest word accuracy over PCA and DWT features. LDA has become a common method to extract compact features that contain dynamic information of speech in acoustic, audiovisual and visual speech. We also see a variant version of LDA as in the hierarchical linear discriminant analysis (HiLDA) feature [113] in the audiovisual domain.

The last group is visual feature extraction via a non-linear data-driven approach. Most of this group is based on deep learning. Visual features can also be derived via a deep learning in the tandem approach. The basic idea is to learn data via multiple layers of non-linear functions. Deep learning features can also be extracted via an unsupervised approach such as RBM, DAE or a supervised approach such as CNN. We explain this group in Section 2.3.2.

2.3 Deep learning in lipreading and AVSR

HMMs have been the dominant algorithm of speech recognition both in auditory speech and visual speech systems for more than 20 years. Recently, DNNs, a modified version of the ANN, have become increasingly common. A good starting point for the use of DNNs in speech recognition were the 2012 reports by Hinton et al. [55] and Mohamed et al. [89]. They proposed a DBN to solve the vanishing gradient problem that is found when training a very deep model. Since then, the deep learning approach has been increasingly replacing the HMM model.

A deep learning approach is a type of machine learning technique that learns multiple layers of data through time and/or space. The motivation for deep learning relates to the human brain where several nodes and layers mimic the work of neurons in the human brain. Deep learning has many different architectures and various learning types. Examples of deep learning architectures are feed-forward neural networks handling complexity in a high dimensional space, recurrent neural networks (RNN) handling sequence data, and CNN handling information of local connection in sub-regions.

This section demonstrates the four main approaches that have been used in audiovisual speech and lipreading systems, which are the classification approach, tandem approach, hybrid approach, and end-to-end approach.

2.3.1 Classification approach

Ngiam et al. [98] applied the DAE to learn the cross-modality and the shared representation of audio and visual speech features and then evaluated the performance of isolated word recognition tasks on CUAVE and AVLetters audiovisual speech corpora. They found that the visual-only DAE provided better representation and outperformed the other feature representation methods on both tasks. Moreover, the shared representation of audio and visual streams that were learned by using RBMs in an unsupervised fashion, significantly outperformed the conventional shared representation learning technique[2] even when one stream was missing while testing.

Srivastava and Salakhutdinov [135] investigated the DBM method for joint representation learning in multimodal aspects. This method involved the layer-wise pre-training procedure (to train one layer at a time), which effectively predefines the network weight by training on unlabeled data, and the discriminative fine-tuning procedure using supervised back-propagation. Evaluating in the isolated word recognition task, their method outperformed the classification performance proposed by [98] when compared to the best result of the bimodal DAE.

Mroueh et al. [92] proposed bilinear bimodal DNNs for audio-visual phoneme recognition. The proposed DNN models are constructed by fusing the softmax bilinear layer of DNNs trained on each modality individually, then the weights in the entire networks are optimised by using factored bilinear sharing back-propagation. The experiments have been done on the IBM AV-AVSR large vocabulary speech dataset, which contains about 40 hours of audio-video speech data and the vocabulary size is 10400 words. Compared to the baseline audio-only DNNs, the proposed method has shown significant improvement by 7.22% absolute reduction in phoneme error rate (PER) and the bilinear bimodal DNN shown to be a better method for capturing the correlation between the audio and visual streams.

2.3.2 Tandem approach

The tandem approach aims at extracting more informative representations from signals by using deep learning techniques. An extracted representation is then used as a feature in conventional GMM-HMM training. The benefit of using these as features is to gain advantage from several advanced techniques in GMM-HMM training especially discriminative model training methods such as minimum phone error (MPE) and maximum mutual information (MMI). In this approach, several deep network structures can be adopted for extracting audio and visual features such as deep denoising autoencoder (DAE), the CNN, LSTM, DBMs.

Noda et al. [101] investigated the robustness of audiovisual speech recognition by using the CNN to extract visual features and the DAE to extract more robust audio

[2] Canonical Correlation Analysis (CCA) [48].

features from the bottleneck layer. The audio and visual features which were obtained from the deep networks were used to train a multi-stream HMM. They found that using a bottleneck audio feature could gain approximately 65% word accuracy in the unimodal setting compared to the raw Mel-frequency cepstral coefficient (MFCC) features and the huge gain can be more clearly seen at less than 10 dB SNR. Moreover, the proposed approach on a multi-stream HMM can result in further improvements for SNR conditions below 10 dB, provided the proper stream weights were carefully selected. Similarly, Takashima et al. [144] use convolutive bottleneck networks (CBN) to extract more robust audiovisual features to enhance speech recognition in noise.

Additionally, Ninomiya et al. [100] and Tamura et al. [145] propose a similar method of using deep bottleneck features from the individual modality DNNs to train multi-stream GMM-HMMs. Ninomiya et al. [100] have further focused on exploring the level of integrating both modality features, while Tamura et al. [145] mainly investigated constructing robust visual features. Indeed, their experiments were then set on the same task which is the connected-digits by using a Japanese audio-visual corpus. Their results show that multi-stream HMMs trained on the bottleneck features can achieve 81.1% word accuracy and the further gained about 8.77% absolute from the availability of more informative visual features.

Sui et al. [140] proposed DBMs to extract visual features in the form of bimodal infer learning. Instead of learning DBM features for visual-only, they incorporate the audio signal into the DBMs training process using a separate DBM network, and then combine the DBMs from both modalities in the final layer. The idea of this study is that the learnt DBMs can be used for feature extraction even though one of the speech modalities was missing while testing. Their visual DBM feature was used in GMM-HMM training in combination with DCT before applying LDA. They evaluate the lipreading system on the digit-string task from the Austalk [21] database. The proposed method achieved 69.1% accuracy of which 15% was gained from a standard DCT + LDA feature and 1.3% from the state-of-the-art of this corpus at the time.

Zimmermann et al. [166] proposed the combination of PCA and LSTMs to extract visual features to train conventional GMM-HMMs for visual speech modelling. They evaluated the proposed techniques on phrase recognition task using OuluVS2 dataset. The results show that the proposed features achieved 79% sentence correctness on the cross-validation set, which is a 5% increase from the OuluVS2 baseline, and 73% sentence correctness on the test set.

2.3.3 Hybrid approach

The hybrid approaches mainly use a deep network to predict the state instead of GMM then the HMM was applied on top of the network to estimate the state transition probabilities.

Huang and Kingsbury [62] demonstrated the effectiveness of using a DBN to find a robust audiovisual representation, by reporting on continuous spoken digit recognition, which was compared against a baseline multi-stream GMM-HMM system. They also investigated the effectiveness of using DBN in AVSR by comparing between training the hybrid DBN-HMM recogniser and training the GMM-HMM on top of the DBN bottleneck feature. The best result showed that these methods can reduce word error rate by as much as 21% relative over a baseline multi-stream audio-visual GMM-HMM system.

2.3.4 End-to-end approach

Unlike a conventional speech recogniser that needs a feature extraction process, a speech model, a language model, and a lexicon model, the end-to-end approach learns a direct mapping between the input speech and the character sequence. There are two types of deep architectures to handle such complex mappings, (1) attention-based encoder-decoder methods, (2) Connectionist Temporal Classification (CTC). Attention-based methods use an attention mechanism to generate the alignment between an input sequence and output characters. The model then trains on encoder-decoder structure via LSTM networks. CTC uses the Markov assumption and dynamic programming to avoid the requirement of a predefined time alignment.

Chung et al. [26] proposed the ‘Watch, Listen, Attend and Spell’ (WLAS) network that uses attention-based encoder-decoder architectures in lipreading and the AVSR system. The proposed system uses a CNN as visual features and MFCC as acoustic features. In the visual-only system, they achieved 50.2% word error rate (WER) on the LRS dataset.

2.4 Audio visual database

We aim to tackle the challenge of building a lipreading system on the large vocabulary continuous speech recognition task which is more realistic. We, therefore, select the largest vocabulary size that was available at the time we began experiments to evaluate our lipreading system. So, we use three audiovisual speech corpora: RM-3000 [60], RMAV [77] and TCD-TIMIT [49]. RM-3000 and RMAV were derived from the acoustic RM database and TCD-TIMIT was derived from the TIMIT database. They have high potential in terms of vocabulary size, continuous speech task and large number of speech utterances as shown in Table 2.4. Note that at the time we did experiments, larger AV datasets such as LRS and MV-LRS were not available.

Table 2.4 Statistics from all three datasets. Note that statistics of RM-3000 are provided in [60].

Statistic	RM-3000	RMAV	TCD-TIMIT
Total number of speakers	1	20	59
Total number of sentences	3,000	4,000	5,488
Total number of unique phonemes	45	45	38
Total number of phoneme tokens	105,561	139,951	213,115
Total number of unique words	979	984	5,958
Total number of word tokens	26,114	33,031	47,503
Average number of phonemes per sentence	35.19	34.98	38.83
Average number of words per sentence	8.70	8.25	8.65
Average number of phonemes per word	4.04	4.23	4.48

2.4.1 RM-3000 single-speaker audiovisual speech corpus

The Resource Management (RM)-3000 corpus [60] is a single-speaker continuous audiovisual speech database, which contains 3000 speech utterances spoken by a single male native English speaker. The data were captured in the frontal pose in clean conditions. The corpus contains 260 minutes of audio-visual speech data. The vocabulary size is about 1000 words, and the mean sentence length is 8.7 words equating to approximately 5 seconds. The AAM features were produced by an AAM built by hand-labelling 20-30 video frames, which was then fitted to the video sequences automatically.

2.4.2 RMAV multi-speaker audiovisual speech corpus

The Resource Management Audiovisual speech corpus (RMAV) [77] is a multi-speaker continuous audiovisual speech database. This corpus contains totally 4000 utterances of speech data from 20 speakers, ten male and ten female, i.e. 200 utterances for each

Fig. 2.3 Examples of the RM-3000 corpus.

speaker. The vocabulary size is 1000 words. The corpus was recorded in full-frontal by HD cameras, and several other views and the AAM features were also provided by using a speaker-independent AAM automatic tracker.

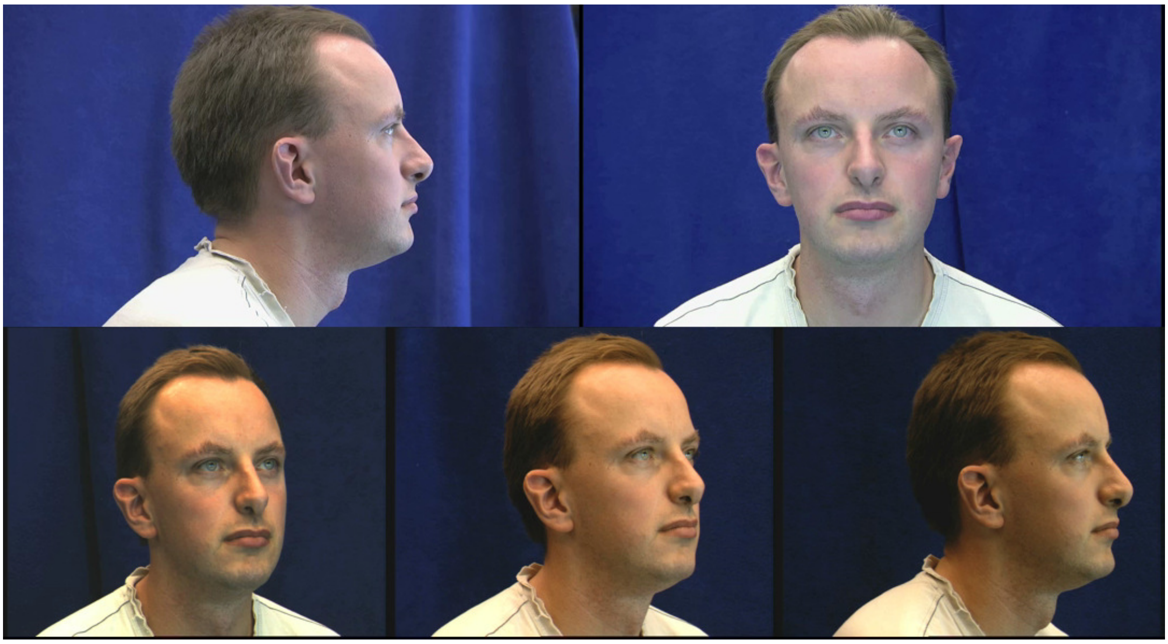


Fig. 2.4 Examples of the Resource Management Audiovisual speech corpus (RMAV).

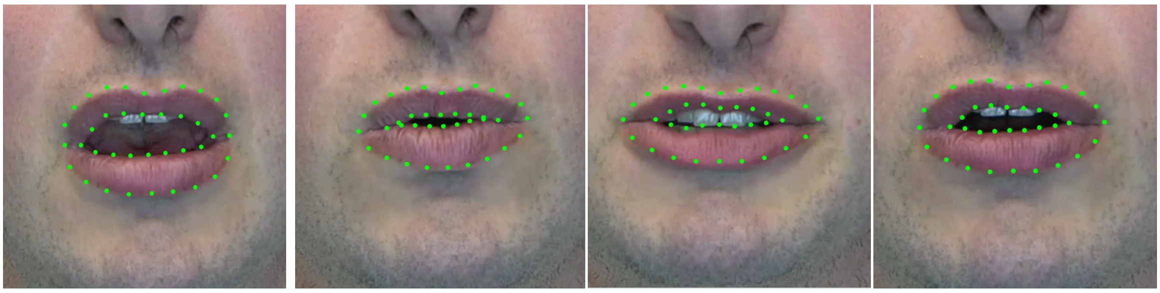
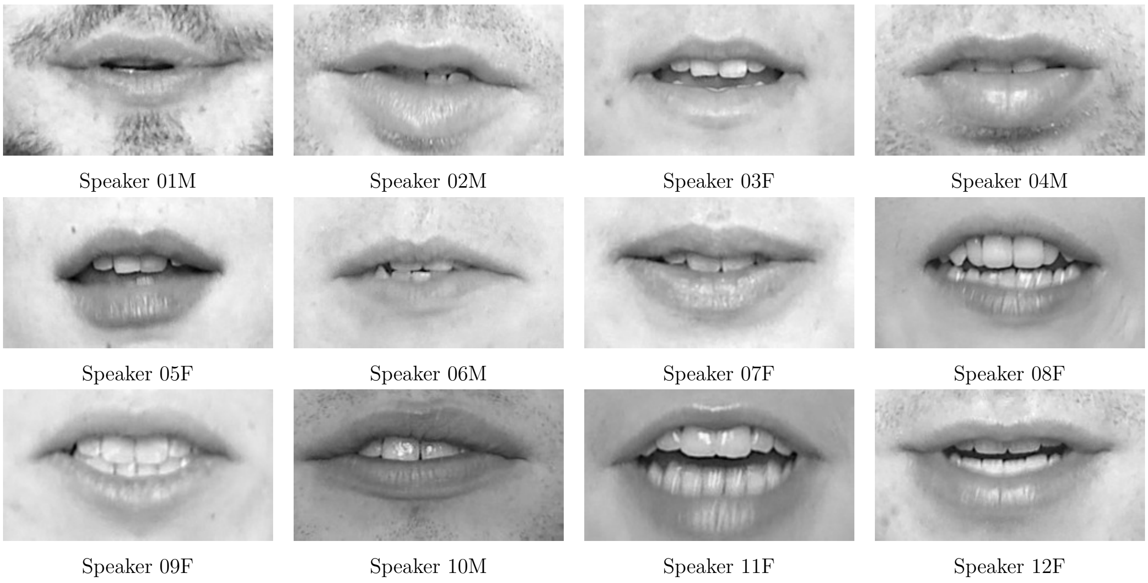


Fig. 2.5 Examples of the AAM automatic tracker on the RMAV corpus. The green dots are the landmarks.

2.4.3 TCD-TIMIT audiovisual speech corpus

TCD-TIMIT [49] is a publicly available audio-visual continuous speech corpus that has a 6019 word vocabulary recorded from 59 talkers (the volunteer set) and three professional lip speakers comprising over seven hours of speech data. The video is recorded in two views: frontal and 30° view captured in a studio environment with Sony PMW-EX3 cameras and a wireless clip-on microphone. We use only the frontal view. Each talker reads 98 sentences selected from TIMIT. The majority of talkers (56) have an Irish accent. The remaining three talkers are removed as prescribed in [49]. Thus the total number of utterances is 5488, captured from 56 speakers. Harte and Gillen [49] provide a preliminary report of the accuracy using 12 viseme classes: the best results were 34.54% and 34.77% viseme accuracy in the speaker-dependent (SD) and speaker-independent (SI) conditions respectively.



Speaker 01M Speaker 02M Speaker 03F Speaker 04M

Speaker 05F Speaker 06M Speaker 07F Speaker 08F

Speaker 09F Speaker 10M Speaker 11F Speaker 12F

Fig. 2.6 Examples of the TCD-TIMIT corpus.

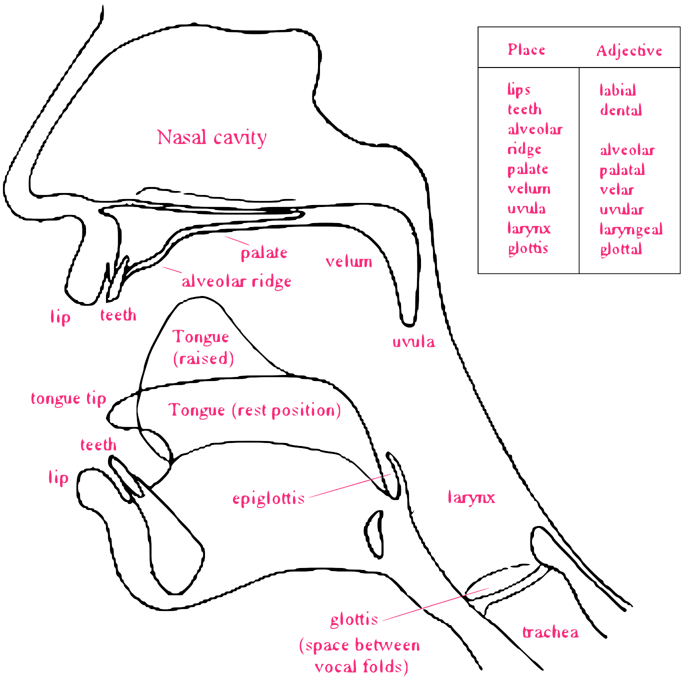
Chapter 3

Visual aspects of speech

This Chapter provides background knowledge of speech organs for speech production. We give a summary of human speechreading. We describe linguistic representations and the homophene problem.

3.1 Speech production

Here we provide the basic knowledge of speech production (summarising from Parsons [106]). Speech production involves three parts of the vocal organs: lungs and trachea, larynx, and vocal tract, as shown in Figure 3.1. These three parts have different functions to control the airstream. First, the lungs and trachea, known as a power source, mainly control the loudness of speech by controlling the delivery of compressed air. Since humans use lungs for the exchange between oxygen and carbon dioxide, we, therefore, have to manage breath during talking. Second, the larynx, known as a vibrator, controls the vocal cords. Third, the vocal tract involving the oral cavity, and the nasal cavity provides speech modulation.



Place	Adjective
lips	labial
teeth	dental
alveolar ridge	alveolar
palate	palatal
velum	velar
uvula	uvular
larynx	laryngeal
glottis	glottal

Fig. 3.1 Speech organs presented in [54]

Speech information provided by the vocal cords is the *excitation* that is generated by the different form of the glottis. These organs vibrate when the air stream generated in the lungs passes through. The vibration of the vocal cords is called *voiced* sound and speech with no vibration of vocal cords is called *unvoiced* or *voiceless* sound. Here are examples of sound pairs between voiced and voiceless sounds: *Zoo* (/z/) vs. *Sue* (/s/); *Down* (/d/) vs. *Town* (/t/). The vocal tract in the oral cavity and nasal cavity works as a modulation. This modulation is the primary factor to produce consonants and vowel sounds.

3.2 Visual speech unit

Visual speech units divide into two broad categories: phonemes and visemes. A phoneme is the smallest unit of speech that distinguishes one sound from another [8]. Therefore it has a strong relationship with an acoustic speech signal. In contrast, a viseme is the basic visual unit of speech that represents a gesture of the mouth, face and visible parts of the teeth and tongue, the visible articulators. Generally speaking, mouth gestures have less variation than sounds, and several phonemes may share the same gesture so a class of visemes may contain many different phonemes. Table 3.1 represents phonemes and their phonetic representation along with the viseme mapping.

Table 3.1 Phonetic alphabets and Neti et al. [95] viseme class. This table is adapted from [106]

IPA symbol	Phoneme	Viseme	Examples	IPA symbol	Phoneme	Viseme	Examples
i	iy	V4	h*ee*d	v	v	G	*v*erve
ɪ	ih	V4	h*i*d	θ	th	F	*th*ick
e	ey	V3	h*ay*ed	ð	dh	F	*th*ose
ɛ	eh	V3	h*ea*d	s	s	B	cease
æ	ae	V3	h*a*d	z	z	B	pi*zz*a*z*
ɑ	aa	V1	h*o*d	ʃ	sh	D	me*sh*
ɔ	ao	V1	h*aw*ed	ʒ	zh	D	mea*s*ure
o	ow	V2	h*oe*d	m	m	E	*mom*
u	uh	V2	h*oo*d	n	n	C	*noon*
ʊ	uw	V2	wh*o*'d	ŋ	ng	H	ri*ng*i*ng*
ɝ	er	V1	heard	l	l	A	*l*u*l*u
ə	ax	V4	*a*head	l	el	A	batt*le*
ʌ	ah	V1	b*u*d	n	en	C	butt*on*
ɑɪ	ay	V3	h*ide*	ɾ	dx	G	ba*tt*er
ɑʊ	aw	V1	h*ow*'d	ʔ	q		
ɔɪ	oy	V1	b*oy*	w	w	H	*wow*
h	hh	V1	*h*eat	j	y	A	*y*o*y*o
p	p	E	*pop*	r	r	A	*r*oar
b	b	E	*bob*	ʧ	ch	D	*ch*ur*ch*
t	t	C	*t*ug	ʤ	jh	D	*j*udge
d	d	C	*d*ug	ʍ	wh		*wh*ere
k	k	H	*k*ick				
g	g	H	*gig*				
f	f	G	*f*ife				

In Table 3.1, the first column indicates the international phonetic alphabet (IPA), which is widely used to represent the different speech sounds. The second column shows the computer symbol of a phoneme. The third column indicates the visual clue of a phoneme which has a many-to-one mapping relationship. There are many choices of visemes [12]. Here we use the Neti mapping [95] as shown in Table 3.2. This mapping groups multiple phonemes into 13 viseme clusters including silence. The phonemes in the same viseme class are hardly distinguishable on the lips. For example, the words like *d*ug, *t*ug and *p*o*p*, *b*o*b* look pretty much the same.

Table 3.2 Phoneme-to-viseme mapping by Neti [95].

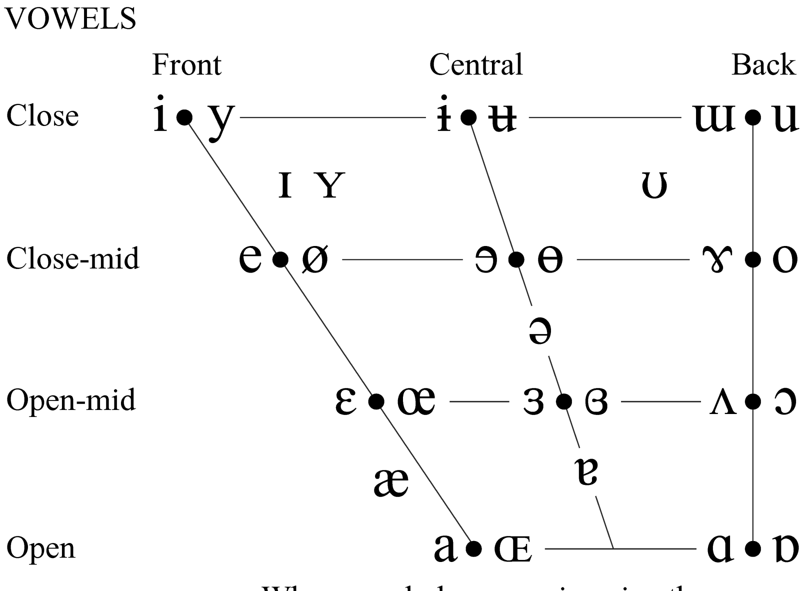
Viseme	TIMIT phonemes	Description
A	/l/ /el/ /r/ /y/	Alveolar-semivowels
B	/s/ /z/	Alveolar-fricatives
C	/t/ /d/ /n/ /en/	Alveolar
D	/sh/ /zh/ /ch/ /jh/	Palato-alveolar
E	/p/ /b/ /m/	Bilabial
F	/th/ /dh/	Dental
G	/f/ /v/	Labio-dental
H	/ng/ /g/ /k/ /w/	Velar
V1	/ao/ /ah/ /aa/ /er/ /oy/ /aw/ /hh/	Lip-rounding based vowels
V2	/uw/ /uh/ /ow/	"
V3	/ae/ /eh/ /ey/ /ay/	"
V4	/ih/ /iy/ /ax/	"
S	/sil/ /sp/	Silence

The Neti et al. [95] viseme clusters are inspired by linguistic as can be seen in the description of each viseme group. Therefore, this viseme cluster is related to speech organs as in characteristics of acoustic phonetics. In acoustic, each phonetic unit corresponds to consonant sounds or vowel sounds. Consonants and vowels have different characteristics, and consonants have some restriction of airflow but vowels do not.

Table 3.3 The categorisation of consonants.

	Manner	Voicing	Place						
			Bilabial	Labiodental	Interdental	Alveolar	Palatal	Velar	Glottal
Obstruent	Stop	Voiceless	p			t		k	ʔ
		Voiced	b			d		g	
	Fricative	Voiceless		f	θ	s	ʃ		h
		Voiced		v	ð	z	ʒ		
	Affricate	Voiceless					ʧ		
		Voiced					ʤ		
Sonarant	Nasal	Voiced	m			n		ŋ	
	Lateral	Voiced				l			
	Rhotic	Voiced					r		
	Glide	Voiced	w				j	(w)	

In articulatory phonetics [106], consonants can be described via three criteria: voicing, place of articulation, and manner of articulation as shown in Table 3.3. The voicing refers to the vibration of the vocal cords: voiced stand for vibration, and voiceless stand for no vibration. However, both cases are visually unobserved. The place of articulation defines the location of the articulator where the airflow is restricted. Here the positions range from the front to the back of the mouth. The phoneme classes that appear in the front are easier to observe than those that occur at the back. For example, phonemes /p/,/b/ in the bilabial group (happened between the lips) are easier to see than phonemes /k/,/g/ in the velar group occurring at the back of the tongue. The manner of articulation determines the level of airflow constriction. Manners of articulation in the top of the table are called obstruent, have full airflow constriction as in a stop consonant (/b/, /p/), or partial constriction as in a fricative (/f/, /s/) and an affricate (/ch/). Manners of articulation in the bottom called sonorant, have less constriction. In the nasal group, for example, most of the air passes through the nasal cavity. Since this group has low airflow constriction while still functions as a consonant, some of them are known as semivowels such as /w/, /j/, /l/, /r/.



Where symbols appear in pairs, the one to the right represents a rounded vowel.

Fig. 3.2 IPA vowels chart CC-BY-SA-3.0.

There are three criteria to describe vowel sounds: height, backness, roundedness. The height and the backness refer to the position of the tongue, and the roundedness refers to the lip shape. The tongue can stay in three positions to produce a vowel sound: front position, central position, and back position. There are four levels of height: close (high), close-mid (middle-high), open-mid (middle low) and open (low) as shown in Figure 3.2. The changes of the tongue are difficult to see. The roundedness of a vowel is the only information that is possible to observe from the lips. Each pair of vowels the rounded vowel presents on the right and the unrounded vowel presents on the left. For example, /i/ is an unrounded vowel. Thus the lip shape of the vowel /i/ is wider than the vowel /u/ which is a rounded vowel.

3.3 Speechreading

Speechreading in a hearing-impaired society is a very rich skill. It is composed of multiple components that are used together to understand speech. Kaplan et al. [68] explain that these components could be separated into two main groups: the analytic component perceived by eyes and the synthetic component interpreted by the mind. The analytic component involves physical body movements, such as the speaker's face, gesture and body language, facial expression, and other clues such as situation and linguistic factors. The synthetic component combines all those clues to "fill in the blanks" before interpreting a message. A speechreader uses these redundancies in information to avoid misunderstandings since information on the lips alone is limited. So, they use a situation to scope the topic and word choices and also use the linguistic knowledge together with other physical movements as a hint to deal with the ambiguities.

Speechreading has limitations in which it is nearly impossible to understand every speaker in any situation in the same level. Kaplan et al. [68] summarise these limitations into four groups of problems: 1) problems due to the talker, 2) problems due to the environment, 3) problems due to the speechreader, and 4) problems due to the speech signal.

The first problems are related to talkers where some people are easier to lipread than other people. Speechreading tends to be more difficult with an unfamiliar person, a person who barely moves their lips, or persons with facial hair. Other factors are related more on speaking styles. A talker who speaks more loudly and a bit slow is easy to understand. In fact, a low speaking rate relates directly to a clearer lip movement and voice projection.

The second problem is related to environmental factors such as distance from a talker, lighting conditions, and other sources of distractions. A speechreader needs a clear vision to see the talker face, which requires that it is not too close or too far, and not too dark or too shaded. They also need strong attention and concentration that could be distracted by irrelevant movements, and busy background both visually and acoustically.

The third problem is related to the speechreader themselves. Speechreading will be more successful if a speechreader has these characteristics: visual acuity, visual attention, familiarity with languages and topics, and the right attitude. Moreover, a good speechreader uses context and also has the flexibility to think about the possibility of a sentence that makes sense. These refer to the level of their synthetic skill.

The last problem is related to the limit of speech signal itself where speech information is partly visible on the lip. According to the study of Woodward and Barber [157], about 60% of speech sounds are invisible which means that almost 40% of the sounds are possible to see. However, Jeffers and Barley [64] found even less. They reported that visual speech presents only about 25% of information available in the sounds. This missing information leads to the homophene words problem: words that differ but appear to be identical on the lips.

3.4 Homophenes

A major reason for the difficulty in speechreading is the invisibility of many sounds. Since the vocal cord information is completely missing one cannot observe which sounds are voiced or voiceless. (Many consonants and vowels are invisible when they originate from inside the mouth.)

Table 3.4 Example of phoneme and Neti viseme [95] dictionary with its corresponding IPA symbols.

Word Entry	IPA Symbol	Phoneme Dictionary	Viseme Dictionary
TALK	t ɔ k	t ao k	C V1 H
TONGUE	t ʌ ŋ	t ah ng	C V1 H
DOG	d ɔ g	d ao g	C V1 H
DUG	d ʌ g	d ah g	C V1 H
CARE	k e r	k eh r	H V3 A
WELL	w e l	w eh l	H V3 A
WHERE	w e r	w eh r	H V3 A
WEAR	w e r	w eh r	H V3 A
WHILE	w ɑɪ l	w ay l	H V3 A

Homophenes are words that have a similar lip shape or movement. Table 3.4 reveals examples of homophenes where the first column shows words, the second and third columns are its phoneme sequence, the fourth column indicates the viseme sequence. It can be seen that talk, tongue, dog and dug have a different sequence of phonemes, but share the same viseme sequence. These make speechreading much more confusing especially as the number of homophenes increases.

Chapter 4

Machine learning techniques

This section provides background knowledge ranging from conventional techniques to recent advanced techniques employed in lipreading.

4.1 Overview of lipreading architecture

A computer lipreading system converts video frames of lip ROIs into text. This can be done with a similar approach to that of an automatic speech recogniser, but the model is built using visual speech observations instead of acoustic speech observations.

To decode any speech sequence $\mathbf{X}$, the word sequence $\hat{\mathbf{W}}$ can be defined by

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} P(\mathbf{W}|\mathbf{X}), \tag{4.1}$$

or, after applying Bayes' Rule:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} P(\mathbf{X}|\mathbf{W})P(\mathbf{W}), \tag{4.2}$$

where $P(\mathbf{X}|\mathbf{W})$ is the visual speech model likelihood and $P(\mathbf{W})$ refers to the language model likelihood. The word sequence that has the best likelihood score from both the speech model and the language model is the output of the system. Figure 4.1 illustrates the system architecture used in this work. We use a weighted finite-state transducer (WFST) framework to decode visual speech observations. We compute the visual speech likelihood via a DNN-HMM model. The language model probability is generated from word N-grams. A lexicon dictionary represents words with corresponding phonetic pronunciations. The WFST decoder has two-passes: the first-pass generates a word lattice; the second-pass re-scores the lattice and returns the 1-best word transcription.

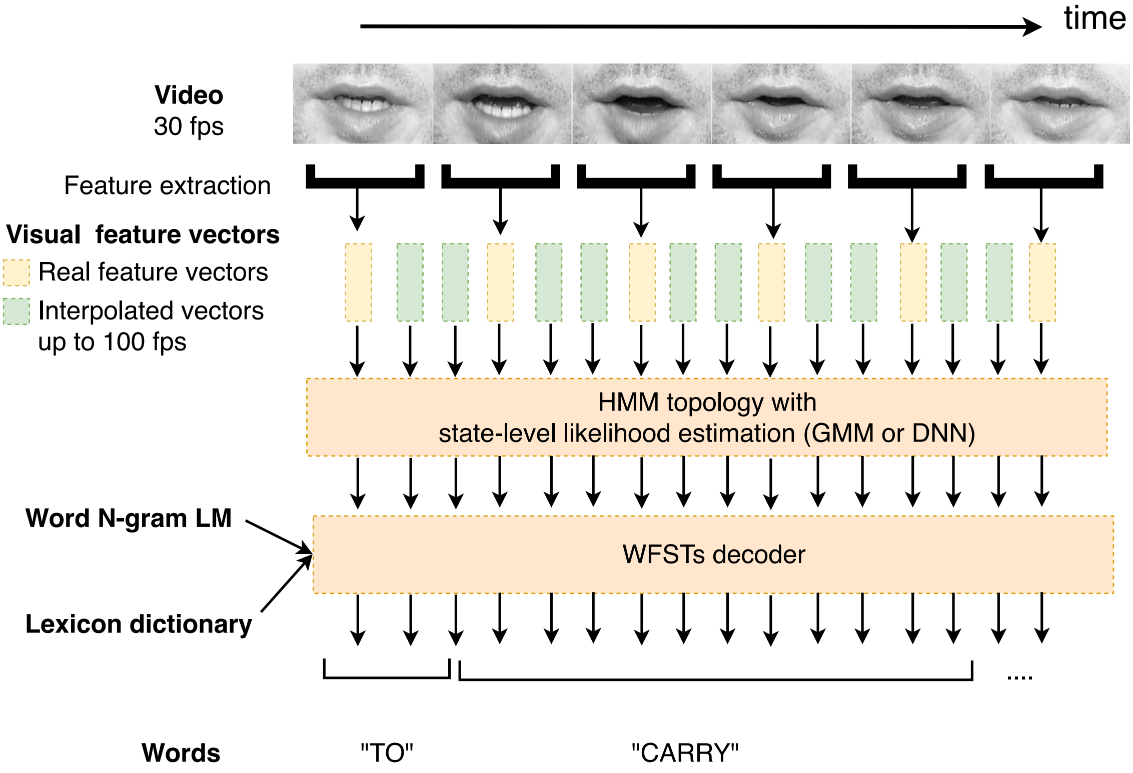


Fig. 4.1 Overview of our lipreading system.

Alternatively, these modules can easily be replaced by a deep learning architecture to build an end-to-end system. However, to achieve a good performance, it might require a few thousand hours of speech data from various speakers to learn a speech representation and a language representation. In practice, it seems a massive size of available training data is essential for building an end-to-end speech recogniser. For example, Deep Speech II [5] is trained on 12k hours of labelled speech data (around 8 million utterances) to achieve performance comparable to a human. A conventional architecture, such as a hybrid DNN-HMM technique works well in a small dataset. For example, the results in Graves and Jaitly [47] illustrate that a DNN-HMM system with a bigram and a trigram language model, as a baseline system, outperforms their proposed end-to-end methods in both 14-hour and 81-hour training sets. The benefits of a deep end-to-end system in a small training set was that their proposed method can be still functional even without a dictionary and a language model. A disadvantage of the end-to-end system is that it ignores prior knowledge of linguistics.

Since most lipreading corpora are relatively small, we can still get benefits from using a separate speech model, N-gram language model and dictionary. Therefore, we build our lipreading systems via a conventional architecture with hybrid DNN-HMM models. Our systems have a similar architecture to the baseline system in Graves and Jaitly [47]. The rest of this Chapter explains each component of these lipreading systems in more detail.

4.2 Visual speech model

This section presents background knowledge about algorithms to model visual speech observations. We describe techniques to train a conventional approach HMM. We also provide the basic idea to train a DNN. Information on the hybrid DNN-HMM model and the training methods are available at the end of the section.

4.2.1 Hidden Markov Model (HMM)

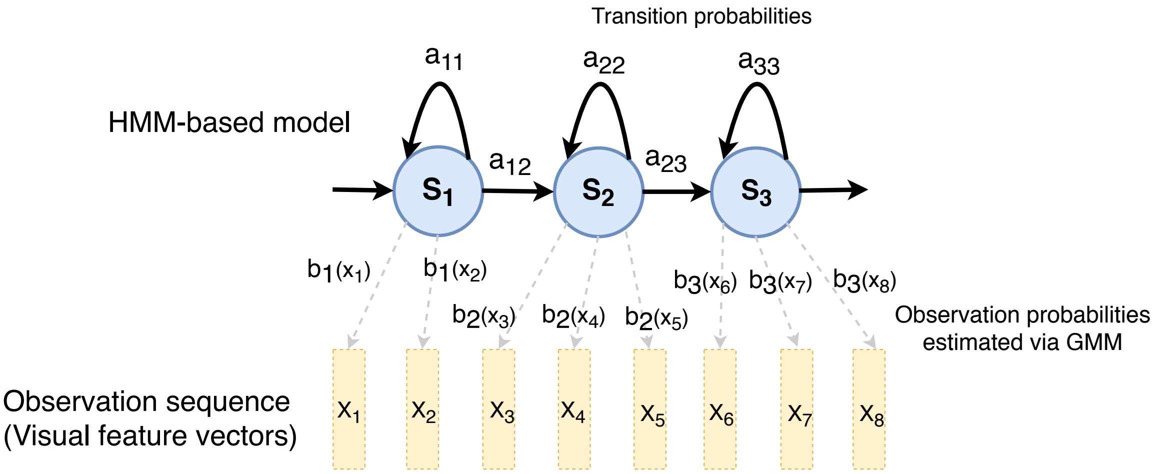


Fig. 4.2 The HMM-based model adapted from [43].

HMMs have been used in ASR/AVSR and lipreading as the state-of-the-art for speech modelling for the last several decades. Figure 4.2 shows the structure of the HMM-based model of a phoneme. HMM-based speech modelling uses a probabilistic model that is composed of two main probabilities: the transition probability and the observation probability which is derived from a probability density function. The transition probability is present to capture the sequence information from speech via a Markov process. The probability density function (PDF) of speech feature vectors is usually modelled by a GMM that is parameterised by the mean and the variance of each component (covariance matrices are usually assumed to be diagonal also the mixture weights).

Gaussian Mixture Model (GMM)

A GMM is a generative model that can represent the PDF of an arbitrary random variable (RV). A GMM is a linear combination of a number of multivariate Gaussian models.[1] It can be shown that a GMM can model the PDF of an arbitrary RV to any required degree of precision by adding more components to the GMM. A univariate Gaussian model can be used to model the distribution of data points where the mean (μ) specifies the center of the distribution, and the variance (σ^2) determines the spread of the distribution. Figure 4.3 shows a univariate example of how a GMM can model an arbitrary RV. The red trace represents the PDF of an arbitrary RV and the blue traces are the PDFs of three normal distributions with different means and variances that, when added together in the correct proportions, model the red trace to a high degree of accuracy. The weights and the set of means and variances of the three blue distributions are the parameters of the GMM.

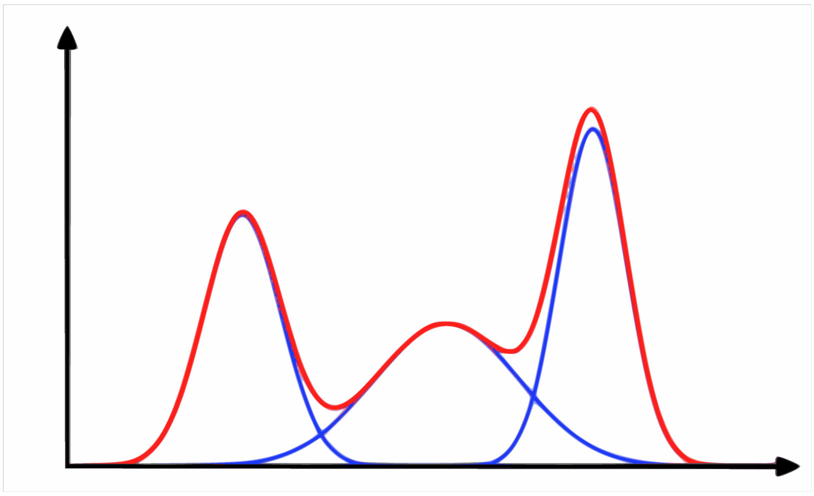


Fig. 4.3 An example of univariate (1D) Gaussian Mixture Model.

In a multidimensional dataset, the PDF of the observation x of multivariate data is a joint probability distribution that is defined as

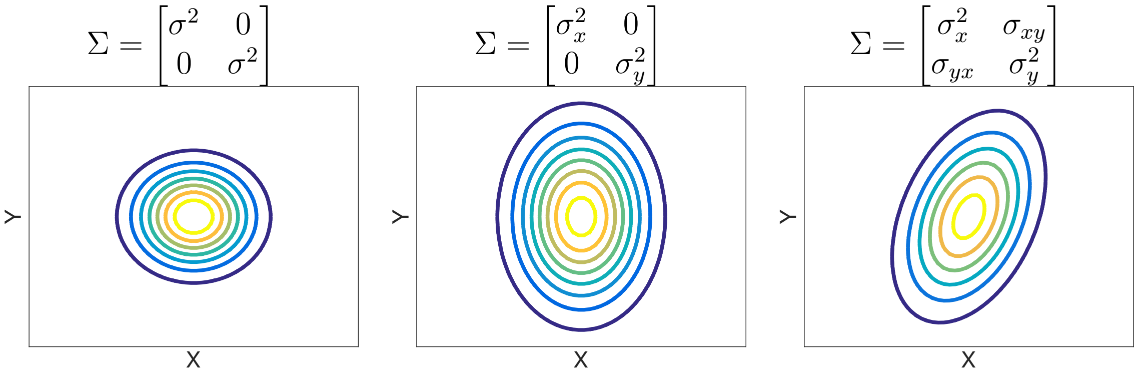
[1] Gaussian distribution are common in nature due to the central limit theorem.

$$p(x) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} exp\left\{ -\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu) \right\}, \tag{4.3}$$

also denoted as

$$\mathbf{X} \sim \mathcal{N}(\mu, \Sigma), \tag{4.4}$$

where $\mathbf{X}$ refers to multivariate data that have a normal distribution of a single Gaussian, μ is the mean vector, Σ is the covariance matrix, and D is the number of dimensions. The mean vector has elements that centre along every dimension. The covariance matrix determines the spread and the orientation of the data intensity of the joint space. The elements along the diagonal of the covariance matrix give the variance σ^2 of each dimension. The off-diagonal elements specify the correlation structure of the distribution which determines the orientation of the data space.



Diagonal covariance Diagonal covariance Full covariance

Fig. 4.4 Variations of the covariance matrice (Σ) of the multivariate normal distribution with a single Gaussian.

The GMM is used to estimate the observation probabilities of a sequence of frames derived from a speech utterance. The GMM is usually constructed in a constrained version in which the components of the GMM have a diagonal covariance matrix rather than a full covariance matrix. This reduces the number of free parameters that need to be estimated for the Gaussian models (the number of parameters reduces from $O(D^2)$ to D), as shown in Figure 4.4. Since a diagonal covariance matrix does not contain correlation coefficients between each dimension of random variables, the spread of distribution is axis aligned, and any changes in each variable do not affect the other variables. Figure 4.5 illustrates the GMM distribution of a speech model. The Gaussian mixture distribution has the joint PDF of

$$p(x) = \sum_{m=1}^{M} w_m \mathcal{N}(x; \mu_m, \Sigma_m), (w_m > 0), \tag{4.5}$$

where M refers to the number of Gaussian components, w_m denotes the weight of each component which has to be larger than zero, Σ_m is the diagonal covariance of each component, and $\sum_{m=1}^{M} w_m = 1$.

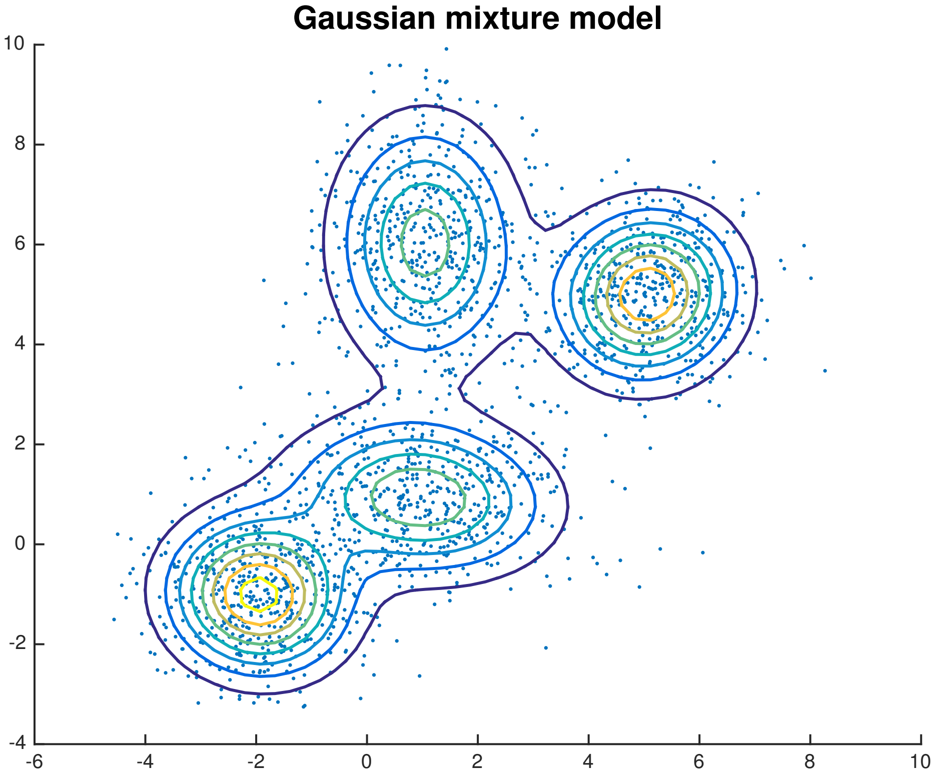


Fig. 4.5 An example of a multivariate Gaussian Mixture Model. It is a 2-d representation, where the contours represent equal probability and the dots are the data points.

Expectation-maximisation (EM) parameter estimation algorithm

The expectation-maximisation (EM) algorithm [36] is used to fit the GMM to any distribution. The EM algorithm is an iterative technique that adjusts the parameters of a model to maximise the likelihood of the data given the model and any distribution. It is a powerful tool to optimise GMM parameters but it must be emphasised that no guarantee exists that it converges to the maximum likelihood estimate for the parameters: it converges to a local maximum.

The EM algorithm iteratively maximises the likelihood $p(X|\theta)$ of the GMM parameters, where the parameter θ consists of $\{(w_m, \mu_m, \Sigma_m), m = 1, ..., M\}$. The likelihood of θ is increased by computing

$$\theta^* = \arg\max_\theta p(\mathbf{X}|\theta) = \arg\max_\theta p(\mathbf{X}|\theta) \prod_{j=1}^{J} p(x_j|\theta), \tag{4.6}$$

where j is the number of iteration. The EM algorithm involves two steps: the expectation step (E-step), and the maximisation step (M-step).

The expectation step (E-step) estimates the likelihood of the distribution given the data with the current Gaussian parameters.

$$h_m^j(t) = \frac{w_m^{(j)} \mathcal{N}(\mathbf{x}^{(t)}; \mu_m^{(j)}, \Sigma_m^{(j)})}{\sum_{i=1}^{M} w_i^{(j)} \mathcal{N}(\mathbf{x}^{(t)}; \mu_i^{(j)}, \Sigma_i^{(j)})}. \tag{4.7}$$

The EM algorithm assigns the posterior probability as a soft decision to each data point. At the current iteration (j), the posterior probability of a mixture component (h_m) of the data point ($\mathbf{x}^t$ where $t = 1, .., N$ and N is the sample size) is computed by estimating the likelihood of the data point given that particular mixture component then dividing by the summation of the likelihood of all components.

The maximisation step (M-step) computes the parameters (θ) that maximise the expected log-likelihood.

$$w_m^{j+1} = \frac{1}{N} \sum_{t=1}^{N} h_m^j(t), \tag{4.8}$$

$$\mu_m^{j+1} = \frac{\sum_{t=1}^{N} h_m^j(t) x^{(t)}}{\sum_{t=1}^{N} h_m^j(t)}, \tag{4.9}$$

$$\Sigma_m^{j+1} = \frac{\sum_{t=1}^{N} h_m^j(t)[x^{(t)} - \mu_m^j][x^{(t)} - \mu_m^j]^T}{\sum_{t=1}^{N} h_m^j(t)}. \tag{4.10}$$

This step updates parameters with the soft updates by re-estimating the parameters (θ) from the likelihood of the data points.

Weakness of GMM

The GMM has a useful property which is the ability to learn a normal distribution without labels. This property benefits DNN-HMM training since it is used to initialise the transition probabilities of the HMM and generate time alignment labels. However, GMMs also have weaknesses.

The GMM suffers from the “curse of dimensionality” problem. This problem refers to the sparse data problem as the volume of the space increases when the feature dimension is increased. Bouveyron and Brunet-Saumard [16] claim that in GMM parameter estimation, the number of free parameters grows quadratically with feature dimension (D). Their paper illustrates that this problem leads to poor performance.

There are some weaknesses of GMMs in acoustic modelling pointed by Hinton et al. [55]. It is statistically inefficient for modelling data that lie on or near a non-linear manifold in the data space. For example, if the data lies on the surface of a sphere, it can be modelled by computing sphere of radius r where the volume of D-dimensional is $O(r/D)$. This is much smaller than the volume of the D-dimensional in GMM either diagonal covariance GMM or full covariance GMM. Furthermore, a GMM is also impractical to model information of dynamic speech, which refers to a large window of frames, because the signal is not stationary over a longer window. These deficiencies motivate the consideration of alternative learning techniques.

4.2.2 Deep neural network (DNNs)

A DNN is a feed-forward artificial neural network with many hidden layers. Each layer consists of many neurons. Each neuron, or node, has a non-linear property (or nonlinearities) via a non-linear activation function which applies on top of the combination of linear input. These nonlinearities enable the DNN to represent any function as a universal function approximator. In a DNN, multiple nodes are interconnected, and the input signal passes through these nodes allowing the data to be learnt hierarchically. This structure represents the data in multiple layers of abstraction.

Figure 4.6 shows an example of a fully connected deep network architecture containing five layers including an input layer, an output layer and three hidden layers. The input layer handles a high dimensional vector that is usually in the form of concatenated feature vectors from many input frames. The hidden layers perform nonlinear feature transformations. Classification is performed on the transformed feature that appears at the output layer.

Each neuron, also called a hidden unit, represents a linear combination of the data from the layer below with weights and bias, and then passes it through an activation function:

$$x_j = b_j + \sum_i y_i w_{ij}, \tag{4.11}$$

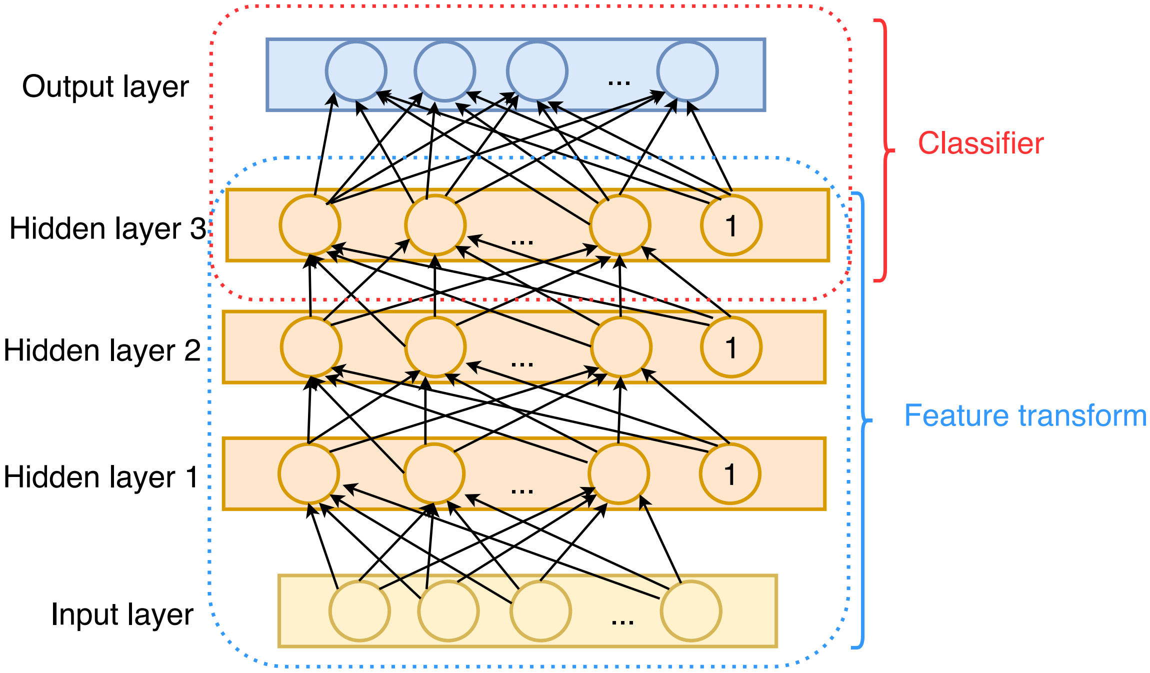


Fig. 4.6 An example of the deep network architecture adapted from Yu and Deng [163].

and

$$y_j = f(x_j), \tag{4.12}$$

where w_{ij} refers to the connection weight between the neuron unit j and the neuron unit i from the layer below, b_j is the bias of the unit, and y_j refers to a scalar output of an activation function $f(x)$.

Non-linear activation function

The activation function $f(x)$ in each neuron maps the total input from the lower layer to a scalar that is passed to the next layer. An activation function models the action potential (an electrical impulse sending between neurons) of each hidden unit as it should or should not be activated. There are many choices to use as a non-linear activation function. Here are the three commonly used functions:

the sigmoid function (output range from 0 to 1)

$$f(x) = \frac{1}{1+e^{-x}}, \tag{4.13}$$

the hyperbolic tangent function (tanh) (output range from -1 to 1)

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \tag{4.14}$$

and the Rectified Linear Unit (ReLu) (output range from 0 to x)

$$f(x) = \mathbf{max}(0, x). \tag{4.15}$$

In Figure 4.7, we plot all three activation functions. These non-linear activation functions are used in the hidden layers.

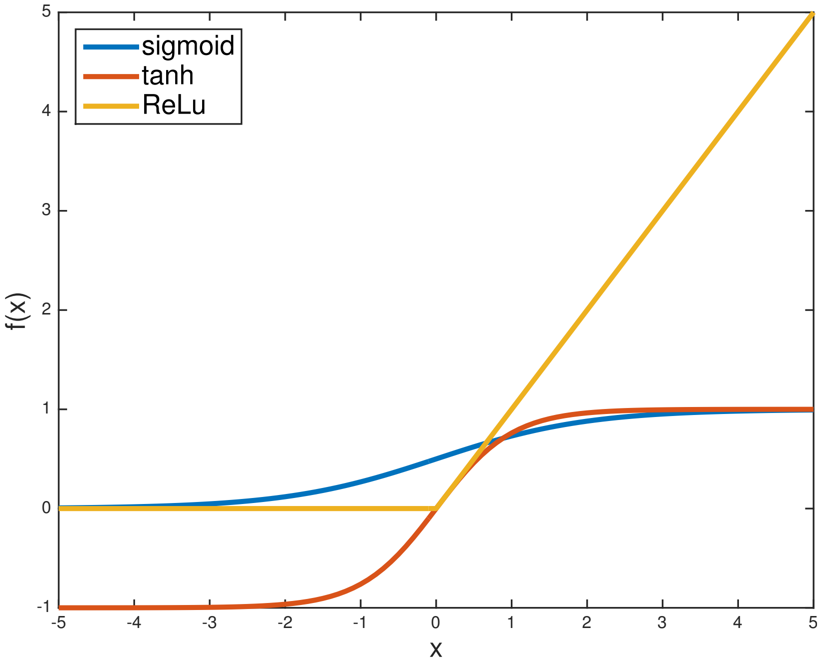


Fig. 4.7 Non-linear activation functions.

Softmax function

To estimate the class probability (p_j), the softmax layer output based on the softmax non-linearity function can be computed as:

$$p_j = \frac{\exp(x_j)}{\sum_k \exp(x_k)}, \tag{4.16}$$

where k is a total number of classes. In the output layer of a multi-classes classification task, we use the softmax function (Bridle [18]) to estimate the class probability.

DNN training

DNN training is a discriminative learning process where a class label (which is used as the target output) for every data frame is needed. The full training process involves three steps: forward propagation, backward propagation, and optimisation to update the weights.

The forward propagation process makes a prediction from the current weights, then compares the predicted output with the target label. Here, the weight matrix is initialised via random values or via pre-training. An output in the softmax layer is iteratively computed throughout the hidden layers, by multiplying the input and the weight matrix via (4.11) and then applying the activation function. A loss function (also called a cost function) is then applied to quantify the error between the predicted output and the actual target. A typical loss function is cross-entropy (CE):

$$C = -\sum_{j} d_j \log p_j, \tag{4.17}$$

where C refers to the CE cost function, p is the softmax output, and d is the actual target output. Here C determines the prediction error generated from the current parameters.

The backward propagation process uses the backpropagation algorithm proposed by Rumelhart et al. [126]. It computes the derivative of the error and propagates this back to each layer of the network using the chain rule. This process calculates the error of each hidden unit in the hidden layer and goes backwards to calculate layer by layer. This can be done by computing the partial derivative of the error function with respect to each weight of the network ($\frac{\partial C}{\partial W}$). The chain rule is used to compute the partial derivative of a composite function; it is necessary to use in the backpropagation since neural networks are a nested composite function.

The optimisation process updates all the weights to minimise the error via gradient descent. Gradient descent is the process of finding an optimal value of each weight of the network which satisfies the smallest error. The process starts with computing the gradient of the error surface and taking a step toward the minimum error which is in the opposite direction to the gradient.

In the "vanilla" version of the gradient descent method, the gradient is computed from all training samples: this is sometimes called *batch processing.* The weights of the network can be updated as

$$\hat{W} = W - \alpha \frac{\partial C}{\partial W}, \tag{4.18}$$

where W is the current weight matrix, α is the learning rate, $0 < \alpha < 1$. Updating weights via the batch process is pretty slow as the gradient of each iteration is computed from all training samples.

An alternative approach is to use stochastic gradient descent (SGD). Stochastic gradient descent computes the gradient from a small batch (training samples) of the available training data, usually called a mini-batch. The training samples of each mini-batch are selected randomly. The stochastic gradient descent method is then used to update the weights of the network as

$$\hat{W} = W - \frac{\alpha}{mb} \sum_{t} \frac{\partial C^t}{\partial W^t}, \tag{4.19}$$

where t refers to the index of mini-batch and mb is the mini-batch size.

Figure 4.8 illustrates the different gradient descent methods. In each learning iteration, the weights are tuned by following the gradient descent to reduce errors in the training data (shown by a small arrow). This process can be called as a discriminative fine-tuning process. The mini-batch SGD method aims to avoid "noisy" gradients which might occur in normal SGD. The effectiveness of back-propagation based on SGD depends on the availability of a large amount of labelled training data along with the proper mini-batch size.

Generative pre-training

There are some alternative approaches (such as transfer learning and generative pre-training) that initialise DNN weights using a specified technique rather than using random initialisation. The transfer learning approach [160] uses a deep model optimised on one task as a feature extractor, and then does fine-tuning on the last couple of layers with the actual target in another task. The transfer learning approach usually takes a pre-trained model for a similar task learned from a massive corpus. For example, VGG-Face [105] is a pre-trained convolution neural network for face recognition and GoogLeNet [143] is a pre-trained convolution neural network for image classification.

Another approach is initialising weights from unlabelled data using the unsupervised pre-training method. This method uses an unsupervised technique to learn weights from input data. Examples of unsupervised pre-training techniques include a method to learn for reconstruction such as autoencoder [58]; a method to learn the distribution

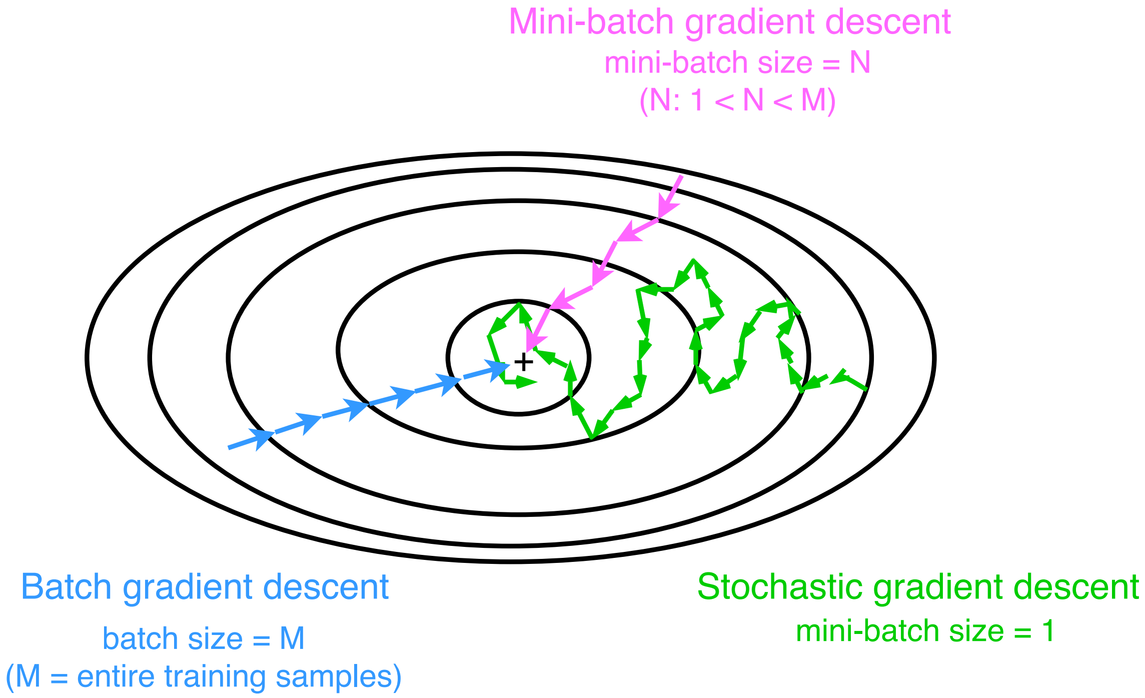


Fig. 4.8 Variation of gradient descent optimisation. The blue arrows indicate the batch gradient descent method. The green arrows refer to the SGD method. The pink arrows indicate the mini-batch SGD method. The surface here is an error surface where the smallest error is in the middle of the ellipses.

of the data such as RBM [57]. RBM is an energy-based model that can learn a complex distribution of the data. The generative pre-training approach is helpful for convergence when the available training data is limited.

In this work, we use a pre-training method based on DBNs proposed by Hinton et al. [55], Mohamed et al. [89]. A DBN model is a type of deep learning model constructed via the layer-wise generative pre-training. This model was proposed to reduce a difficulty encountered when training multi-layered neural networks with gradient-based methods and backpropagation called the vanishing gradient problem. The vanishing gradient problem is a fundamental problem where the gradient in the earlier layers becomes very small and difficult to train for layers near the output. Therefore, the weights in very early layers may not change during the training process. Hinton et al. [55], Mohamed et al. [89] use the RBM to model the distribution of the input data one layer at the time, then stack the trained RBMs as a deep network called a DBN. They then apply the standard backpropagation fine-tuning process on top of the DBN model. Here, the DBN model is constructed by extracting meaningful features using

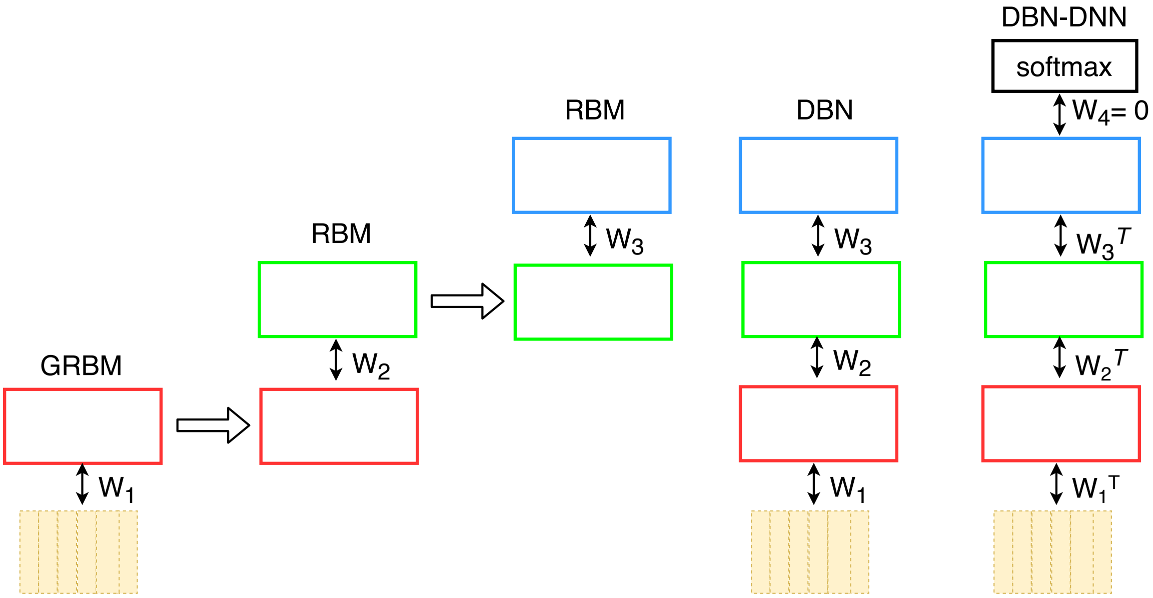


Fig. 4.9 A pre-trained DBN-DNN training method proposed by Hinton et al. [55], Mohamed et al. [89].

an RBM energy function. The RBM training process is repeated until every layer of the network is trained as shown in Figure 4.9.

RBM training is now described. An RBM is an undirected graphical model that finds distributions over the input vector using a layer of binary hidden units. The RBM training is optimised by using *contrastive divergence* (CD) [56]. The distribution of the input data is then learned layer by layer where the output of the previous step becomes an input of the current step.

The RBM energy function of the joint distribution between visible units (v) and hidden units (h) is defined as

$$E(\mathrm{v},\mathrm{h}) = \sum_{i\in\text{visible}} a_i v_i - \sum_{j\in\text{hidden}} b_j h_j - \sum_{i,j} v_i h_j w_{ij}, \tag{4.20}$$

and the energy function of the real values is Gaussian-Bernoulli RBM (GRBM) which is defined as

$$E(\mathrm{v},\mathrm{h}) = \sum_{i\in\text{visible}} \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_{j\in\text{hidden}} b_j h_j - \sum_{i,j} \frac{v_i}{\sigma_i} h_j w_{ij}, \tag{4.21}$$

where v_i denotes a visible unit i and a_i is a bias of the visible unit, σ_i is the variant of i, h_j refers to a hidden unit j and b_j is its bias, w_{ij} represents the weight between those units.

4.2.3 DNN-HMMs hybrid structure

Here we describe the visual speech modelling method using the hybrid DNN-HMM structure. In acoustic speech recognition, the hybrid DNN-HMM structure is known to provide significant performance gains over the standard GMM-HMM [89, 34, 132]. There have also been some preliminary applications to lipreading systems [149, 4].

The deep network structure can be considered as a feature extractor in which neurons in multiple hidden layers learn the essential class patterns from the input features. In addition, the backpropagation algorithm, with its learning method, is essentially optimising the model to fit to the training data discriminatively. However, to decode a speech signal, temporal features and models that can capture the sequential information in speech such as an observable Markov sequence in the HMM is still necessary. Thus, the DNN-HMM hybrid structure in which the DNN has been using instead of GMM in the HMM, essentially combines the advantages from those two algorithms.

The state observation probability is computed by the softmax function in the softmax output layer of the network, and the transition probability conventionally defines the HMM transition between the states. Then, the decoding techniques that have been developed for GMM-HMM systems can be simply applied to the hybrid DNN-HMMs structure to recognise the input speech sequence. This concept improves the performance of speech recognition systems significantly in many speech recognition tasks as well as in audiovisual speech recognition.

In the DNN-HMM hybrid approach, let $\mathbf{X} = \mathbf{x}_1, ..., \mathbf{x}_T$ be the T sequence of feature vectors extracted from each video and $\mathbf{w}$ be a word sequence that is represented by a language model. The likelihood of an input sequence can be computed by

$$p(\mathbf{X}|\mathbf{w}) = \prod_{t=1}^{T} p(\mathbf{x}_t|s_t)p(s_t|s_{t-1}), \tag{4.22}$$

where $p(x_t|s_t)$ denotes the emission probability and $p(s_t|s_{t-1})$ is the transition probability obtained from the HMMs state transition. The emission probability can be approximated by $p(\mathbf{x}_t|s_t) = p(s_t|\mathbf{x}_t)p(\mathbf{x}_t)/p(s_t)$, via a GMM, in which case we have

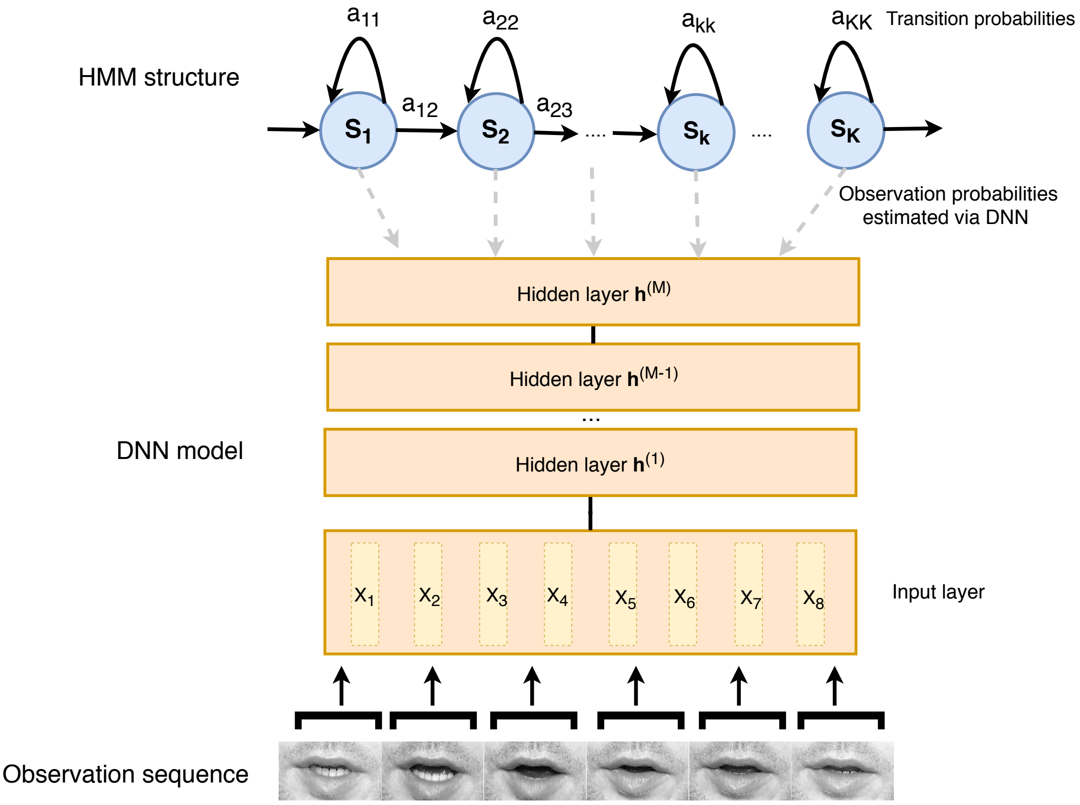


Fig. 4.10 The hybrid DNN-HMM architecture adapted from [33]. The HMM structure captures the sequential information and the GMM is replaced with a DNN to model the speech observation.

the conventional HMM speech recognition architecture or, via a DNN. To estimate the DNN's posterior $p(s|\mathbf{x}_t)$ on each state of an utterance u, the DNN uses a pseudo log-likelihood obtained via the softmax activation function

$$p(s|\mathbf{x}_{ut}) = \frac{\exp\{a_{ut}(s)\}}{\sum_{s'} \exp\{a_{ut}(s')\}}, \tag{4.23}$$

where $a_{ut}(s)$ refers to an activation of state s at the output layer. In which case, the pseudo log-likelihood of the visual speech model is

$$\log p(\mathbf{x}_{ut}|s) = \log p(s|\mathbf{x}_{ut}) - \log p(s), \tag{4.24}$$

The DNNs used in visual speech modelling have conventionally been trained to optimise the CE between the prediction and the target HMM-state labels using mini-batch Stochastic Gradient Descent (mini-batch SGD) optimisation and error backpropagation (BP) algorithm [126], to provide the posterior probability estimates of the HMM states. The HMM-state alignments are obtained from a GMM-HMM training process. Here, we use BP to minimise the cross-entropy between the predicted output and the HMM-state target. This is similar to DNN-HMM training in acoustic speech recognition as in [55].

Cross-entropy (CE)

The cross-entropy objective is a frame-level training criterion for classification tasks and usually provides significant performance gain over standard GMM-HMM acoustic modelling in speech recognition. In visual speech model training, we use frame level alignment generated from a context-dependent GMM-HMM system and the initial DNN-HMM parameters via stacking RBMs pretraining [55]. We use CE to fine-tune the DNN parameters.The CE objective function is defined as

$$\mathcal{F}_{CE} = -\frac{1}{T}\sum_{u=1}^{U}\sum_{t=1}^{T_u}\sum_{s} l_{ut}(s)\log p(s|\mathbf{x}_{ut}), \tag{4.25}$$

where the T here is the total number of frames from all training utterances and $l_{ut}(s)$ is the Kronecker delta of the target state.

Sequence-discriminative DNN training

Here, we describe some other discriminative techniques for training DNNs.

MMI

The MMI training criterion [9, 67] aims to maximise the mutual information between the distributions of observation and the reference word sequences. Let $\mathbf{X}_u$ represent the sequence of visual features and w_u is the word reference in an utterance u. MMI attempts to maximise

$$\mathcal{F}_{MMI} = \sum_{u}\log\frac{p(\mathbf{X}_u|S_u)^k P(\mathbf{w}_u)}{\sum_{\mathbf{w}} p(\mathbf{X}_u|S_{\mathbf{w}})^k P(\mathbf{w})}, \tag{4.26}$$

where S_u is the state sequence corresponding to the correct word $\mathbf{w}_u$ and k is the model scaling factor. For computational efficiency, the sum in the denominator may be practically estimated from a decoding lattice (generated from a weak language model instead of using all the possible word sequences. We also apply frame rejection proposed by [152] to avoid infinite gradients, caused by missing words in the denominator lattice.

sMBR/MPE

The sMBR/MPE training criteria aims to minimise the expected error, measured at state-level (sMBR [72]) or phone-level (MPE, [119]), between the sequence of visual features and the word sequence of each training utterance. Specifically, sMBR/MPE attempts to minimise

$$\mathcal{F}_{MBR/MPE} = \sum_{u} \log \frac{\sum_{\mathbf{w}} p(\mathbf{X}_u|S_{\mathbf{w}})^k P(\mathbf{w}) A(\mathbf{w}, \mathbf{w}_u)}{\sum_{\mathbf{w}'} p(\mathbf{X}_u|S_{\mathbf{w}'})^k P(\mathbf{w}')}, \tag{4.27}$$

where $A(\mathbf{w}, \mathbf{w}_u)$ is the *raw accuracy* between the word sequence $\mathbf{w}$ and the reference $\mathbf{w}_u$. Raw accuracy refers to the number of correct state labels in sMBR and the phone labels in MPE.

4.3 WFST decoder

Here is a brief explanation of the framework in visual speech decoder based on the WFSTs. A WFST is a weighted finite state automaton that transduces an input sequence to an output sequence [90]. Each state in a transducer is connected by a transition that has an input symbol, an output symbol, and a weight. The WFSTs decoder is comprised of four transducers: HMM structure (H), phonetic context-dependency (C), lexicon model (L), and grammar or n-gram language model (G), called collectively the $HCLG$ decoding-graph. Details of the transducers are listed below.

HMM transducer: H represents an HMM where the input sequence is the HMM states and the output sequence is the context-dependent phones (CD).

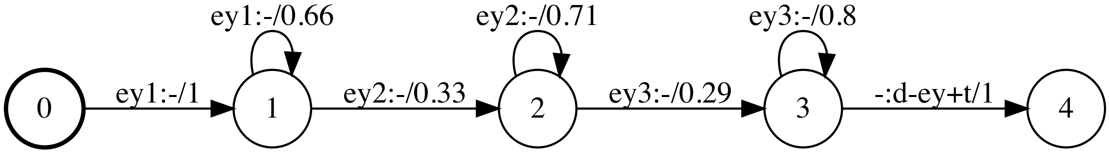


Fig. 4.11 An example of HMM transducer.

Context-dependency transducer: C represents phonetic context-dependency where the input sequence is the CD phones and the output is the phones.

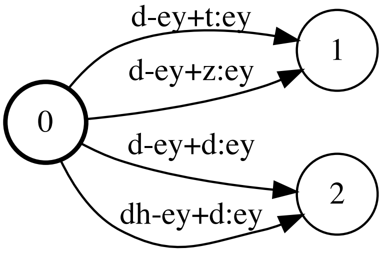


Fig. 4.12 An example of context-dependency transducer.

Lexicon transducer: L represents pronunciation lexicon where the input sequence is phones and the output sequence is words. The L transducer can contain a set of alternative pronunciations in each word if it is available in the dictionary. Here is an example of an L transducer from [90].

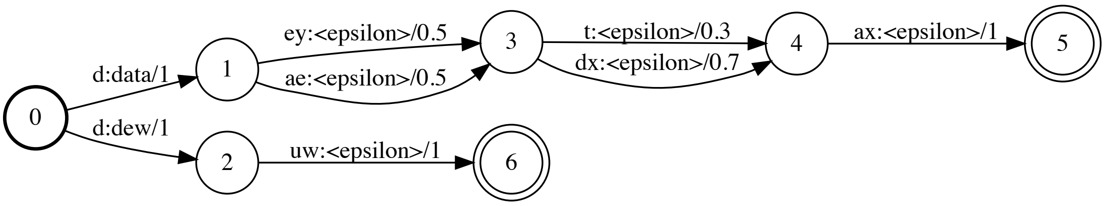


Fig. 4.13 An example of lexicon transducer.

Grammar transducer: G represents word-level grammar where the input sequence is words and the output sequence is words. Here is example of G transducer from [90].

There are three steps to decode the speech utterances: decoding graph generation, decoding and lattice generation, and lattice re-scoring. The first-pass decoder generates a word lattice containing possible seqeunces of words that match the input lip signal; then the final result comes from the lattice re-scoring via a language model.

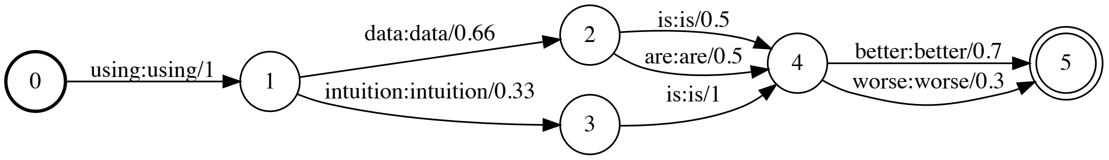


Fig. 4.14 An example of grammar transducer.

HCLG decoding-graph

A *HCLG* decoding-graph is generated using three algorithms: composition, determinisation, and minimisation, to combine the different levels of linguistic representations while maintaining the feasibility to decode. The *HCLG* is composed as

$$\text{HCLG} = \min(\det(\text{H} \circ \text{C} \circ \text{L} \circ \text{G})), \tag{4.28}$$

where H, C, L, and G are the transducers defined earlier, $\circ$ refers to the composition algorithm that is used to combine transducers, *det* refers to a determinisation algorithm that is used to transform a nondeterministic weighted automation into a deterministic automation to reduce redundancy, and *min* is the minimisation algorithm that is used to reduce the size of a deterministic automaton to save search space and time. These algorithms help to make a compact HCLG decoding-graph suitable for a large-vocabulary decoder and long n-gram language model. As presented by Mohri et al. [90], this also has the benefit of speeding up the processing time without damaging the system performance.

Lattice generation in the first-pass decoder

To generate a lattice, each arc of an *HCLG* is traversed for each input feature vector and state-level arcs are created for the acoustic and graph costs. In the Kaldi toolkit [116], a lattice generator, described in [117], handles graph and acoustic costs separately to keep track of both the acoustic score and the language model score by simply using a data structure of a full-state HCLG. Then a beam search algorithm with beam width pruning (as suggested between $4 \leqslant \alpha \leqslant 8$) is applied every 25 frames (rather than waiting to the end as this helps conserve memory). The beam search algorithm retains to keep only the most likely result with the corresponding graph and acoustic costs in the lattice.

In the first step, the pruned graph P is generated from

$$P = prune(B, \alpha), \tag{4.29}$$

where B is the un-pruned $HCLG$ graph which is a full state-level lattice, and α is the lattice beam width. The Kaldi decoder sets α=8 in a DNN-based decoder and α=6 in a GMM-based decoder as a default parameter. We use a larger α in the DNN-based decoder since the DNN-decoder runs on a higher performance machine (we run this decoder on a GPU machine). We then use the pruned graph P, which is an acyclic graph that keeps only the best path within the beam α. This pruned graph can be used in the language model re-scoring step directly. However, the Kaldi decoder embeds the state-level alignment information into the lattice and also keeps track of acoustic and graph costs separately. The decoder in Kaldi extends the algorithm to process inv(P) which is the inverted version of the pruned state-level lattice, where the input symbols are words and the output symbols are the PDF labels. The decoder in Kaldi also defines E as the encoded version of inv(P) that encodes the state labels into the weights. The final lattice is defined as

$$L = prune(det(rmeps(E)), \alpha), \tag{4.30}$$

where *rmeps* is an operation to remove the epsilon symbol. The lattice L is the word lattice (containing the best pass within the beam α) that has information of the graph cost, the acoustic cost and the state-level alignment embedded into the weights. Note that the larger the value of α used, the slower the processing time and the deeper the lattice.

Lattice re-scoring in the second-pass decoder

In a multi-pass decoder, the second-pass uses an external score to reorder the lattice. Here, the final transcription is generated by re-ranking the lattice using the score of a language model. The lattice L that contains the entire surviving path is re-scored by applying a language model scaling factor over the range 15-20 (between 4-40 and 15-20 appear to be the optimum range for lipreading). This rescoring technique over the lattice gives a low score to nonsensical sequences of words that have a low probability in the language model. We pick only the first best as the transcription result.

4.4 Visual feature extraction

4.4.1 Front-end processing

Active Appearance Models (AAM)

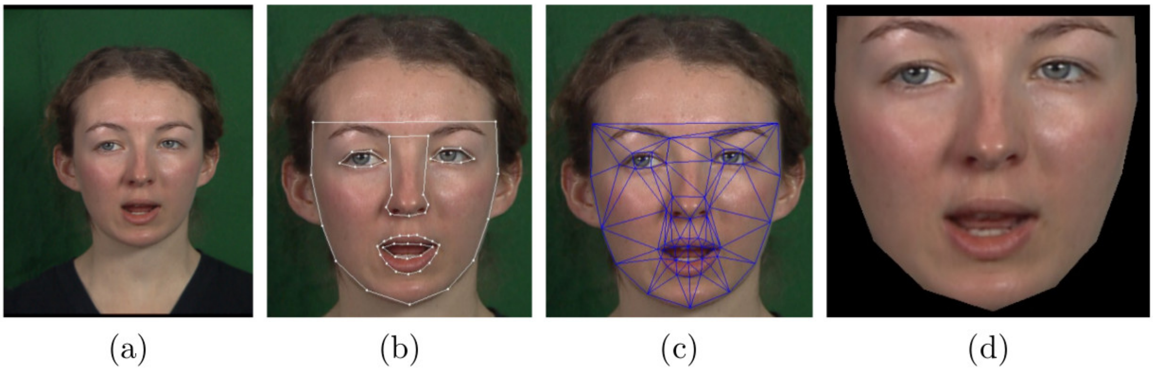


Fig. 4.15 An example of AAM feature from Deena [35].

An active appearance model (AAM), as described by Cootes et al. [31], consists of a shape component plus an appearance component that models the lip region in a video frame. The shape component is constructed by, first, hand-labelling a set of images with the x- and y-coordinates of the set of n vertices of a mesh, and then applying PCA to the shapes:

$$s = s_0 + \sum_{i=1}^{m} p_i s_i \tag{4.31}$$

where s is a vector of (x, y) coordinates of the shape vertices, s_0 is the mean shape, p_i are the modes of shape variation corresponding to the m largest eigenvectors, and s_i a vector of shape parameters. The appearance component is constructed by warping the pixels inside the mesh in each training image to the mean shape (s_0). PCA is then applied to the images, providing a compact, linear model of appearance variation of the form:

$$A = A_0 + \sum_{i=1}^{m} \lambda_i A_i \tag{4.32}$$

where λ_i are the appearance parameters, A_0 is the mean appearance and A_i are the eigenvectors corresponding to the m largest eigenvalues.

Discrete Cosine Transform (DCT)

The Discrete Cosine Transform (DCT[2]) is a simple transform used in image coding and compression. The DCT aims to represent the frequency domain of a signal periodically and symmetrically using the cosine function. The DCT has a great advantage in energy compaction (Rao and Yip [123]). In particular, the DCT is a part of the Fourier Transform family but contains only the real part (Cosine). Because of its popularity, most modern processors execute it very quickly (roughly $\mathcal{O}(N)$ for modern algorithms), so this also explains its ubiquity. For strongly correlated Markov processes the DCT approaches the Karhunen-Loeve transform in its compaction efficiency. Possibly this also contributes to its popularity as a benchmark feature [95]. Figure 4.16 shows the DCT basis functions.

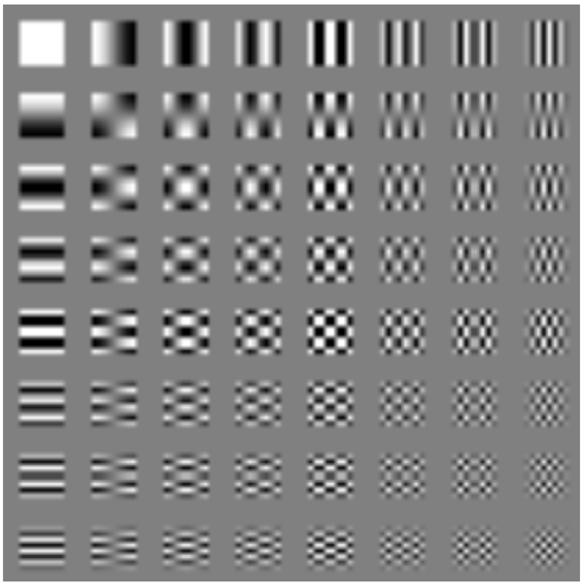


Fig. 4.16 The 2D DCT basis functions. This image illustrates the 64 DCT basis functions that are formed by 8-by-8 matrices. The contrast patterns represent positive (white) and negative (black) values of the funtion.

DCT feature selection can be done in many ways; for example, conventional energy selection [53, 114], or mutual information based selection [130]. Here we use simply zigzag scanning [158]. The quality of the image reconstructed from DCT features is shown in Figure 4.17.

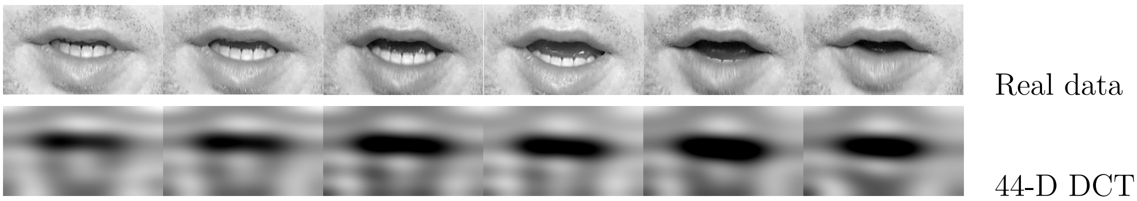


Fig. 4.17 An example of reconstructed lip ROIs from DCT features. The real data refers to the original lip ROIs provided in the TCD-TIMIT corpus. We keep the same dimension as presented in the TCD-TIMIT baseline results.

Dual-tree complex wavelet transform (DTCWT)

DTCWT (proposed by Kingsbury [73]) is the enhanced version of the DWT. The DTCWT has approximate shift invariance in magnitude i.e. it is relatively insensitive to the object position in an image. This property is well-suited to texture analysis in a lip-image since it makes the texture feature independent of the texture location. Additionally, it provides a multi-resolution, sparse representation and is a useful characterisation for image reconstruction.

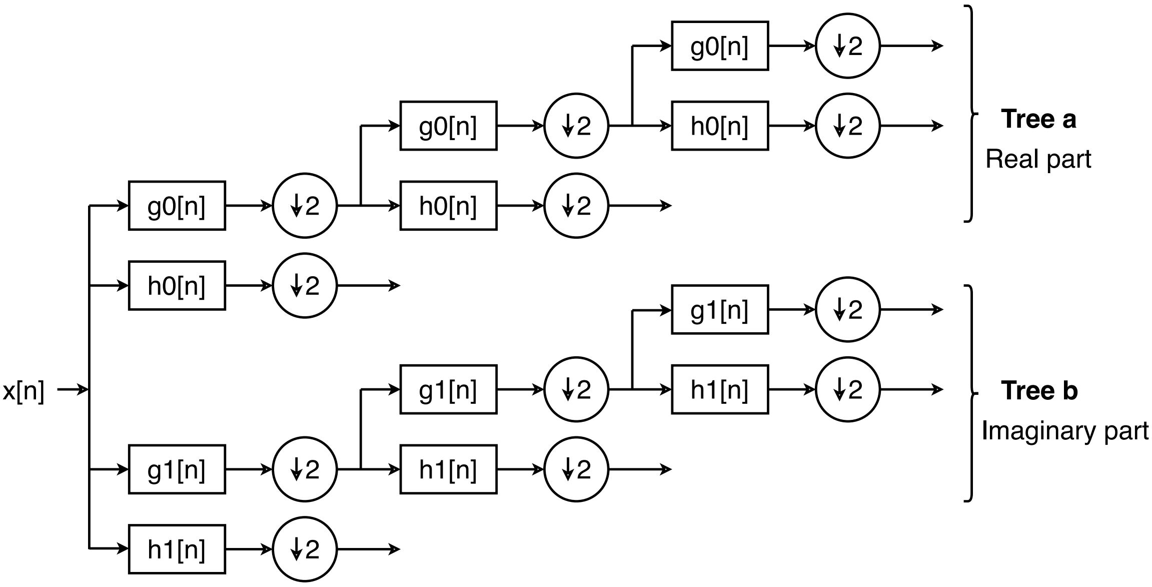
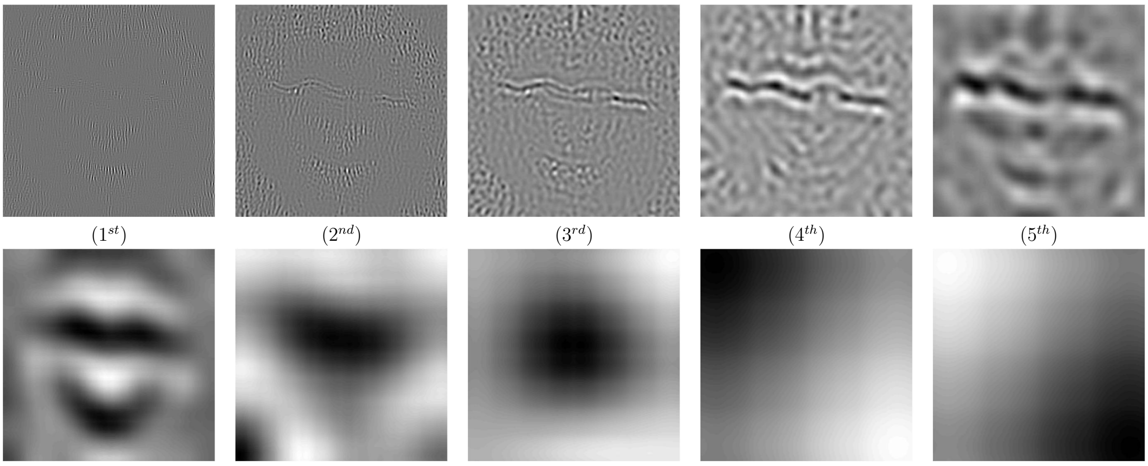


Fig. 4.18 The DTCWT proposed by Kingsbury [73] taken from [73]

The DTCWT coefficients are computed from the two wavelet trees as shown in Figure 4.18. These trees separate into real and imaginary parts where both parts work as a complex transformation. These transforms handle orientation as there are different subbands covering six directions: ±15°, ±45°, ±75°. Figure 4.19 shows image reconstruction from different levels. The quality of an image reconstructed from DTCWT features is shown in Figure 4.20.



(1^{st}) (2^{nd}) (3^{rd}) (4^{th}) (5^{th})

(6^{th}) (7^{th}) (8^{th}) (9^{th}) (10^{th})

Fig. 4.19 Image reconstruction from DTCWT coefficients at different levels (from 1^{st} to 10^{th}).

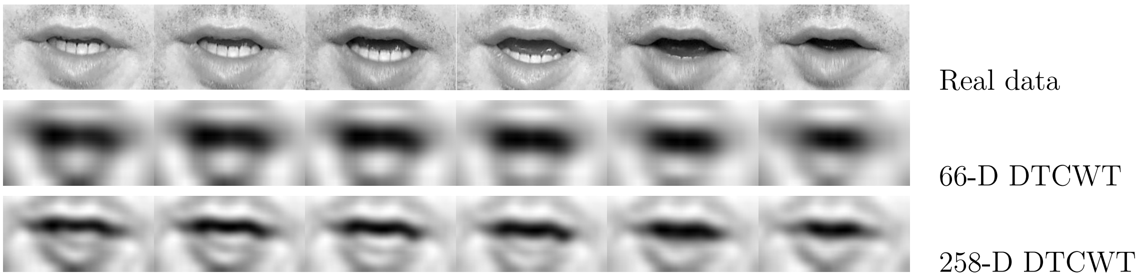
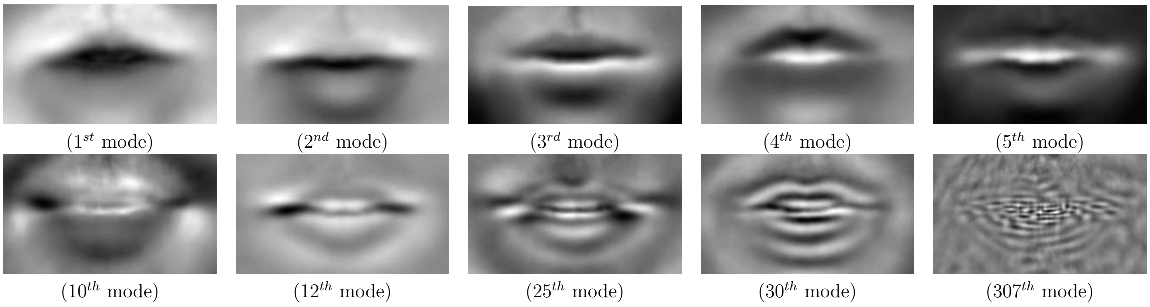


Fig. 4.20 Examples of reconstructed lip ROIs from DTCWT features. Note that 66-dimensions come from the concatenation of the top three levels (from 5^{th} to 7^{th}) containing most information (highest energy) and 258-dimensions come from the combination of the top four levels (from 4^{th} to 7^{th}).

Eigenlips

The Eigenlips feature is another appearance-based approach [74]. The Eigenlips feature has been generated via PCA [156]. It models latent factors that exist in the data. The PCA optimisation algorithm maximises the variance of the lip ROI data. Only the top k eigenvalues are retained, where, in our case, $k = 30$. The reconstructions of each individual mode are shown in Figure 4.21 and the quality of images reconstructed from Eigenlips features is shown in Figure 4.22.



(1^{st} mode) (2^{nd} mode) (3^{rd} mode) (4^{th} mode) (5^{th} mode)

(10^{th} mode) (12^{th} mode) (25^{th} mode) (30^{th} mode) (307^{th} mode)

Fig. 4.21 Image reconstruction from Eigenlips coefficients at different modes (between 1^{st} and 307^{th}).

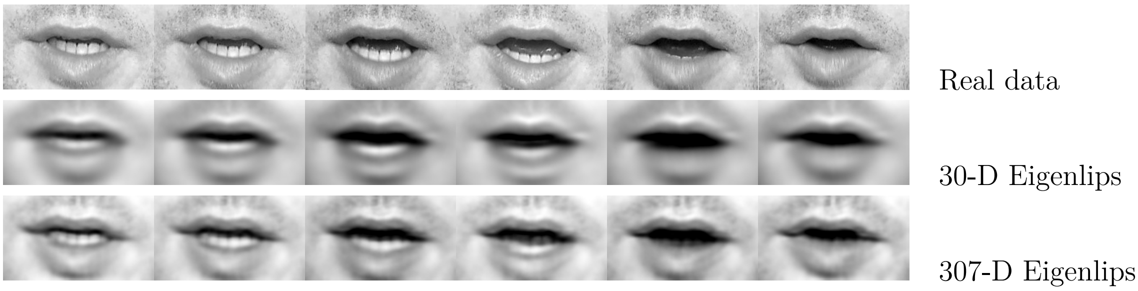


Fig. 4.22 Examples of reconstructed lip ROIs from Eigenlips features. Note that 30-dimensions cover 85% variance and 307-dimensions cover 95% variance.

Deep Autoencoder (DAE)

The Autoencoder (AE) was introduced to address the problem of dimensionality reduction of multivariate data. The difference between PCA and AE is that PCA finds dimensions that maximise the variance while the AE objective is to minimise the reconstruction error via a non-linear function. Therefore, AE is more suitable for dealing with real-world problems, which lie on the non-linear manifold. A Deep autoencoder (DAE) or multilayer autoencoder is a feed-forward neural network that learns non-linear mappings to reconstruct the input with minimum error. The network structure is separated into two parts: a decoder and an encoder. Figure 4.23 shows the schematic diagram of a deep autoencoder. A DAE feature is obtained from the layer in the middle (called a code layer or a bottleneck layer) that usually contains the smallest number of units, i.e. 30 hidden-units. This code layer is a low-dimensional representation that is trained to yield the best reconstruction of the output. DAE optimises the mean square error (MSE) to minimise the reconstruction error between the input and its reconstruction. The quality of images reconstructed from DAE features is shown in Figure 4.24.

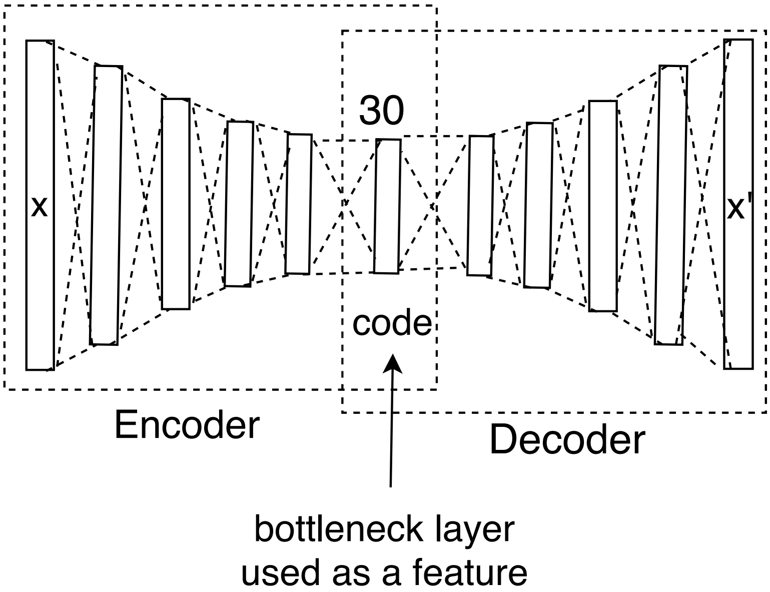


Fig. 4.23 Schematic diagram of a typical deep autoencoder that maps x to x'. Each box illustrates a layer of neurons that is fully connected (illustrated with the dotted lines) to the next layer.

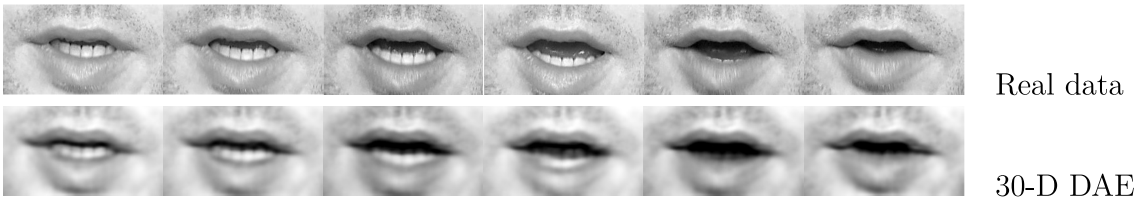


Fig. 4.24 An example of reconstructed lip ROIs from 30-dimensional DAE features.

4.4.2 Feature transformation techniques

Utterance-level mean and variance normalisation

To reduce the high variation in each dimension of a feature vector, we perform z-score normalisation to force a zero mean and unity standard deviation by subtracting a mean of each dimension of a static feature. This is done on a per-utterance basis. Let X refer to the static coefficient vectors of visual features in an utterance where $X = x_1, x_2, ..., x_t$, and t is the time frame. The normalised vector is calculated as:

$$\text{Normalised}(x_i) = \frac{x_i - \bar{X}}{\text{std}(X)}, \tag{4.33}$$

where

$$\bar{X} = \frac{1}{t}\sum_{i=1}^{t} x_i, \tag{4.34}$$

and

$$\mathrm{std}(X) = \sqrt{\frac{1}{t-1}\sum_{i=1}^{t}(x_i - \bar{X})^2}. \tag{4.35}$$

In the event of scale differences between dimensions, using this common technique modifies the feature vector to be equally scaled which generally improves machine learning, especially in algorithms that use scaled various of the input.

Capturing visual dynamics via delta coefficients

Delta coefficients were proposed by Furui [41] to deal with the temporal coherence or dynamics in the speech signal and have become widely used since then. The idea is straightforward but useful when enhancing the performance of speech recognition.

Delta coefficients are computed as

$$\Delta = \frac{\sum_{\theta=1}^{\Theta} \theta(x_{t+\theta} - x_{t-\theta})}{2\sum_{\theta=1}^{\Theta} \theta^2}, \tag{4.36}$$

where Δ is the delta coefficients at time t computed from static coefficients $x_{t+\theta}$ and $x_{t-\theta}$, and θ refers to the delta window.

Incorporating this kind of dynamic information into a feature vector is beneficial to acoustic speech and also visual speech. Most reported lipreading results usually combine the first- and the second-order derivative into the visual feature.

Linear discriminant analysis and maximum likelihood linear transform (LDA-MLLT)

For capturing dynamic information over time while searching for discriminatory features in a raw input space, we use the LDA transformation [40] as an intermediate representation of the visual feature. LDA is a supervised dimensionality reduction method that finds discriminatory features for specific classes via a linear analysis. The method maximises the ratio of between-class variance to the within-class variance thus ensuring maximal linear separability.

The within-class variance is given by

$$S_w = \sum_{c=1}^{C} \sum_{i=1}^{N_c} (x_i^c - \mu_c)(x_i^c - \mu_c)^T, \tag{4.37}$$

where c refers to a class, μ_c is the mean of the class c, x_i^c denotes the data point x_i of the class c. Note that the total number of the classes (C) here indicates the HMM states.

The between-class variance is given by

$$S_b = \sum_{c=1}^{C} (\mu_c - \mu)(\mu_c - \mu)^T, \tag{4.38}$$

where μ refers to the mean of the data set.

A further step is to apply a maximum likelihood linear transform (MLLT) [46] that rotates the features into a new space to maximise the observation likelihood in the original feature space. LDA-MLLT has been used regularly in the audio speech, visual speech and audio-visual speech recognition (also as a part of the HiLDA feature extraction [113]). It accommodates expanding the wider context window of the dynamic information while retaining the compact size of feature dimension. To examine the influence of dynamic information, we apply LDA-MLLT method to the spliced $\pm n$ context feature where n is ranged between $1 - 15$ and then reduce the dimension to $15 - 45$.

Feature space maximum likelihood linear regression (fMLLR)

Feature space maximum likelihood linear regression (fMLLR) [42] is a feature transformation method based-on the maximum likelihood linear regression (MLLR) technique. The MLLR transformation is a model space adaptation method to reduce the mismatch due to speaker, channel or additive noise effects. The method adapts the Gaussian mean of each HMM state to maximise the likelihood of the data from a particular speaker or an environment. The fMLLR transform is speaker specific. The main idea of the fMLLR transformed features is to normalise features to better fit a speaker dependent model.

We use the fMLLR transformation for speaker adaptive training (SAT). For a speaker, s, the fMLLR transformed feature is defined by

$$\hat{x}^t = W^{(s)} \xi^{(t)}, \tag{4.39}$$

where $W^{(s)} = [A^{(s)}b^{(s)}]$ denotes a transform matrix of a particular speaker [118], and $\xi^{(t)} = \begin{bmatrix} x^t \\ 1 \end{bmatrix}$ is the extended input vector by 1 as to multiply with a bias term. Note that $A^{(s)}$ refers to a variance transform $(A^T\Sigma^{(m)}A)$ where m indicates the Gaussian index.

According to the fMLLR transform presented in [118], the auxiliary function equals:

$$\log(|\det(A)|) - \sum_{i=1}^{d} w_i^T k_i - 0.5 w_i^T G_i w_i, \tag{4.40}$$

where the linear term is:

$$k_i = \sum_{m=1}^{M} \frac{c^{(sm)} \mu_i^{(m)} \varepsilon(\xi)^{(sm)}}{\sigma_i^{2(m)}}, \tag{4.41}$$

and the quadratic term is:

$$G_i = \sum_{m=1}^{M} \frac{c^{(sm)} \varepsilon(\xi\xi^T)^{(sm)}}{\sigma_i^{2(m)}}. \tag{4.42}$$

The $c^{(sm)}$ is a soft count of Gaussian m from the current speaker, $\varepsilon(\cdot)^{(sm)}$ is the mean value for speaker s and Gaussian m, d is the dimension of speech features.

4.5 Full pipeline DNN-HMM training

We use Kaldi toolkit [116], an open source toolkit for ASR, to train visual speech models following a pipeline of the wall street journal (WSJ) recipe. Kaldi toolkit provides a proper workflow to build ASR called a recipe. However, we have to implement novel feature extraction methods to support visual speech signals since Kaldi does not offer any visual features. We then pass visual features into the DNN-HMM training pipeline taken from the sub-folder s5 in the WSJ recipe. The WSJ recipe contains several training steps including GMM-HMM training, Subspace GMM-HMM training, DBN-DNN-HMM training (nnet1), DNN-HMM training (nnet2), Time Delay Neural Network (TDNN)-HMM training (chain). We use nnet1 setting to build DNN-HMM visual speech models (the main training script called run_dnn.sh can be found in the folder egs/wsj/s5/local/nnet). Here we give detail step-by-step to build visual speech DNN-HMM modeling where all of the modules related to these steps exist in the Kaldi.

DNN-HMM training involves six steps: the steps in GMM-HMMs training are used as an initialisation for the DNN-HMM training. This pipeline can be called a sequence training process as the next step uses outputs generated in the previous step. The main outputs from a GMM step include visual speech GMM-HMM, state-level alignment, feature transformation matrix. The following step uses these products to generate a feature and to initialise the training process. The HCLG graph is usually re-estimated in each particular step except for the DNN-HMM steps where the HCLG graph is obtained from the GMM with an SAT system. Figure 4.25 illustrates all those steps which we explain the main idea of each step and the parameter set as followed.

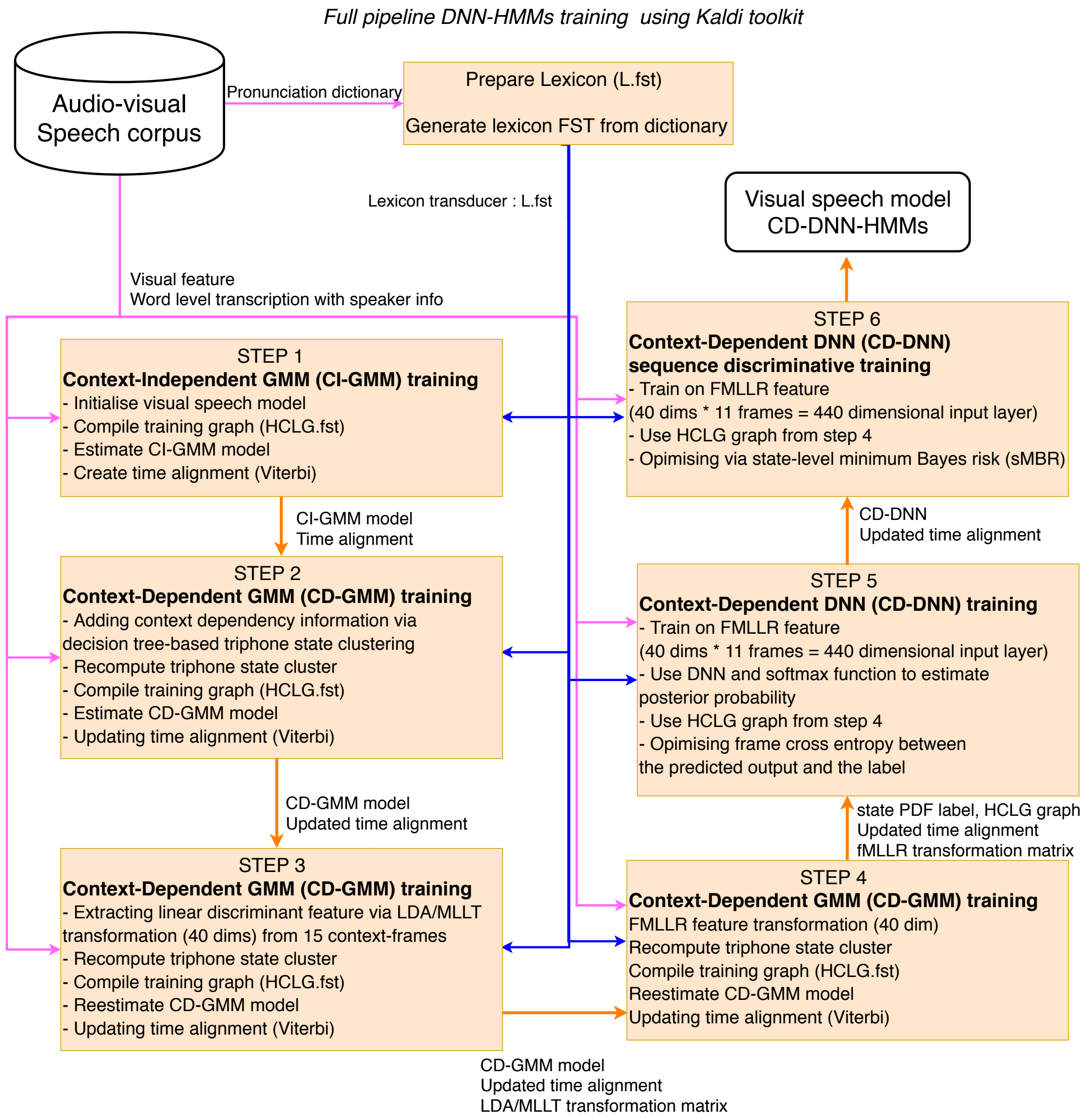

Full pipeline DNN-HMMs training using Kaldi toolkit
Audio-visual Speech corpus
Pronunciation dictionary
Prepare Lexicon (L.fst)
Generate lexicon FST from dictionary
Lexicon transducer : L.fst
Visual speech model CD-DNN-HMMs
Visual feature
Word level transcription with speaker info
STEP 1
Context-Independent GMM (CI-GMM) training
- Initialise visual speech model
- Compile training graph (HCLG.fst)
- Estimate CI-GMM model
- Create time alignment (Viterbi)
CI-GMM model
Time alignment
STEP 2
Context-Dependent GMM (CD-GMM) training
- Adding context dependency information via decision tree-based triphone state clustering
- Recompute triphone state cluster
- Compile training graph (HCLG.fst)
- Estimate CD-GMM model
- Updating time alignment (Viterbi)
CD-GMM model
Updated time alignment
STEP 3
Context-Dependent GMM (CD-GMM) training
- Extracting linear discriminant feature via LDA/MLLT transformation (40 dims) from 15 context-frames
- Recompute triphone state cluster
- Compile training graph (HCLG.fst)
- Reestimate CD-GMM model
- Updating time alignment (Viterbi)
CD-GMM model
Updated time alignment
LDA/MLLT transformation matrix
STEP 4
Context-Dependent GMM (CD-GMM) training
FMLLR feature transformation (40 dim)
Recompute triphone state cluster
Compile training graph (HCLG.fst)
Reestimate CD-GMM model
Updating time alignment (Viterbi)
state PDF label, HCLG graph
Updated time alignment
fMLLR transformation matrix
STEP 5
Context-Dependent DNN (CD-DNN) training
- Train on FMLLR feature
(40 dims * 11 frames = 440 dimensional input layer)
- Use DNN and softmax function to estimate posterior probability
- Use HCLG graph from step 4
- Opimising frame cross entropy between the predicted output and the label
CD-DNN
Updated time alignment
STEP 6
Context-Dependent DNN (CD-DNN) sequence discriminative training
- Train on FMLLR feature
(40 dims * 11 frames = 440 dimensional input layer)
- Use HCLG graph from step 4
- Opimising via state-level minimum Bayes risk (sMBR)

Fig. 4.25 Visual speech modelling scheme used in DNN-HMM based machine lipreading system trained on Kaldi [116].

Initialise with GMM-HMMs training

Step 1: Context-Independent Gaussian Mixture Model (CI-GMM)

The first step is to initialise a model by creating a simple Context Independent (CI) GMM. This step creates a time alignment for the entire training corpus by simply constructing a mono phoneme/viseme model that contains a 3-state GMM-HMM for each speech unit.

The CI-GMMs are trained on the raw features along with their first and second derivative coefficients ($\Delta + \Delta\Delta$). We use 3-state GMM-HMMs on each visual speech unit. Instead of setting a fixed number for increasing Gaussian mixtures, we have set the maximum number of Gaussians to be 1000 so that each state will continue to increase the number of Gaussians independently until their variances reach the maximum. When the training process starts, the training data is uniformly segmented and the segmentation is then updated using the Viterbi algorithm [122] in every iteration for the first ten iterations, then updated every two iterations until a maximum of 40 iterations.

Step 2: Context-Dependent Gaussian Mixture Model (CD-GMM)

The context-dependent visual speech model (CD-GMMs) is specified using the same features as in the CI-GMM system. The context-dependent model uses information from phonetic context in order to handle visual coarticulation. It is well known that adding phoneme context provides speech coarticulation information. Coarticulation refers to changes in the articulation of a speech segment depending on preceding and succeeding segments. This concept applies to both acoustic speech [124] and visual speech [29, 146] modalities.

Here we use tied states of triphone-context visual speech model, where the tied-states are obtained from the data-driven approach tree-clustering [116]. A tied-state cluster is a reduced set of triphone-states that has been tied together to share the same training data. A triphone context model is usually built to cover all possible cross-word triphones that can be encountered, so it is unlikely to have enough data to train. Here we use the conventional state-tying technique [162] to reduce the number of phonetic states. We have specified the maximum number of leaf nodes to be 2000, which limits the number of states. The maximum number of Gaussians is set to $10k$ for the system. The training iterations continue until there is convergence which in practice is after fewer than 35 iterations. We realign every 10 iterations.

Step 3: CD-GMM with LDA-MLLT feature transformation

This training step is also based-on CD-GMMs, but trained on the LDA-MLLT features. The CD-GMM model obtained from step two and the time alignment information are used to estimate the LDA transform. Indeed, the model from the previous step is used to accumulate statistical information from the training set which is essential to estimate a new set of tied-states triphones. The 40-dimensional LDA-MLLT features are formed by splicing 15 frames of the current frame (seven on the left and seven on the right) then reducing, via LDA, to 40 dimensions per frame. This compact set of LDA-MLLT features parameterises to the 40-dimensions that best associate with the visual speech units and also comprises the dynamics of visual speech over 150 ms. Here, the different set of tied-state CD-GMM has been constructed considered to the current feature. The maximum number of leaf nodes is set to 2500, and the total number of Gaussians is $15k$. This step utilises the same number of training iterations and the realignment iterations as those used in the previous step.

Step 4: CD-GMM with Speaker Adaptive Training (SAT)

This step normalises speaker variations in CD-GMM training via the SAT method [6]. The SAT method is a type of speaker adaptation technique based-on feature normalisation via a linear transformation. Since speaker identity has a substantial effect on speech recognition results in both acoustic speech and visual speech, the SAT method had been proposed to overcome the mismatch between speakers. This method estimates affine transforms of mean and variance parameters for each speaker via the constrained MLLR method [42]. The CD-GMM model is then trained on the fMLLR transformed features. In the training process, the SAT model and fMLLR feature transformation are re-estimated simultaneously via the supervised method. In each iteration, the speaker adaptive model is trained on a recent estimation of the fMLLR transformed features. There are two passes in the decoding process where the result from the first pass is used to estimate the fMLLR transform and then used to decode with the trained SAT model in the second pass. The decoding process works by using an unsupervised method.

Here, the CD-GMMs are built on an fMLLR transformation on top of LDA-MLLT features by estimating a transform for each speaker. The same training process in the preceding step is then applied on the 40-dimensions of the fMLLR features, where the number of leaf nodes and Gaussians are identical.

DNN-HMM training

Step 5: Frame-level DNN-HMM training

This step initialises the DNN-HMM model using RBM pre-training and optimises the model using backpropagation based on frame level cross-entropy. DNN-HMMs are trained on the fMLLR features, where the input layer is a splicing of fMLLR features with $\pm N$ context frames and the output layer refers to the state label PDF generated from the SAT system.

In the pre-training process, the model learns a distribution of the features without any labels. The RBM learning rate is 0.4 and the L2 penalty (a regularisation term also known as weight decay) is 0.0002. The first layer is the Gaussian-Bernoulli RBM that is used for modelling the distribution of input features. The later layers are trained on Bernoulli-Bernoulli RBM.

In the optimisation process, the parameters are updated using the mini-batch SGD method and the backpropagation algorithm. The learning rate (α in (4.19)) is 0.008 which is the optimal value evaluated among eight values (0.1, 0.01, 0.008, 0.006, 0.004, 0.002, 0.001, 0.0001) (Chapter 6.5.2). We use a halving factor to decrease the learning rate while training. The weights are updated using mini-batches of size 256 frames (mb in (4.19)). Note that we found no improvement on a smaller mini-batches size. We train the model with 50 epochs and monitor the training loss and validation loss. We apply the early stop criterion when the improvement of the validation loss is less than 0.001.

Step 6: DNN-HMM sequence discriminative training

The final step optimises the DNN-HMM parameters using the sMBR method. The sMBR method takes the inter-frame sequence into consideration and minimises state-level errors rather than the frame-level cross-entropy. We train the DNN-HMM model with 10 iterations of sMBR training. The state-level errors are estimated by decoding training utterences with a unigram language model. Here the learning rate is 1×10^{-5}.

4.6 Measurement objectives

Word Error Rate

Word error rate (WER) is an objective measure of speech recognition systems. The WER reports the edit distance in words between the hypothesised result (the transcription) and the ground-truth (what was actually said). The edit distance is the sum of the number of insertions, deletions or substitutions of words in the transcription compared with the ground truth. These edits are computed by aligning the transcription result and the ground-truth via dynamic programming and minimising the Levenshtein distance [80].

The WER is computed as

$$\mathrm{WER} \;=\; \frac{S+D+I}{N} \times 100, \tag{4.43}$$

where S refers to the number of substitution errors, D is the number of deletion errors, I represents the number of insertion errors and N is the number of words in the ground-truth.

Word Accuracy

Word accuracy (WAcc) directly reports performance of speech recognition system. The WAcc is simply computed by

$$\mathrm{WAcc} \;=\; 100 \;-\mathrm{WER} \;=\; \frac{\mathrm{N}-\mathrm{S}-\mathrm{D}-\mathrm{I}}{\mathrm{N}} \times 100. \tag{4.44}$$

Here is an example of this computation.

REF:	INTERNAL	national	responsibility	now	A	truism	need	not	be	documented
HYP:	NEITHER	national	responsibility	now	*	truism	need	not	be	documented
Eval:	S				D					

The ground-truth of this sentence (REF) has ten words. The transcription (HYP) has two errors: a substitution error (from INTERNAL to NEITHER) and a deletion error (missing A). Therefore, the WER is 20% and WAcc is 80%, which can be computed as $\mathrm{WER} \;=\; \frac{1+1+0}{10} \times 100 \;=\; 20\%$, and $\mathrm{WAcc} \;=\; 100 \;-\; 20 \;=\; \frac{10-1-1-0}{10} \times 100 \;=\; 80\%$.

In our lipreading system, we measure the word accuracy since it is much easier to see the changes of the performance when the WER is still high. Note that the terms error rate and accuracy are used on other linguistic levels depending on the measurement unit e.g. sentence level, phoneme level, viseme level, character level, etc.

Significant test

There are standard methods to measure a statistical difference between speech recognisers. Here we introduce four: McNemar's test; the matched pair sentence segment test; the signed paired comparison test; and the Wilcoxon signed rank test. All of these methods are available in the NIST scoring toolkit (SCTK).

The McNemar's test [88] calculates a chi-square statistic to measure the significance of the difference in transcription results from two recognisers. McNemar's test is used to compare the results of two classifiers tested on the same data where the classification result is either "correct" or "incorrect". Hence it is most suitable to be used on an isolated word recognition task, but it can be used on complete sentence transcriptions if they are marked as either "correct" or "incorrect". Table 4.1 represents the cross-matched table that is used to identify the number of correct and incorrect sentences between two systems via cross comparison.

Table 4.1 A cross-matched 2 × 2 contingency table.

		System B	
		Correct	Incorrect
System A	Correct	a	b
	Incorrect	c	d

The chi-square is calculated as:

$$\chi^2 = \frac{(b-c)^2}{b+c}, \tag{4.45}$$

where a is the number of utterances if both systems predict correctly, b and c denote the number of utterances that one system predicts wrong but another system predicts right, and d is a number of the utterances if both systems get it wrong.

The matched pair sentence segment error (MAPSSWE:MP) [44] is similar to the McNemar's test except that it operates in the sentence segment error instead. This technique is more appropriate for measuring a significant difference between two classifiers in a continuous speech recognition task.

The above example shows how to identify segment errors of the hypothesis results from system A and system B. There are three segments where the first segment (I) can

	I				II			III		
REF:	INTERNAL	national	responsibility	now	A	truism	need	NOT	BE	documented
HYP A:	NEITHER	national	responsibility	now	*	truism	need	not	be	documented
HYP B:	internal	national	responsibility	now	*	truism	need	NO	*	documented

be counted as c, the second segment (II) can be counted as d, and the third segment (III) can be counted as b. We can compute the chi-square (χ^2) via (4.45).

The signed paired comparison (SP) observes the differences between results in each pair of a speaker from two recognisers. The SP test computes the difference of WER of each test speaker and records only the plus sign ($+$) and the minus sign ($-$) of each pair. The critical value and P-value are defined via the sign test table using two parameters: the total number of plus signs (r) and the total number of test speakers (n).

The Wilcoxon signed-rank test (WI) [155] also operates on the WER of a speaker pair to identify the significant difference between two systems. It is similar to the SP test. The WI test computes the differences of the WERs of each test speaker, except that it keeps both the sign and the values of the difference between each pair. The rank is computed on the absolute value of the paired differences. The critical value is defined via the signed-rank table for paired differences. The WI method needs two parameters: the total number of test speakers (n) and the summation of the smallest rank between the plus and the minus sign (W_{stat}). The null hypothesis will be rejected if the W_{stat} is less than the critical value (W_{crit}).

4.7 Visualising data via t-Distributed Stochastic Neighbor Embedding (t-SNE)

The t-Distributed Stochastic Neighbour Embedding (t-SNE) visualisation technique was introduced in 2008 by Van der Maaten and Hinton [150]. T-SNE is a tool for visualising high-dimensional data. It reduces high-dimensional data into a 2D or a 3D space that retains the original information and illustrates the structure existed in the data. It groups similar data points and estimates distances between dissimilar ones. This can be done by computing the similarity matrix of data points which are converted into a joint probability then minimised via the Kullback-Leibler divergence [76] between the joint probabilities of low-dimensional (Q) and the original high-dimensional data (P). The similarity matrix of the high-dimensional data is computed from a Gaussian distribution. Conversely, in the low-dimensional space, the similarity matrix is computed from a t-distribution instead.

As presented in [150], the t-SNE minimises the Kullback-Leibler divergence between P and Q using gradient descent. The cost function is defined as:

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}, \tag{4.46}$$

where p_{ij} refers to the pairwise similarities of the high-dimensional space, and q_{ij} is the pairwise similarity of the low-dimensional map.

The p_{ij} are defined as

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}, \tag{4.47}$$

where

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2)/2\sigma_i^2}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2)/2\sigma_i^2}, \tag{4.48}$$

where $p_{j|i}$ is the probability to illustrate that the data points x_i and x_j are neighbors.

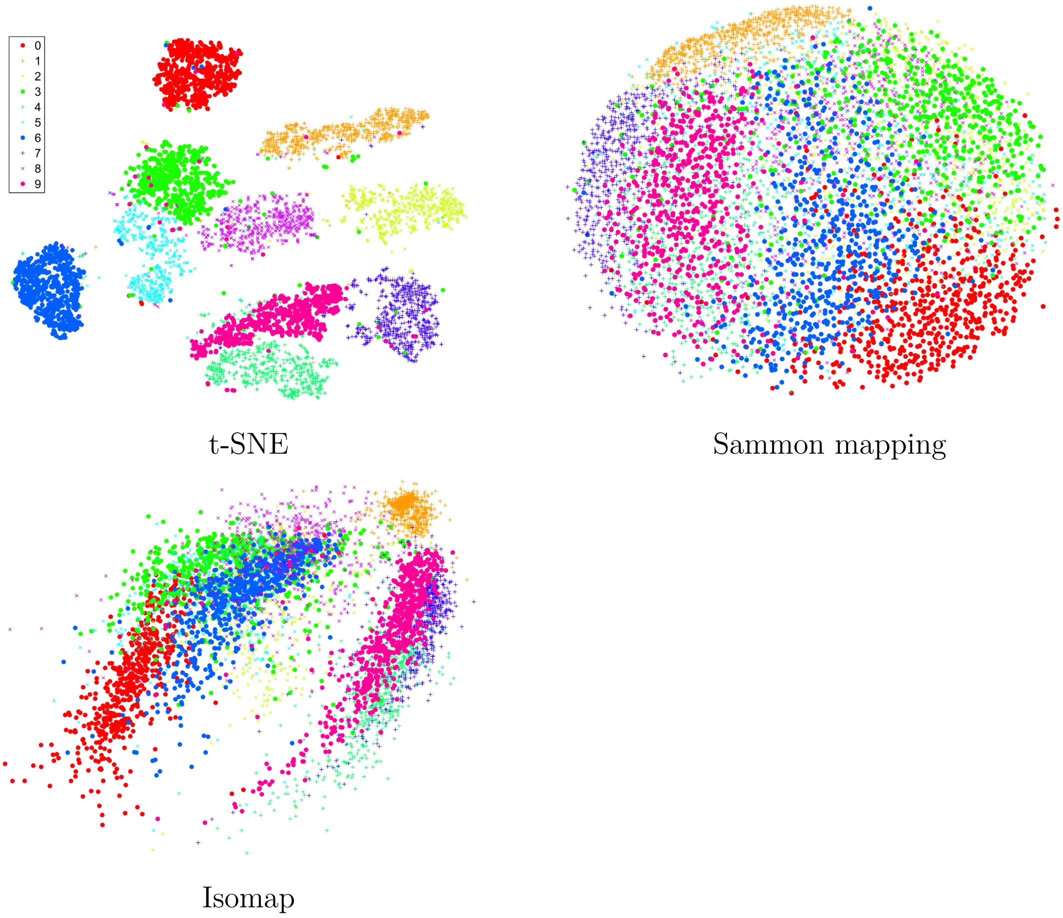
The q_{ij} are defined as

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|y_k - y_l\|^2)^{-1}}, \tag{4.49}$$

where y refers to the data point in the low-dimensional map, and i, j, k, l are indexes of the data points.

We use the t-SNE method to visualise features (in Section 7.3.1). We prepare an M by D matrix, where M is the number of samples and D is the dimensionality of each sample. We can also provide a class label vector in the form of an M by one vector. To use t-SNE, we simply put the input and the label matrix to the tsne() function, which is available in MATLAB. The algorithm will start reducing the dimensionality and finally plots the transformed data points in a 2-D space. Note that each plot of t-SNE using the same data will appear different due to random initialisation of the parameters. T-SNE also reduces high dimensional data into the initial dimension (30 dimensions by default) using PCA. (Further information of t-SNE user guide can be found in https://lvdmaaten.github.io/tsne/User_guide.pdf)

Compared to other nonlinear dimensionality reduction techniques such as Sammon's mapping [128] and Isomap [147], t-SNE provides superior results as shown in Figure 4.26. In [150], they summarised a weakness of Sammon mapping regarding its cost function as it specifies a high value to model the small distance between data points that are very close. Whereas, Isomap concerns only with modelling large distances rather than small ones.



t-SNE

Sammon mapping

Isomap

Fig. 4.26 A comparison of data visualisations on MNIST dataset (6000 handwritten digits) with three different mapping methods (a) t-SNE, (b) Sammon mapping, and (c) Isomap. Each colour is a different digit class. These visualisations are from Van der Maaten and Hinton [150]

Chapter 5

Lipreading and audiovisual speech recognition in small vocabulary tasks

This chapter concerns the connection between the acoustic speech signal and the visual. We are interested in the correlation that might exist between the two modalities. Our approach to explore this question is to build novel machine learning systems that allow us to exploit any common information. Ultimately we hope to answer the extent to which the visual signal can be useful in multispeaker noisy environments.

The contributing publication of this chapter is:

- Thangthai, K., Harvey, R. W., Cox, S. J., Theobald, B. J., **Improving lipreading performance for robust audiovisual speech recognition using DNNs**. – In *Proceedings of The 1st Joint Conference on Facial Analysis, Animation and Auditory-Visual Speech Processing (FAAVSP)*, Vienna, Austria, pp 127–131, 2015.

5.1 An overview of audiovisual speech recognition

Audiovisual speech recognition, which is illustrated in Figure 5.1, comprises three main processing blocks that are audio feature extraction, visual front-end processing and the audiovisual fusion. Compared to conventional audio-only ASR, this bimodal ASR system has the additional challenge of finding a useful signal in the visual speech and in the effective integration of the two modalities.

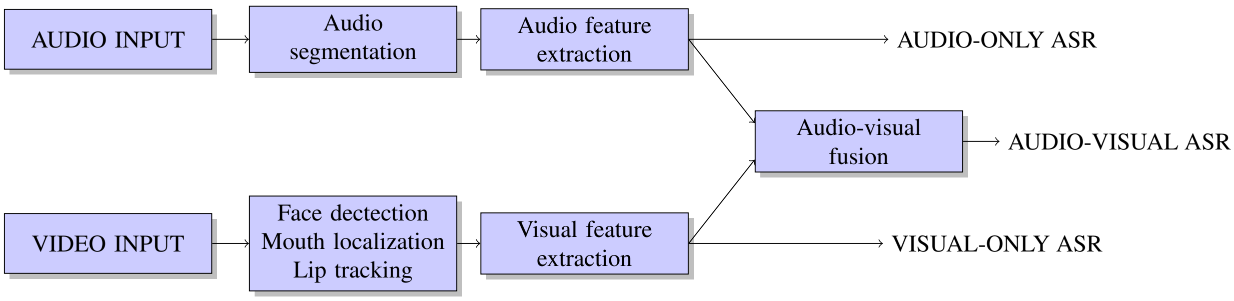


Fig. 5.1 Overview of audiovisual speech recognition system adapted from [115]

Audiovisual integration and decoding

Audio and visual speech information can be integrated using two different types of fusion techniques: feature and decision fusion.

Feature fusion techniques

Feature fusion, so-called early integration (EI) technique, can be as simple as concatenating the features from both modalities and then training the speech recogniser on those feature vectors. We can then apply a state-of-the-art model in ASR such as an HMM to train the AVSR system directly. More advanced methods in feature fusion employ discriminant feature extraction to improve the representation of the original features for both modalities by using the LDA and MLLT transform feature techniques.

The early-integration audiovisual speech recognition in this work achieves feature fusion by concatenating the audio and visual features together. The system used in this work is implemented via the Kaldi toolkit [116].

Our system is illustrated in Figure 5.2. First, raw features need to be prepared and converted into a format suitable for reading and writing in Kaldi. The acoustic features are 39-dimensional MFCCs with energy, delta and delta-delta coefficients appended. The visual features are 23-dimensional AAMs features; hence the fused audiovisual features are 62-dimensional vectors formed by concatenating the audio and the visual features.

To train the DNN model, the precise time alignments of training data obtained from the GMM system are needed. Therefore, we initialise and train the monophone and triphone models based on the GMM technique. The DNN model can be trained on top of the LDA-MLLT features as well as on the concatenated audiovisual features. To construct LDA-MLLT features, supervectors consisting of a central frame and the three

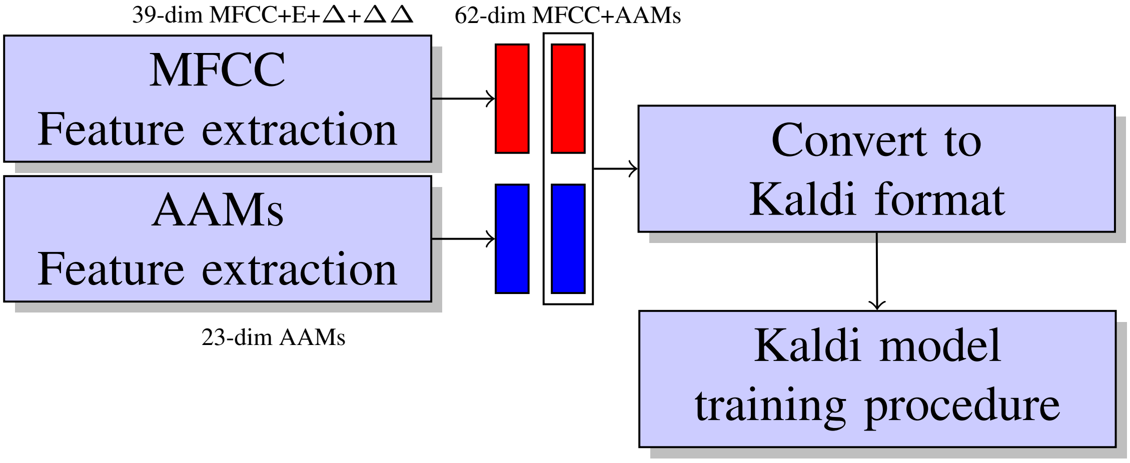


Fig. 5.2 Early integration based on feature fusion technique.

frames before and after it are formed by concatenating these frames. LDA and MLLT are applied to these supervectors and the top 40 components are used to represent the frame.

Our context-dependent (CD) DNN-HMMs (referred to as the CD-DNN model) are trained based on the tanh recipe in Kaldi. We train the neural network using plain 'vanilla' SGD for four hidden layers, using the hyperbolic tangent activation function. The input feature vectors are the original features (audio, visual or audiovisual) or the 40-dimensional LDA-MLLT features. We set the learning rate to 0.004 (the default setting), and the weights are updated using a mini-batch size of 64 frames (related to α and mb in equation (4.19) respectively). The alignments for the training data are generated from the standard CD-GMM system.

Decision fusion techniques

While feature fusion is more focused on the improvement of audiovisual speech features, decision fusion tends to be applied at the classification stage. The most popular technique uses a multi-stream HMM to construct a separate model for each modality. An alternative method is to train a separate HMM model for audio and video, then combine the result into a single recogniser. These fusion techniques are usually called intermediate- or late-integration (LI) depending on the decision level. Additionally, decision fusion is also widely explored via some other model structures such as the coupled HMM (CHMM) which composes the HMM states rather than simply training a multi-stream system.

Another technique that has been investigated in this work is the late-integration (LI) using the recogniser output voting error reduction (ROVER) technique proposed by [39].

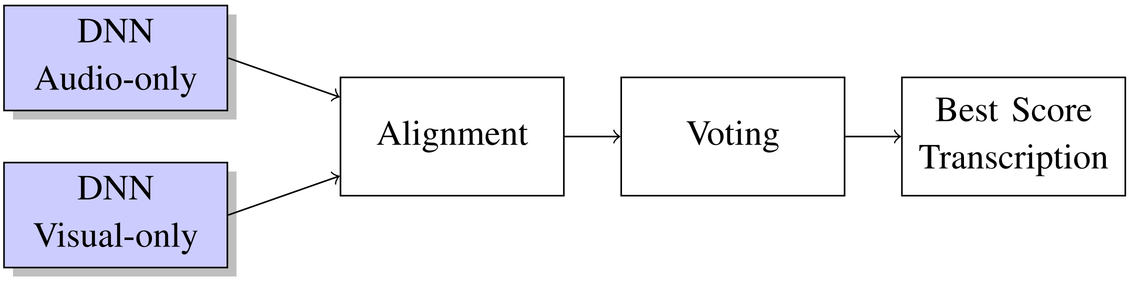


Fig. 5.3 Late-integration using ROVER technique.

The results from the DNN-HMM systems of the individual audio and visual recognisers were used as input to the ROVER system which was meant to find the best transcription from both results. The ROVER technique then tries to reduce the error that occurred in the results by aligning all those results together and selecting the word that has highest confidence score or the mostly occurred word.

5.2 Experiments and Results

5.2.1 Experimental setting

Lexicon model and Language model

The British English Pronunciation (BEEP) pronunciation dictionary was used as the lexicon. The 5000 sentences in the original RM corpus [120] that were held-out from RM-3000 were used to train a trigram language model using the SRILM toolkit [137]. The total size of language model training data is 42708 words of which 989 are unique, and the language model perplexity is relatively low (mean over folds is 13.73).

Standard feature setting

We extract the 39 dimensional MFCC+E+Δ+$\Delta\Delta$ from 16-bit wav files sampled at 16kHz (mono) with a frame size 25 ms and a 10 ms step giving 100 feature vectors per second. For the visual features, the AAM features, which contain the inner and outer lip shapes along with the appearance, were extracted from the video frames. The total vector size of AAM is 23 dimensions where the first 12 dimensions are the shape, and the rest are the appearance information. To be comparable to the frame rate of the audio feature, the AAMs were post-processed by upsampling from 25Hz to 100Hz. For audiovisual features, the audio and visual features are concatenated into a single feature vector.

K-fold cross-validation evaluation

We evaluate the performance via *k*-fold cross-validation and report the word accuracy from the mean over *k*-fold with ±1 standard error. In the *k*-fold cross-validation, the parameter estimation process has to be done *k* times repeatedly using different train-test splits. We partition the data into *k* non-overlapping subsets with equal samples, where each subset called a fold. We then evaluate one subset at a time. The rest of the subsets are used as a training set. This method ensures that all observations will be used for training and once for evaluation.

Baseline system

Most experiments on speech reading systems have used the HTK toolkit [161] for training and decoding. Therefore, the baseline system, which was trained on phoneme units, was based on HMMs using the standard HTK training and decoding procedures [151]. Both context-independent (CI) and context-dependent (CD) HMMs, which represent the mono-phones and triphones respectively were used to observe the difference in the recognition results. In addition, two different Viterbi decoders, including HVite and HDecode, were used as the recognisers for the baseline system.

5.2.2 Speaker-dependent results

The first preliminary experiments were conducted based on a speaker-dependent setting on the RM-3000 corpus which has the larger available training data to train the network, in order to avoid the effect and errors of too little data. The results are reported the mean of ten-fold cross-validation with ±1 standard error.

Visual-only speech recognition system

To evaluate the performance of visual-only speech recognition, we compare the word accuracy of the visual only recogniser with CI-GMM and CD-GMM systems with our

CD-DNN. In addition, the baseline CI/CD-GMM results are compared with the results from CI/CD-GMM on Kaldi to observe the performance of the Viterbi and WFST decoder. Furthermore, the result of the DNN on simple AAM features and LDA-MLLT features was considered.

Table 5.1 Word accuracy of visual only speech recognition system

Decoder	**Model**	**Feature**	**%Word accuracy**
HTK	CI-GMM	AAM	33.32 ±0.3
	CD-GMM	AAM	47.48 ±0.3
Kaldi (WFST)	CI-GMM	AAM	37.76 ±0.8
	CD-GMM	AAM	49.19 ±0.8
	CD-DNN	AAM	77.49 ±0.3
	CD-DNN	LDA-MLLT	84.67 ±0.3

Table 5.1 shows the performance of the visual only speech recognition systems by comparing the baseline system and the DNN-based system. It shows that the use of DNNs leads to very large improvement over conventional GMM systems, and LDA-MLLT features further improve performance to a word accuracy of 84.67%.

Audio-only speech recognition system

To investigate and compare the noise robustness of the audiovisual recognition system, the audio only recogniser was set up as a baseline system.

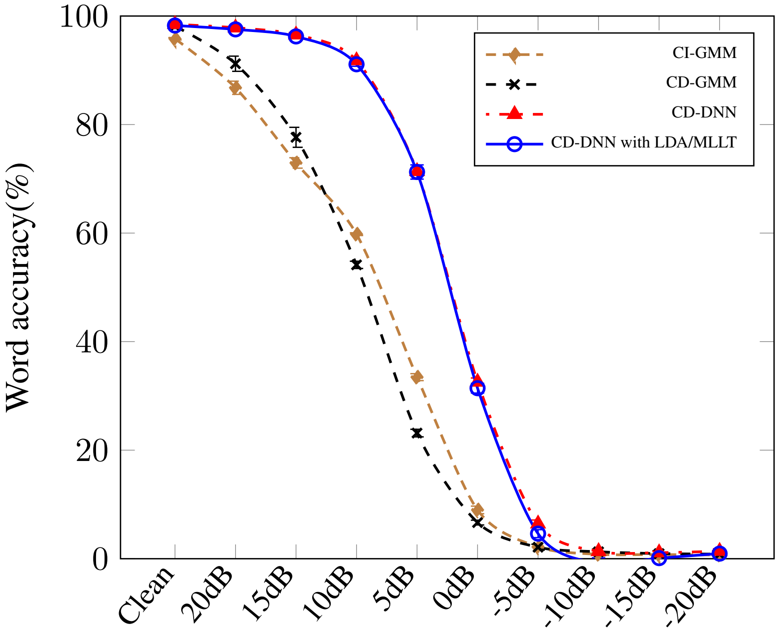


Fig. 5.4 The result of the audio-only speech recognition system in ten conditions including clean and nine babble noise levels.

The acoustic model was trained on clean speech and then evaluated using test data at nine different SNR levels of babble noise. The babble noise refers to a background noise that contains human speech sounds. The result, shown in Figure 5.4, is that the DNN is more robust to noise than the GMM. The DNN is able to achieve 90% word accuracy at 10dB, while the GMM accuracy drops to nearly 60% at the same SNR level. However, in clean conditions, the recognition result of CI and CD-GMM is relatively high at 95.81% and 98.13% respectively. The DNN is slightly better at 98.45% for raw features and 98.25% for LDA-MLLT features, which is not a significant improvement ($p = 0.064053$ by the MP test). The benefit of DNNs becomes apparent as noise is introduced.

Early-integration audiovisual speech recognition

Results for the audiovisual recognition (Figure 5.5) show that the DNN systems have an effective gain of about 12 dB over the GMM systems. Interestingly, for audiovisual CI/CD-GMM models in clean conditions, we observe a slight drop when compared to the audio-only recogniser, but the DNNs recover the accuracy to levels comparable with the audio-only system.

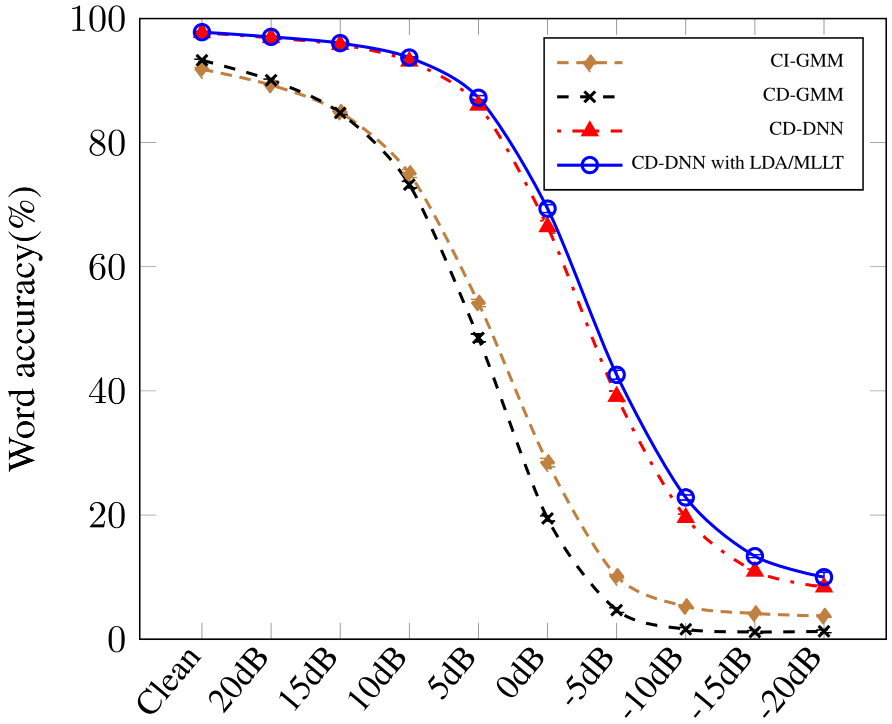


Fig. 5.5 The result of the audiovisual speech recognition system.

Figure 5.6 illustrates the CD-DNN result of three recognition systems: Visual-only, audio-only and audiovisual, and also compares raw features vs. LDA-MLLT features.

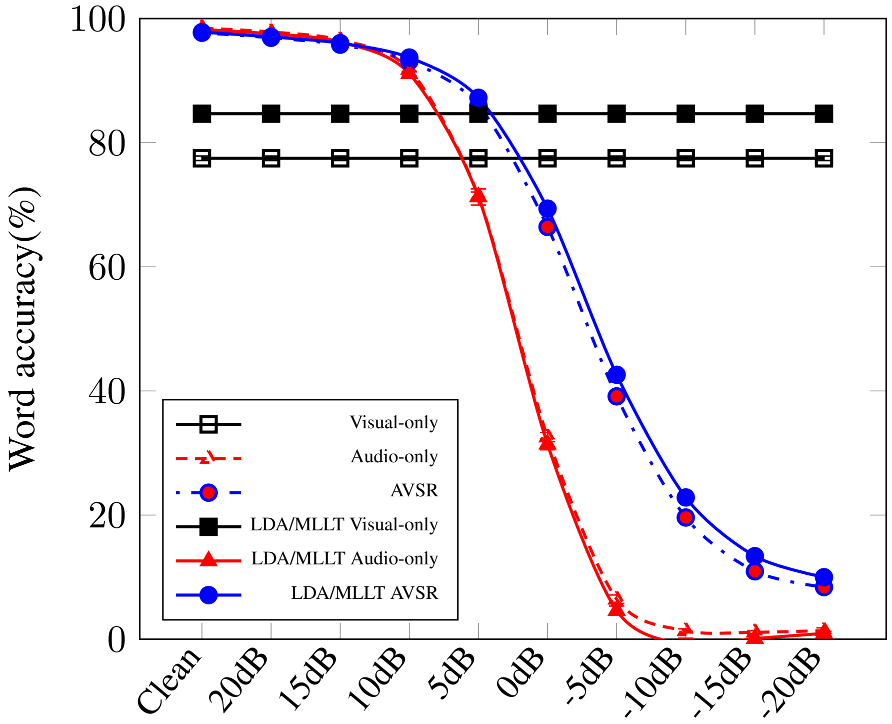


Fig. 5.6 Comparing the result of visual only, audio only and audiovisual speech recognition system using the DNN model.

As shown in Figure 5.6, there are two points that require discussion: the effect of LDA-MLLT features and the performance of early integration audiovisual speech recognition.

Firstly, the LDA-MLLT features significantly improve recognition accuracy of visual-only speech ($p < 0.01$ by the MP test) and provide a modest improvement in AV accuracy in high noise conditions. However, they do not show any significant improvement for an audio-only system ($p = 0.893266$ by the MP test). This is because the MFCC feature together with the velocity and acceleration coefficients have sufficient discriminative information for the DNN.

Secondly, we have found that it is only beneficial to add visual information in an early integration style when the SNR is at (or above) 5dB. After 5dB SNR there is no effective gain by incorporating the visual features, and the audiovisual system degrades much like the audio-only system.

Late-integration AVSR using the ROVER technique

The proposed use of the late-integration approach by using the ROVER technique is mainly focused on training individual models for each modality, then recognising each separately, before fusing those results together based on their confidence scores. Indeed, results from lipreading systems and audio-only system were aligned, then the best word confidence score was selected from those results, which finally become the final result of the system.

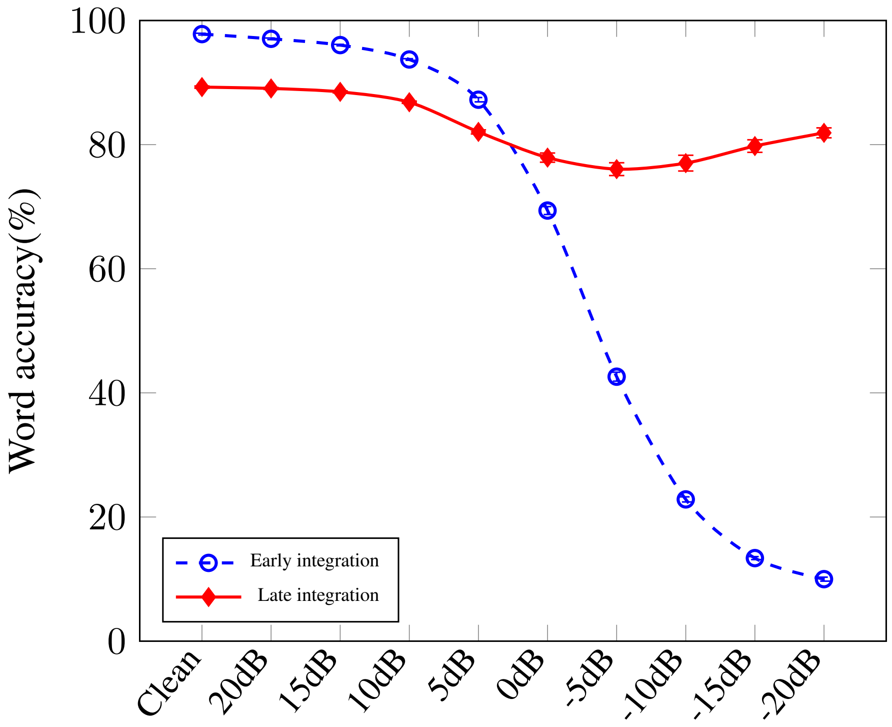


Fig. 5.7 Early-integration (EI) vs. Late-integration (LI) Audiovisual Speech Recognition

In contrast to the early-integration approach, the result of late integration in Figure 5.7 is much closer to the visual-only system. These seem to show that this approach is not affected by acoustic noise, especially when the noise is much louder than the actual speech. However, the performance of LI-AVSR was significantly lower than the EI-AVSR in the clean condition and higher SNRs ($p < 0.01$ by the MP test).

Matched vs. unmatched condition recognisers

It has long been known that visual features can improve the accuracy of a speech recogniser in noise. By using deep learning with a visual only recogniser, we are able to achieve almost 85% word accuracy.

The assumption of mismatched acoustic conditions (i.e. trained in clean conditions and tested in noisy conditions) has a big effect on the recognition system so we setup another decoder, which was trained in matched conditions (i.e. recognisers were trained with acoustic features trained to a particular SNR). This result is shown in Figure 5.8, which shows that the performance of a matched condition audiovisual DNN-based audiovisual recognition system is significantly better than the audio-only system at lower SNR ($p < 0.01$ by the MP test), and is close to the visual-only performance.

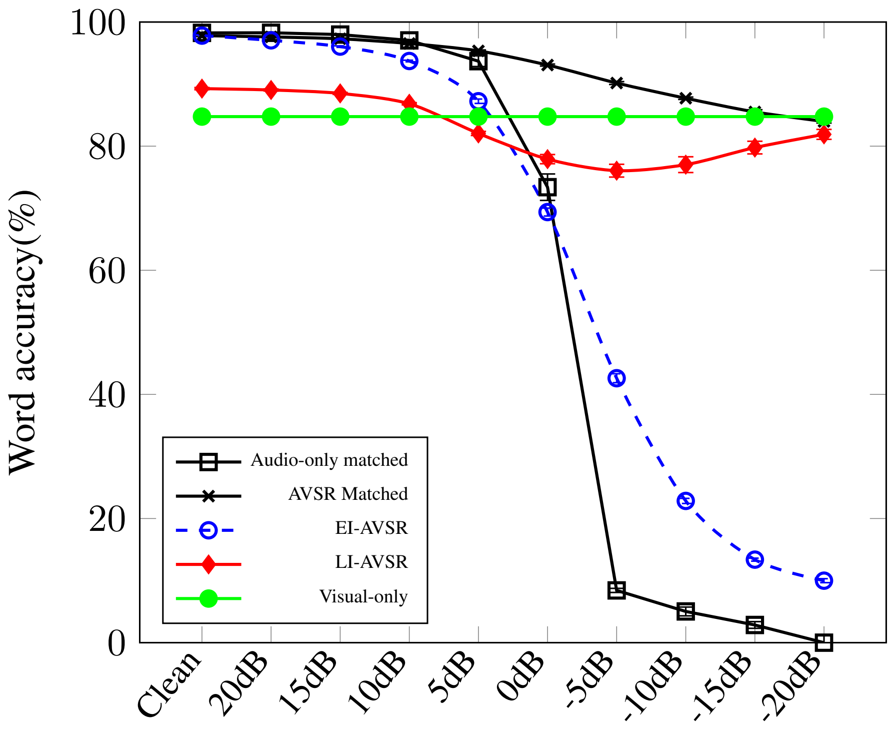


Fig. 5.8 Comparing the results under matched acoustic conditions for the CD-DNN with LDA-MLLT.

5.2.3 Speaker-independent results

As shown in the previous section, the DNN-HMM has successfully improved the performance of the all recognition systems in a speaker-dependent setting. This section will explore the performance of hybrid DNN-HMMs in a speaker-independent setting on RMAV corpus described in Chapter 2.4.2. This is a more challenging problem on a smaller dataset (per talker). We use audio and video of 12 subjects (seven male, five female). Each subject reads around 200 sentences that were selected from the Resource Management corpus [120]. The the dataset contains 2358 utterances (about three hours), and the vocabulary size is about 1000 words. There are 42 utterances short of 200 per speaker because eight speakers missed a couple of utterances. This corpus provides 16-bit 16 kHz audio data along with AAMs [85] features as a representation of visual features that are tracked by a speaker-dependent AAM tracker.

The RMAV data was in twelve-fold cross-validation with one speaker per fold. However, we only used the four speakers (four folds) for testing that correctly uttered all 200 phrases. Thus our word accuracy are the mean of four speakers (one female and three male) with ±1 standard error. Note this is speaker-independent since the same speaker is never in the test and training sets.

Visual-only

Table 5.2 The result of the visual only speech recognition system on the speaker-independent setting.

Decoder	**Model**	**Feature**	**Dimension**	**%Word accuracy**
Kaldi (WFST)	CI-GMM-HMM	AAM	23	9.99 ±2.0
	CD-GMM-HMM	AAM	23	10.84 ±0.7
	CD-GMM-HMM	LDA-MLLT	40	14.09 ±0.3
	CD-GMM-HMM+SAT	fMLLR	40	39.99 ±2.6
	CD-DNN-HMM	LDA-MLLT	40	37.79 ±2.6
	CD-DNN-HMM+SAT	fMLLR	40	53.26 ±2.0

As shown in the Table 5.2, DNN systems tend to give better performance than GMM systems. However, the best gain obviously comes from SAT on the fMLLR features. This achieved 25.90% absolute gain on word accuracy (from 14.09% to be 39.99%). Indeed, DNNs which were built on top of SAT and fMLLR can provide 53.26% word accuracy. This suggests that with speaker-independent recognisers, SAT can help overcome the difficulty in speaker variation. However, the results of speaker-independent lipreading are still a lot worse than that the speaker-dependent setting. Therefore, speaker-independent lipreading is still a challenge.

Audiovisual speech recognition

In Table 5.2, the best word accuracy in speaker-independent lipreading is about 53% where the single-speaker lipreading performance can reach almost 85% accuracy. However, it can be seen from Figure 5.9 that incorporating visual speech features still has a benefit in terms of improving the robustness of the recognition system even in this challenging task. Therefore, improving the recognition results in the speaker-independent setting over these baseline results is going to be our next research focus.

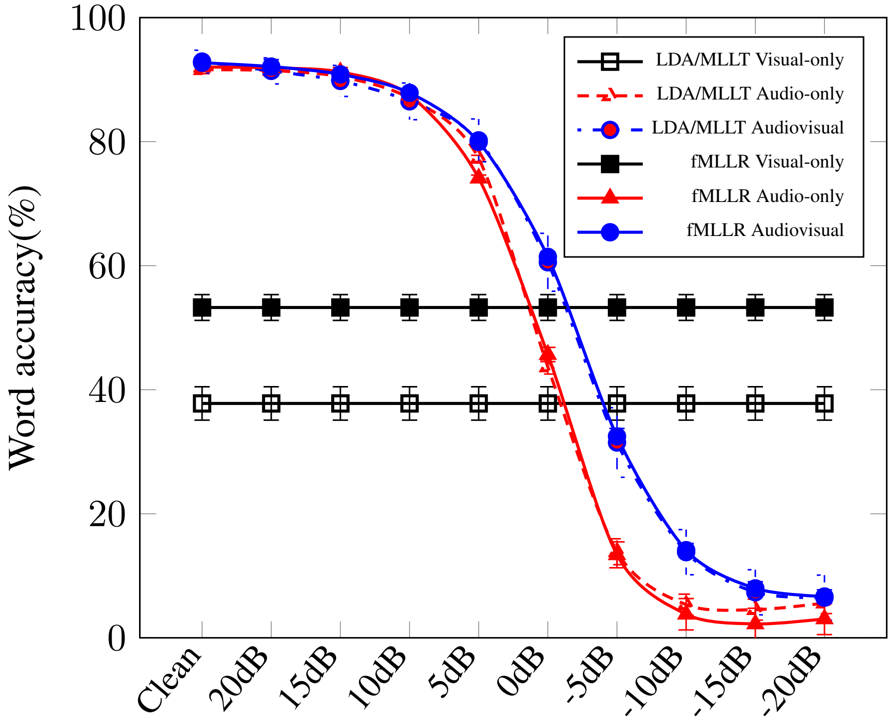


Fig. 5.9 Comparing the result of visual only, audio only and audiovisual speech recognition system of the speaker-independent setting.

5.3 Discussion

This Chapter has explored the use of DNNs in visual and audiovisual speech recognition. The experiments are mainly based on two different settings: speaker-dependent experiments and speaker-independent experiments.

In the speaker-dependent visual speech (lip-reading) experiment, DNNs gave 85% word accuracy, a huge improvement on the baseline HMM performance of 33%. Moreover, we found that DNNs improved the robustness of audio-only and audiovisual recognition tasks by approximately 10 and 12dB respectively. In addition, we found that audiovisual recognisers degraded in a similar fashion to audio-only recognisers in high-noise environments by the early-integration approach, where the late-integration seems to show more benefit when the acoustic signal is corrupted by acoustic noise. Finally, it is interesting to see that the performance is significantly improved for matched-condition recognisers, where the performance of the audiovisual system closely followed the visual-only system.

For the speaker-independent setting, DNNs on top of fMLLR features trained by the SAT technique gives the best performance of 53.26% word accuracy. Incorporating visual information into the recogniser yields better results than pure audio speech recognition.

The experimental results obviously demonstrate that the DNN techniques even in standard settings can beat the conventional GMM-HMM speech recogniser for both unimodal and bimodal speech recognition systems. However, a major problem with the system is due to the size of the data to train DNNs which is quite small. In the next Chapter, we draw our attention to understand and to improve lipreading on TCD-TIMIT (described in Chapter 2.4.3), a large vocabulary multispeaker dataset which is a more realistic task.

Chapter 6

Lipreading for large vocabulary continuous speech recognition task

The previous Chapter illustrates the connection between acoustic and visual speech signals where the improvement of visual speech can directly enhance the performance of speech recognisers. There have been some promising results in computer lipreading especially in the single speaker system where we got 85% word accuracy. However the word accuracy dropped significantly in the 12-speaker scenario.

This Chapter concerns making a viable computer lipreading for a more realistic task. We develop computer lipreading on the large vocabulary continuous speech recognition task using the TCD-TIMIT corpus. The TCD-TIMIT corpus has around 6000 words and seven hours of recorded audio-visual speech collected from 59 speakers. We deploy DNN-HMM models and further improve the models with sequence discriminative training. Furthermore, we explore the effect of visual speech units between phonemes and visemes by evaluating unit recognisers and word recognisers. We compare two type of visual features: DCT and Eigenlips. We then optimise DNN training parameters: number of hidden layers, number of hidden units, and learning rate.

The contributing publications of this chapter are:

- Thangthai, K., Harvey, R., **Improving Computer Lipreading via DNN Sequence Discriminative Training Techniques**. – In *Proceedings of the Annual Conference of the International Speech Communication Association INTERSPEECH 2017*, Stockholm, Sweden, pp 3657–3661, 2017.

- Thangthai, K., Bear, H. L., Harvey, R., **Comparing phonemes and visemes with DNN-based lipreading**. – In *LRDLM Workshop on Lip-reading using Deep Learning Methods (at BMVC 2017)*, London, UK, 2017.

6.1 Dataset

We use the TCD-TIMIT [49] corpus (Section 2.4.3) containing 59 volunteer speakers. We chose this dataset because, at the time we did the experiments, it was the largest vocabulary audio-visual speech corpus available in the public domain. The WFST operates on a vocabulary of almost 6000 words from a dictionary of 160k entries. This dataset provides lists of non-overlapping utterances for training and evaluation in two scenarios: speaker-dependent (SD) and speaker-independent (SI). In the SD scenario, visual models are trained on 3752 utterances and evaluated on 1736 utterances. Whereas in the SI experiment, 3822 utterances from 39 talkers are in the training set and we evaluate on the remaining 17 talkers containing a total 1666 utterances. We report the mean over three-fold cross validations (with ±1 standard error) where we use the recommended set as the first-fold and we prepare another two-folds by retaining the similar proportion.

6.2 Visual Speech Features

6.2.1 Feature extraction

The literature provides a variety of feature extraction methods, often combined with tracking (which is essential if the head of the talker is moving). Here we focus on features that have been previously described as “bottom-up” [85] meaning that they are derived directly from the pixel data and require only a Region-Of-Interest, or ROI. Figure 6.1 illustrates a typical ROI taken from the TCD-TIMIT dataset described in Chapter 2.4.3 plus two associated feature representations: eigenlips (Chapter 4.4.1) and DCT (Chapter 4.4.1).

6.2.2 Discrete Cosine Transform (DCT)

Here we use DCT II with zigzag selection [158], which means that the first elements of the feature vector contain the low-frequency information. The resulting feature vector has 44 dimensions extracted from 64×128 grey-scale lip images, as shown in Figure

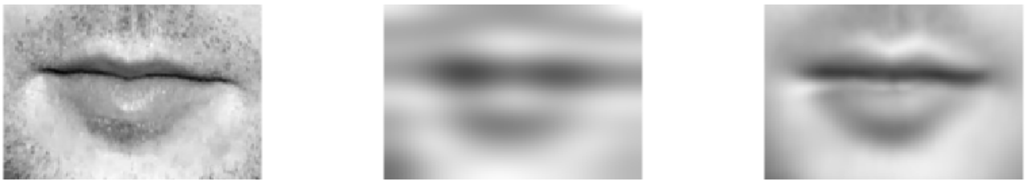


Fig. 6.1 Comparing the original ROI image (left) and its reconstruction via 44-coefficient DCT (middle) and 30-coefficient Eigenlip (right). Note that the 44-coefficient DCT is equivalent to the features of the TCD-TIMIT baseline system [49].

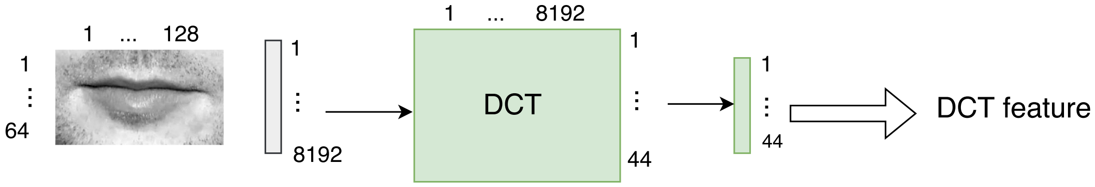


Fig. 6.2 DCT feature extraction.

6.2, which is equivalent to the DCT feature presented in the TCD-TIMIT baseline system [49].

6.2.3 Eigenlips

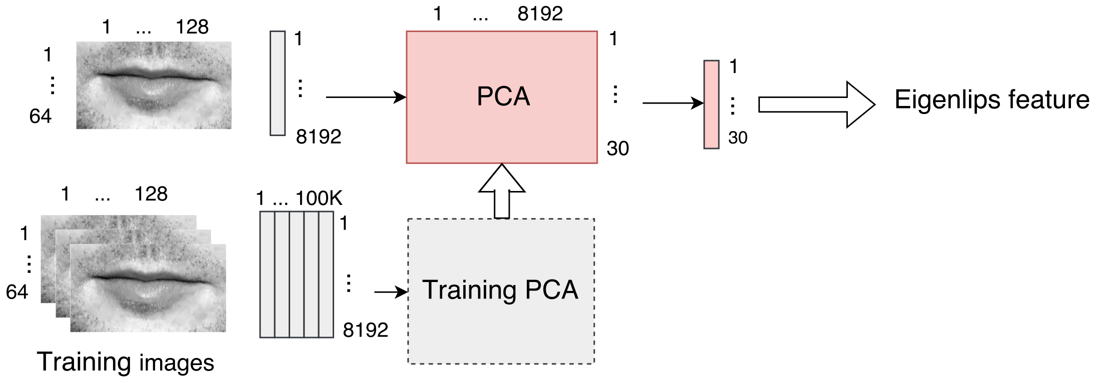


Fig. 6.3 Eigenlips feature extraction.

The Eigenlips feature is another appearance-based approach [74]. The Eigenlips features have been generated via PCA [156]. As shown in Figure 6.3, we use PCA to extract the Eigenlips features from 64 × 128 grey-scale lip images, where we retain only 30-dimensions of PCA (covering 85% of the principal component variances while still maintaining the compact size of feature). To construct the PCA, 25 ROI images of each training utterance were randomly selected to be the set of training images.

Almost about 100k images in the training set were used to perform eigen analysis. Only 30 dimensions of principal components with high variation were retained.

6.2.4 Feature transformation

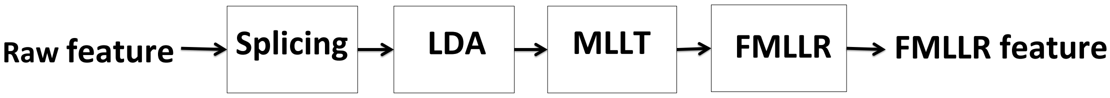


Fig. 6.4 FMLLR feature pre-processing pipeline.

The raw features are 30-dimensional Eigenlips and 44-dimensional DCT features. The 15 (±7) consecutive frames of these raw features are spliced onto the feature to add dynamic information. Second, LDA [40] and MLLT [42] are applied to reduce and map the features to a new space to minimise the within-class distance and maximise the between-class distance, where the class is the HMM-state, whilst simultaneously maximising the observation likelihood in the original feature space.

Finally, fMLLR [118],[42], also known as the feature-space speaker adaptation technique, is employed to normalise the variation within a speaker. These new 40-dimensional fMLLR features are used as inputs to the subsequent machine learning. The use of LDA is quite commonplace in lipreading and is derived in the HiLDA framework [113]. MLLT and fMLLR are commonplace in acoustic speech recognition but have only recently been applied to visual speech recognition [4] albeit on small datasets.

6.3 Pronunciation dictionary

A pronunciation dictionary or a lexicon is essential to a DNN-HMM based lipreading system. This work uses the Irish accent pronunciation dictionary provided in the TCD-TIMIT corpus that contains 156,516 word entries adapted from the CMU dictionary.

However, there is debate if phoneme or viseme units are the most effective for a lipreading system. Some studies use phoneme units even though phonemes describe unique short sounds rather than lip shapes; other studies tried to improve lipreading accuracy by focusing on visemes with varying results.

6.3.1 Analysis of the pronunciation dictionary

Reducing the set of speech units, such as reducing a set of phonemes to a set of visemes, reduces the discriminative power of the classification model whilst increasing the complexity of the pronunciation dictionary by increasing the volume of homophene words. This suggests that word accuracy of a viseme based system will be lower than a phoneme based system. The counter argument is that visemes might be simpler to classify (because there are fewer of them and they are meant to be better matched to the visual signal) so there is clearly a trade-off between homophenes and unit accuracy [32].

Table 3.4 (in Chapter 3.4) shows examples of the homophone and homophene words that occur in the TCD-TIMIT dictionary. Figure 6.5 describes the homophone problem in two ways. On the top words are binned according to how many homophones they have. Thus the column labelled “1 occur” is the count of all unique words, the column labelled “2 occur” is the count of words that have one other homophone and so on. It is evident the switch to visemes causes more homophones particularly large numbers of high-multiplicity homophones. This effect can also be seen in the dictionary size (bottom of Figure 6.5). Homophones cause dictionary entries to merge so the visual dictionary is smaller than the acoustic one.

6.4 Decoding lipreading

6.4.1 Visual speech model

We train GMM-HMM models and DNN-HMM models on fMLLR features (Chapter 6.2.4) by following the full pipeline training (explained in Chapter 4.5).

We construct the CD-DNN model on the hybrid DNN-HMM architecture. The CD-DNNs are trained and optimised by minimising frame-based cross-entropy between the prediction and the PDF target. The PDF refers to the tied-state context-dependent label, which is generated from the SAT system, that is aligned to every frame. The features we adopted for all DNN training are based on LDA+MLLT+fMLLR features with mean and variance normalisation.

The CD-DNNs model is trained on six hidden layers with 2048 neurons per layer (optimised in Section 6.5.2), where we use the sigmoid non-linearity function in each neuron. The input layer is the fMLLR feature with temporally spliced 11 consecutive frames. The model is initialised by a stacking of RBM with 15 iterations on a single-

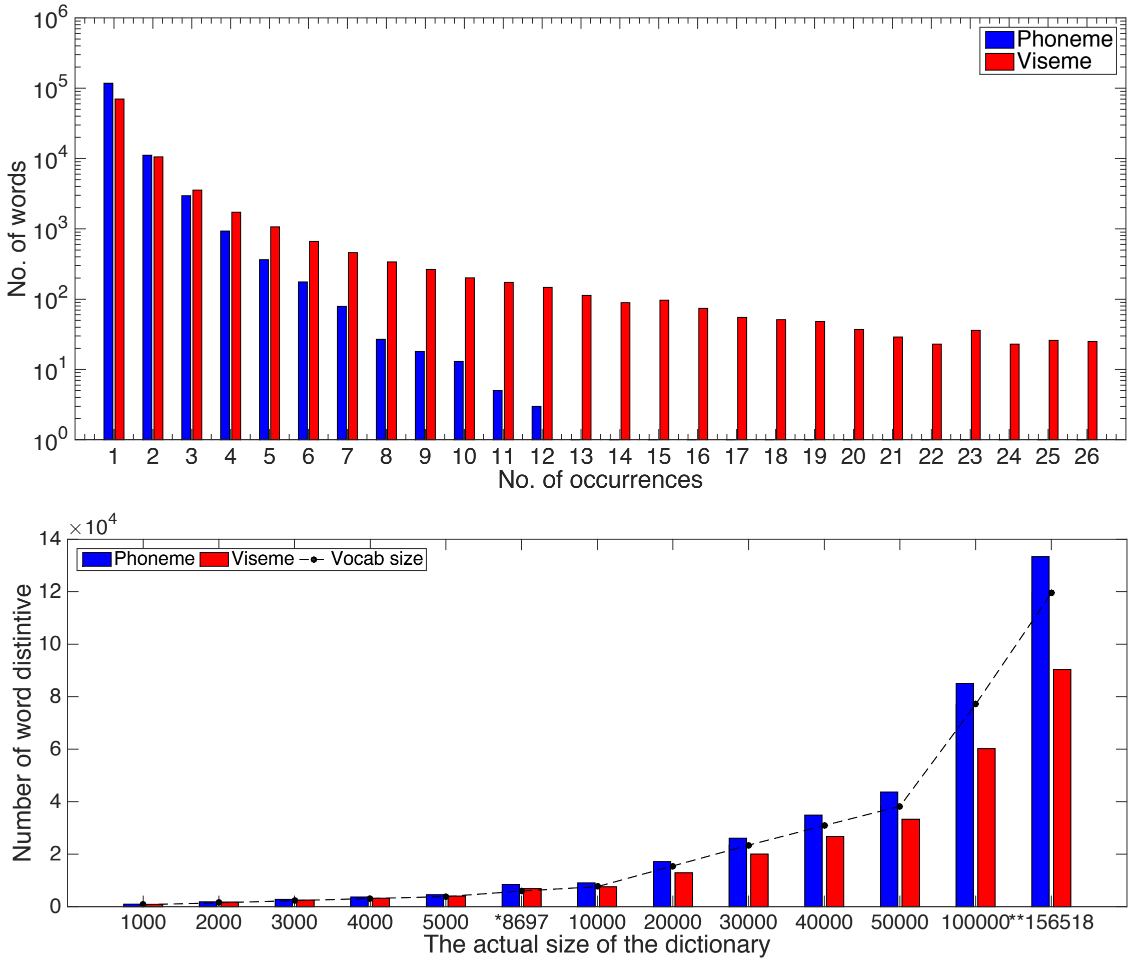


Fig. 6.5 Frequency of duplicated pronunciation in the TCD-TIMIT dictionary (top) and vocabulary size (bottom) for both phoneme and viseme units.

GPU machine. The learning rate for RBM training is 0.4 and applying L2 penalty (weight decay) at 0.0002 (default of Kaldi). The learning rate for fine-tuning has been set to 0.008 (optimised in Section 6.5.2). We use the minibatch-Stochastic Gradient Descent (SGD) for fine-tuning with minibatch size of 256. We produce a development set for tuning the network by randomly selecting 10% of the training data. Every DNN training iteration is required to have a cross-validation loss lower than the previous training iteration. If an iteration is rejected then one retries with a new stochastic gradient descent parameter. The terminating condition is that the new loss is little different from the old loss (specifically we use a difference smaller than 0.001 of the loss as a suitable terminating condition).

Language model

A language model (LM) is also essential to our system. The LM helps discriminate similar input patterns of words found in the lexicon and also reduces the search cost. For the LM, we use a statistical based n-gram model where we train a word bi-gram from the TCD-TIMIT provided text. To make it fair we use only text provided in the training set, thus we have two word bi-gram LMs; one for SD and one for SI. We know already that longer n-grams mean better performance but extending the number of word n-grams can be too restrictive and will lead to difficulty in finding an appropriate parameter for visual speech modelling. Here we evaluate our LM by computing the perplexity of SD (35.16) and SI (33.10) evaluation sets against their LM with no out-of-vocabulary words found in both cases.

6.5 Experiments and results

To clarify, in these experiments we report the lipreading accuracy from the mean over three-fold cross-validation. Also, the model parameters are fine-tuned by observing the best result from the three-fold cross-validation. We want to point out that our method may lead to the over-fitting problem on model selection indicated by Cawley and Talbot [22]. To avoid the overfitting issue, Cawley and Talbot [22] suggest that the nested cross-validation [138] is a better method for model selection.

6.5.1 Effect of visual speech unit

We compare the performance of a lipreading system by modelling visual speech using either 13 viseme or 38 phoneme units. We report the accuracy of our system at both word and unit levels. The evaluation task is large vocabulary continuous speech using the TCD-TIMIT corpus. We complete our visual speech modelling via hybrid DNN-HMMs and our visual speech decoder is a WFST. We use DCT and Eigenlips as a representation of the mouth ROI image.

As lipreading transitions from GMM/HMM-based technology to systems based on DNNs there is merit in re-examining the old assumption that phoneme-based recognition outperforms recognition with viseme-based systems. Also, given the greater modelling power of DNNs, there is value in considering a range of hand-crafted features such as DCT [2] and Eigenlips [17].

In the SAT system, the CD-GMMs are built on an fMLLR transformation on top of LDA-MLLT features by estimating a transform for each speaker. The same training process in the preceding step is then applied on the 40-dimensions of fMLLR features, where the number of leaf nodes and Gaussians are identical.

Viseme-based lipreading

One fundamental measure of the performance of an automatic lipreading system is viseme accuracy. Since the viseme recogniser requires no dictionary or language model, it is quicker to build and optimise. The TCD-TIMIT release includes a baseline viseme accuracy for both speaker dependent and speaker independent settings using the Neti visemes [95] used here. The best viseme accuracy of recognising 12 viseme units reported on TCD-TIMIT is 34.77% in speaker independent tests and 34.54% on speaker dependent tests. The context independent viseme models (referred to as mono-visemes in the paper) were trained on 44-coefficient DCT features with 4-state HMMs and 20 Gaussian mixtures per state.

Table 6.1 lists the accuracies achieved with our viseme based lipreading system. In comparison to the viseme accuracies benchmarked with the TCD-TIMIT corpus, our best SD viseme accuracy is 46.57% with Eigenlips, compared to 34.54%, an improvement of 12.03%. Our best SI viseme accuracy is 45.41% which improves on the benchmark 34.77% by 10.64%, again with the Eigenlips features.

Table 6.1 Viseme-based lipreading accuracy (%).

Model	Feature	Viseme accuracy (%)		Word accuracy (%)	
		SD	SI	SD	SI
CD-GMM + SAT	DCT	41.30 ±0.3	40.70 ±0.1	16.25 ±0.3	14.89 ±0.6
CD-DNN		44.21 ±0.6	42.00 ±1.2	22.43 ±1.1	20.95 ±0.8
CD-GMM + SAT	Eigenlips	45.85 ±0.0	45.16 ±0.1	18.63 ±0.8	17.64 ±0.6
CD-DNN		46.57 ±0.5	45.41 ±0.7	30.65 ±1.3	28.33 ±0.8

Phoneme-based lipreading

Table 6.2 shows the word and phoneme accuracies achieved with our phoneme-based lipreading system. This system achieved the most accurate lipreading with a word accuracy of 45.83%. It is interesting that with the phoneme recogniser, word accuracy is greater than phoneme accuracy, while in the viseme recogniser, this is vice versa.

Again, highest accuracy is achieved with Eigenlip features rather than DCT.

Table 6.2 Phoneme-based lipreading accuracy(%).

Model	Feature	Phoneme accuracy (%)		Word accuracy (%)	
		SD	SI	SD	SI
CD-GMM + SAT	DCT	28.26 ±0.3	27.41 ±0.8	27.73 ±1.0	24.61 ±1.6
CD-DNN		30.31 ±0.4	28.39 ±0.7	37.59 ±1.2	33.91 ±2.2
CD-GMM + SAT	Eigenlips	28.77 ±0.1	27.83 ±0.0	28.61 ±0.8	25.91 ±0.7
CD-DNN		32.85 ±0.4	31.24 ±0.6	**45.83** ±0.4	**41.66** ±0.6

One interesting observation apparent in Tables 6.1 and 6.2 is that the introduction of the DNN makes little difference to the unit accuracy but a bigger difference to a word accuracy for both DCT and eigenlips features.

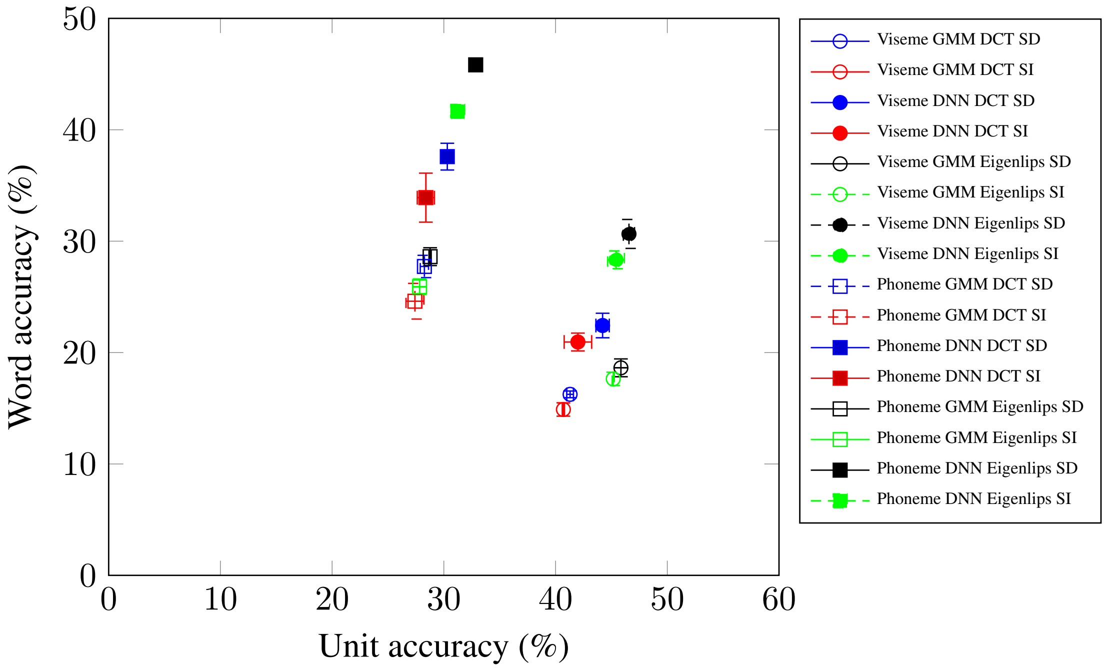


Fig. 6.6 Lipreading system performance in GMM and DNN systems.

Figure 6.6 has two clusters: one, in the bottom right, represents the viseme experiments and the other, on the upper left the phonemes. Here we are representing viseme classifiers with circles (filled represents the DNN, open the GMM) and the phonemes with squares (either filled or open depending on the classifier). The colours represent the various SI/SD or DCT/Eigenlips combinations.

The phoneme recogniser naturally obtained lower unit accuracy scores because it has three times more phoneme classes than viseme classes (13 to 38 respectively). But this does not mean that phoneme classes have less power to model a visual gesture.

This is visualised in the confusion matrices in Figure 6.7 where the colour patterns are consistent between phoneme classes (on the left of Fig 6.7 and between viseme classes on the right of Fig 6.7).

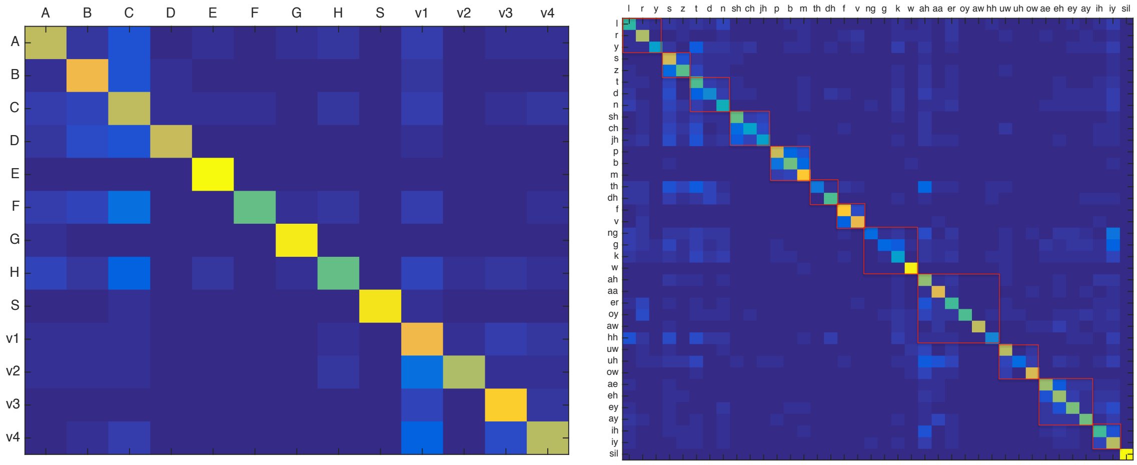


Fig. 6.7 Comparison of visemes confusion matrix (left) vs phonemes confusion matrix (right). The boxes in the phonemes confusion matrix show viseme classes.

We note that reducing the set of visual speech units also reduces the discriminant power of the classification model whilst increasing the complexity of pronunciation dictionary by increasing the volume of homophenes. This suggests that word accuracy of a viseme based system will be less likely to outperform the phoneme based system.

One observation we found is that DNN-HMM viseme recognisers can easily overfit to the training observations. This is shown in the performance disparity between SD and SI configurations. It could potentially be interesting to use visemes as an initialisation for phoneme recognition in a hierarchical training method similar to that in [11] in the future.

6.5.2 Word based DNN-HMMs

The previous experiments demonstrated that phoneme classifiers can outperform those of visemes. We have also illustrated the noticeable performance gain by changing visual representation from DCT to Eigenlips. The best word accuracy in this work is 45.83% on SD and 41.66% on SI achieved with the DNN-HMM phoneme unit recogniser trained on the Eigenlips feature.

The rest of these experiments will carry on by focusing on building the word recogniser from the phoneme-based lipreading with DNN-HMMs and the Eigenlips feature. We optimise the DNN model training parameters: hidden layer, hidden unit, initialising method, and input window size. The CD-DNNs are trained and optimised by minimising frame-based cross-entropy between the prediction and tied-state context-dependent label, which are generated from the Speaker Adaptive Training (SAT) system (Section 4.5), and then aligned into every frame. The features which we adopted for the DNN training process is based on 40-dimensional fMLLR features with mean and variance normalisation, where the fMLLR obtained via LDA-MLLT projection of 15 frames spliced of Eigenlip feature.

DNN-HMM parameter optimisation: Number of hidden layers

The first parameter to optimise is the number of DNN hidden layers. Table 6.3 presents the lipreading word accuracy for various number of hidden layers. Increasing of number of hidden layers improves lipreading accuracy. The best SD result is 46.02% obtained from eight hidden layers with 2048 units per layer. The best SI word accuracy is 41.66% obtained from six hidden layers and 2048 units per layer.

Table 6.3 DNN-HMM lipreading word accuracy with various hidden layers.

Model	Feature processing	No. hidden layer	Word accuracy (%)	
			SD	SI
DNN	fMLLR	1	43.08 ±0.9	39.27 ±0.6
		2	43.36 ±0.9	39.37 ±0.3
		3	43.54 ±1.0	39.91 ±0.5
		4	44.25 ±0.7	40.20 ±0.7
		5	44.79 ±0.7	40.98 ±0.7
		6	45.83 ±0.6	**41.66** ±0.4
		7	45.93 ±0.9	41.19 ±0.9
		8	**46.02** ±0.6	41.27 ±1.0
		9	45.82 ±0.7	41.52 ±0.6
		10	45.84 ±1.0	41.37 ±0.4

Setting the optimum number of hidden layer gains between 2 and 3% word accuracy (2.94% in SD, 2.39% in SI) from the lowest word accuracy at one hidden layer. We use six hidden layers for the rest of the experiments as it obtains the best result on the SI task, and six layers are within an error bar of eight on the SD task.

DNN-HMM parameter optimisation: Number of hidden units

Here we investigate the effects of reducing and extending the number of hidden units. We vary five different sizes: 256, 512, 1024, 2048, and 4096. Table 6.4 presents lipreading word accuracies of various numbers of hidden units. The results show that using 2048 units per layer gives the best results for both SD and SI.

Table 6.4 DNN-HMM lipreading word accuracy with various hidden units.

Model	Feature processing	No. hidden unit	Word accuracy (%)	
			SD	SI
DNN	fMLLR	256	43.29 ±0.8	39.95 ±0.8
		512	44.75 ±0.6	40.11 ±0.7
		1024	44.77 ±0.8	40.94 ±0.5
		2048	**45.83** ±0.6	**41.66** ±0.4
		4096	45.73 ±0.5	41.23 ±0.9

DNN-HMM parameter optimisation: With/without RBM pretraining

Here we investigate the effect of employing the RBM pre-training method. Also, we examine the word accuracy when using different nonlinear functions: Sigmoid and Tanh. Table 6.5 shows that the RBM pre-training method gives the best word accuracy in both scenarios. Results from RBM pre-training achieved around 1% higher compared to no pretraining although the difference is marginal. The sigmoid nonlinear function obtains better results than Tanh by around 10%.

Table 6.5 DNN-HMM lipreading word accuracy with/without RBM pre-training using sigmoid and tanh nonlinear function.

Model	Feature processing	RBM pre-training	Non-linear	Word accuracy (%)	
				SD	SI
DNN	fMLLR	Yes	Sigmoid	**45.83** ±0.6	**41.66** ±0.4
		No	Sigmoid	44.78 ±1.0	40.31 ±1.2
		No	Tanh	35.64 ±1.4	31.67 ±1.7

DNN-HMM parameter optimisation: Learning rate

Here we optimise the learning rate parameter. Table 6.6 shows the word accuracy of the SD and SI tasks using different learning rates, ranged between 0.0001 and 0.1. It seems that performance has changed only a little within the certain values of learning rate from 0.001 to 0.01. Using too large (0.1) and too small (0.0001) learning rate decrease word accuracy significantly ($p < 0.05$). The best results are obtained using a learning rate of 0.008 in both SD and SI.

Table 6.6 DNN-HMM lipreading word accuracy with various learning rates.

Model	Feature processing	Learning rate	Word accuracy (%)	
			SD	SI
DNN	fMLLR	0.1	39.77 ±3.5	29.20 ±0.4
		0.01	45.19 ±0.7	41.06 ±0.6
		0.008	**45.83** ±0.6	**41.66** ±0.4
		0.004	45.45 ±0.7	41.64 ±0.7
		0.002	45.49 ±0.9	41.57 ±0.7
		0.001	45.43 ±0.7	41.21 ±0.5
		0.0001	15.39 ±1.7	14.48 ±1.0

DNN-HMM parameter optimisation: Window size

Here we optimise the window size of the fMLLR feature. We vary the number of dimensions of the input layer by splicing of $\pm n$ consecutive frames of fMLLR features where $n = (0, ..., 7)$.

Table 6.7 DNN-HMM lipreading word accuracy with various temporal context sizes.

Model	Feature processing	Feature dim (±frame splicing)	Word accuracy (%)	
			SD	SI
CD-GMM+SAT	fMLLR	40	28.61 ±0.8	25.91 ±0.7
DNN	fMLLR	40 (±0)	43.55 ±0.4	40.00 ±0.9
		120 (±1)	45.35 ±0.9	41.16 ±0.5
		200 (±2)	45.31 ±0.6	41.42 ±0.7
		280 (±3)	45.72 ±0.7	40.46 ±0.6
		400 (±4)	45.51 ±0.6	40.89 ±0.7
		440 (±5)	**45.83** ±0.6	**41.66** ±0.4
		520 (±6)	45.20 ±0.8	40.88 ±0.5
		600 (±7)	45.06 ±0.9	40.78 ±0.5

Table 6.7 presents the word accuracy of lipreading using the DNN model optimised on CE with various dimensions of the fMLLR input features compared to the GMM-SAT model. We clearly see the learning ability of the DNN systems even with no spliced features ($n = 0$) by a 14.94% increase in accuracy on SD (from 28.61% to 43.55%) and 14.09% on SI (from 25.91% to 40.00%). The benefit of augmenting the neighboring context frames brings further improvement in the accuracy (at least 2.28% on SD and 1.66% on SI) compared to using the current frame alone. Here, the best performance of baseline SD is 45.83% and SI is 41.66% with ±5 context.

6.5.3 Word based DNN-HMM sequence discriminative training

The DNN-HMM training via CE optimisation is the most common objective function to construct a classification model but it is based on a frame-by-frame comparison. For lipreading where co-articulation and context are important, effective training of a DNN-HMM implies consideration of a longer window. We use sequence-discriminative training techniques to fine-tune the existing DNN parameters, initially trained by CE, by using sequence-level criteria which take into consideration the HMM topology and language model. There are some reports of speech recognition systems that apply the sequence-discriminative training in the DNN acoustic model [152, 139] and also the RNN-LSTM acoustic model [127, 153]. This experiment examines three criteria for sequence-discriminative training of the DNN visual speech model: maximum mutual information (MMI); state-level minimum Bayes risk (sMBR) and minimum phone error (MPE).

We conduct experiments on sequence discriminative training on top of the DNN model initiallised by CE via the three training criteria, sMBR, MPE, and MMI. First, decoding lattices and alignments of training data are needed. Here, the DNN trained on the CE criterion has been used as a seed model to decode training utterances by utilising a unigram language model. The DNN model trained on CE is used for generating the posterior probabilities, then the raw state accuracy of each sentence in the lattice is computed. These steps are essential because they give us the actual performance of the current visual speech model by decoding the training data with fewer constraints in the language model so we can identify the errors that need to be improved via sequence discriminative training criteria (Chapter 4.2.3). We set the learning rate to 1×10^{-5}, while the acoustic-scale and LM-scale are 0.1 and 1.0 respectively as in [152].

Table 6.8 Comparisons of three sequence-discriminative training criteria sMBR, MPE, and MMI against the DNN baseline. The results show word accuracy of the first iteration.

Model	Training objective	Word accuracy (%)	
		SD	SI
DNN-beseline	CE	45.83 ±0.6	41.66 ±0.4
sMBR	CE + sMBR	50.67 ±1.0	47.11 ±1.1
MPE	CE + MPE	51.13 ±0.8	47.15 ±0.9
MMI	CE + MMI	49.19 ±0.8	45.06 ±0.8

Table 6.8 shows the word accuracy before and after applying sequence discriminative training. The significant improvement can be seen in all cases compared to CD-DNN trained on CE ($p < 0.05$ by the MP test in all cases).

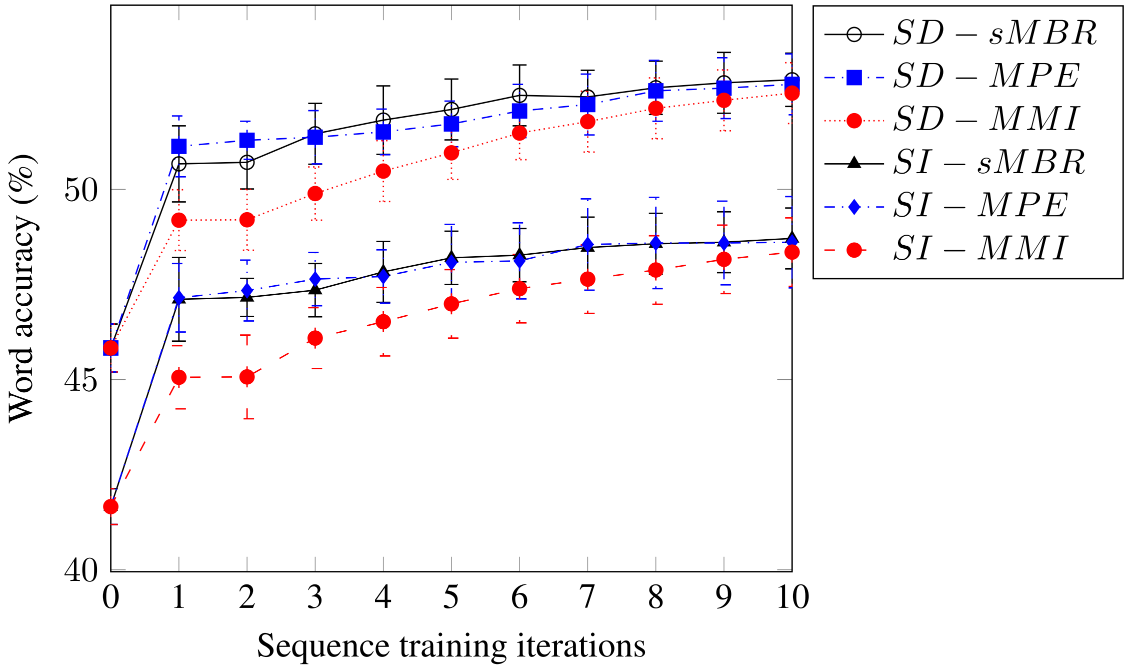


Fig. 6.8 Comparison of lipreading performance of SD and SI systems among three discriminative training criteria; sMBR, MPE, and MMI when we increase the training iterations. The best performance of SD is 52.88% on the 10th-iteration of sMBR and that of SI is 48.71% on 10th-interaton of sMBR. (Note: 0th-iteration means baseline DNN)

We also examine the word accuracy when increasing the number of training iterations. Results in Figure 6.8 illustrate the performance variation over training iterations and alignment updates. The 0^{th} iteration means CE. For the SD configuration, sMBR and MPE have small changes after the sixth iteration, while MMI still increases. However, the best result of SD is 52.88 % obtained from the 10^{th} iteration of sMBR (7.05% higher than CE). For the SI configuration the highest word accuracy is 48.71% at the 10^{th} iteration of sMBR (7.05% higher than CE).

6.6 Conclusions

This section presents the construction of computer lipreading on a large vocabulary continuous speech recognition task using the TCD-TIMIT corpus. We have built a successful lipreading system using DNNs and sequence discriminative training. Comparing our result with a conventional HMM, we see that performance has increased from around 25% word accuracy to around 48% in speaker independent mode. Looking in more detail, large improvements are obtained using fMLLR, the DNN rather than a GMM, some temporal stacking and the use of sequence discriminative training. The sequence discriminative training converges quickly (two or three iterations) but the method does not matter very much (sMBR, MPE, MMI). We think that if we had more data then the methods would differ and possibly more iterations would give greater benefit.

The best word accuracies are 52.88% in speaker dependent and 48.71% in speaker independent obtained on the phoneme unit rather than the viseme unit. We have added more evidence to the argument that phoneme classifiers can outperform those of visemes. Whilst there remains debate about visemes, but given the evidence showing an improvement in word accuracy from the reduction in homophene words in a pronunciation dictionary, we suggest that phonemes are the current best class labels for lipreading.

One of the disadvantages of the DNN is that it is not easy to examine the internals of the network to discover from where it is getting its performance. However, there is a clue in the previous observation which is that the DNN appears to make the most difference to word accuracy rather than unit accuracy. Visual speech is notorious for extensive co-articulation, so the implication is that either there are differences in the window length between the GMM and the DNN or the DNN is better able to model co-articulation than the GMM. Here we were able to use identical features, and we also found the DNN is superior; furthermore we know the DNN is better able to learn data structured on non-linear manifolds. In the next Chapter, we examine where the successful improvement of DNN model comes from.

Chapter 7

Investigation of visual representations

In previous Chapters we have shown that lipreading via DNN-HMMs significantly outperforms the conventional GMM-HMM system in small and large vocabulary tasks. A sequence discriminative training process trained on top of a DNN-HMM offers significant improvement in word accuracies of lipreading systems on the TCD-TIMIT large vocabulary task.

This Chapter investigates the effect of the feature extraction and feature transformation processes used in DNN-HMM lipreading systems. The potential capability of a DNN model is that its deep structure performs feature transformation using a non-linear function. We describe the entire process of feature extraction and feature transformation as the extraction of visual representation. To examine the impact of visual representation, we select four types of appearance-based features to build lipreading systems. These features are the DCT, the DTCWT, Eigenlips, and the DAE. The evaluation task is a large vocabulary continuous speech task using the TCD-TIMIT corpus.

The contributions of this chapter are:

- We achieve the highest word accuracy reported on TCD-TIMIT task. There are 57.35% word accuracy on the speaker-dependent task and 53.83% word accuracy on speaker-independent task. We conclude that it is possible to achieve more than 50% word accuracy in lipreading on a large vocabulary task with DNN-HMMs in combination with a proper feature transformation method.

- Although a DNN may not, in principle, require feature engineering, with only the seven hours of available training data, feature engineering as a prepossessing step is necessary. We have noticed that many features are influenced by the speaker identity. The support evidence shows a large gap between the word accuracy of the seen and unseen speaker scenarios. We found that LDA/MLLT and fMLLR help minimise the speaker identity effect and improve word accuracy.

- We investigate visual speech modelling via the hybrid DNN-HMM approach where we compare results between training shallow and deep models. We found that using a deep model benefits word accuracy. Our conclusion is two-fold. First, the higher capacity in DNN models makes them better able to handle the multivariate features than the GMM. Second, unlike the EM algorithm which optimises the likelihood to fit the fixed distribution of the dataset, each DNN layer represents a dataset in a different form which means that it produces feature transformations throughout hidden layers. The idea behind the DNN model is that it does not require feature engineering since feature transformations can be learnt by minimising the error between the prediction and target classes. We support our conclusion with visualisations where we have shown that the distribution of features extracted from the last layer of the network tends to be more aligned to ground truth words, compared to features extracted from the first layer.

The contributing publication of this chapter is:

- Thangthai, K., Harvey, R., **Building large-vocabulary speaker-independent lipreading systems**. – In *Proceedings of the Annual Conference of the International Speech Communication Association INTERSPEECH 2018*, Hyderabad, India, pp 2648–2652, 2018.

7.1 Visual features

In the previous Chapter, we use two common features to construct lipreading systems which are the DCT and Eigenlips. Here we add two new features which are the DAE (Chapter 4.4.1) and DCTWT (Chapter 4.4.1). We extract all features from 64 × 128 pixels grey-scale lip images.

We use three feature processing steps: (1) z-score normalization, (2) LDA/MLLT, (3) and fMLLR [42] transformation. Previous reports in lipreading use different sizes of LDA context window i.e. ±3 [4, 1], ±7 [114, 148], and 40 dimensions were retained. Here we tune these parameters for each feature to obtain their best word accuracy in GMM-HMM systems (in Appendix A). Therefore we define specific parameters in LDA/MLLT and fMLLR transforms for each feature. We use ±14 context window and retained 25 dimensions for Eigenlips features. And we use ±11 context window and retained 20 dimensions for DCT features.

7.1.1 Deep Autoencoder (DAE)

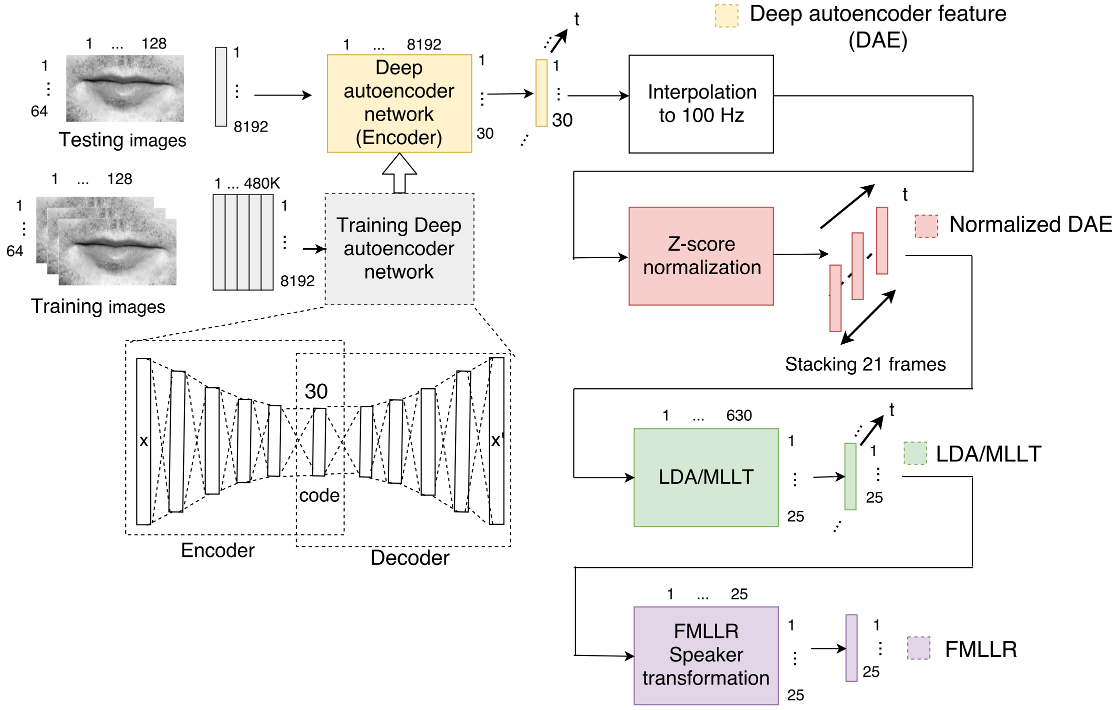


Fig. 7.1 Deep autoencoder feature extraction and feature processing methods.

Our 30-dimensional DAE feature is obtained from 64 × 128 pixels grey-scale lip images. Figure 4.23 shows the DAE network construction. There are 11 hidden layers where the units in the encoder layer are (1024, 512, 256, 128, 64) and the units in the decoder layer are (64, 128, 256, 512, 1024) and 30 units in the code layer. We use the ReLU activation function (more details in Section 4.2.2) in each unit in the hidden layers except in the code layer where we use the linear unit. The DAE model is trained on 480k images which were obtained from all training videos and optimised for the MSE loss-function via the Adam optimisation algorithm [71] using 50 epochs and mini-batch size of 256. The DAE network has been implemented in KERAS [25] on Theano backend [3] and trained on a single GPU node in the high-performance computing cluster (HPC) at UEA.

In the LDA/MLLT transform on DAE, we use dynamic information covering 21 frames window (stacking ±10 frames) and retain 25 dimensions. The fMLLR feature also retains 25 dimensions.

7.1.2 Dual-tree complex wavelet transform (DTCWT)

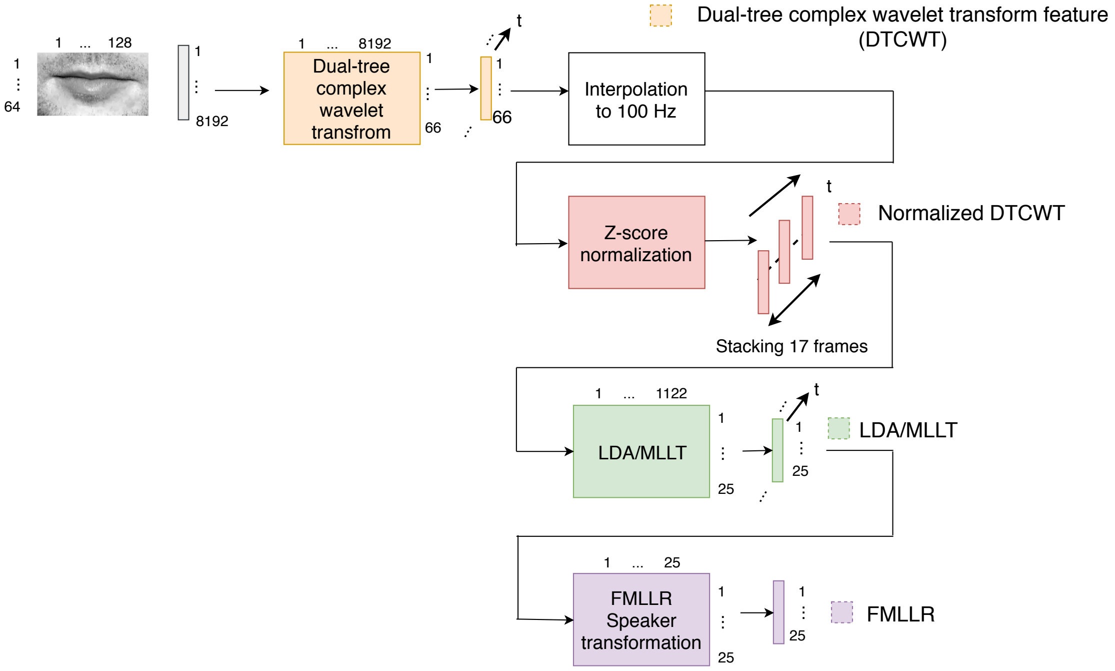


Fig. 7.2 DTCWT feature extraction and feature processing methods.

DCTWT was proposed for lipreading by Feng and Wang [38] where they obtained an 8% improvement compared to PCA via 24-dimensional DTCWT features. Their experiment investigated Chinese isolated-word and regular digit-string tasks. In our work, we retain detail at fifth, sixth and seventh orders. The DCTWT feature becomes a 66-dimensional vector. We use the magnitude to identify the DTCWT order. More details of the DTCWT feature can be found in Section 4.4.1.

For DTCWT, we obtain the best word accuracy from LDA/MLLT with stacking ±8 frames (covering 17 frames window) and retaining 25 dimensions.

7.2 Experimental results

We evaluate visual representations covering the extraction of static features, dynamic features, speaker adaptive features, and the transformations inside a DNN model. We build lipreading systems using GMM-HMM and DNN-HMM training methods with three-state left-to-right HMM topologies. We extract four types of static appearance-based features: DCT; DTCWT; Eigenlips and DAE, as summaried in Table 7.1. We use 30-dimensional Eigenlips; and DAE; 44-dimensional DCT; and 66-dimensional DTCWT. We then apply LDA/MLLT and fMLLR feature transformation methods. The evaluation task is large vocabulary continuous speech recognition using the TCD-TIMIT corpus, detailed in Section 2.4.3. There are two scenarios: speaker dependent (seen speakers), and speaker independent (unseen speakers). We report the word accuracy on the mean of three-fold cross validation with ±1 standard error.

Table 7.1 Summary of features used in this experiment.

Representation	Transform method	Feature dimension
Eigenlips	Unsupervised learning via PCA transform	30
DAE	Unsupervised learning via non-linear transform	30
DCT	Cosine transform	44
DTCWT	Wavelet transform	66

There are five sets of experiments as following here.

- We optimise parameters in GMM-HMM training in Appendix A.
- We investigate the effect of feature transformation methods in DNN-HMM systems (Section 7.2.1).
- We examine the impact of the deep structure in DNN-HMM model by comparing between the shallow network (1-layer) and the deep network (6-layer) (Section 7.2.2).
- We evaluate visual features used in DNN-HMM training with sMBR (Section 7.2.3).
- We investiate the effect of longer order of word N-gram language model (Section 7.2.4).

All experiments were done using the Kaldi speech recognition toolkit (Povey et al. [116]). We apply the same training steps as presented in Chapter 6.4.1. We use word bigram language models and the TCD-TIMIT dictionary which are the same as in Chapter 6. The DCT, DTCWT and Eigenlips features are implemented in Matlab, and the DAE feature is implemented via the Keras neural network API [25] with the Theano framework [3].

7.2.1 Effect of feature processing for DNN-HMM

This experiment aims to investigate the effect of feature transformation in DNN-HMM training. Does the DNN-HMM need feature preprocessing since a deep model has its feature transformation through the network layers? We investigate this effect by using different feature transformation methods and comparing to standard feature processing, via the fMLLR feature transform. Here the DNN-HMM visual speech models are trained on six hidden layers and with sigmoid non-linearity 2048 units per layer. We optimise the DNN parameters using the standard CE. The training parameters are equivalent to the system explained in Section 6.5.2. For each input feature type, we splice ±5 consecutive frames as a dynamic feature covering 11 frames context (optimised in Chapter 6.5.2). We use the constrained time alignments generated from GMM-HMMs with speaker adaptive training.

The scatter plot in Figure 7.3 shows word accuracy (y-axis) of DNN-HMM lipreading as a function of feature transformation methods (x-axis). There are four feature transformation methods: original, Z-Score with Delta, LDA/MLLT, and fMLLR. The different type of marks refer to four static features: Eigenlips, DAE, DCT, DTCWT. The different colours refer to scenarios where SD shows in blue and SI shows in red.

In Figure 7.3, there is a clear trend of increasing word accuracy from each step of feature transformation. This trend applies to all features. The original untransformed features have the lowest performance in both scenarios. The dependency of speaker identity is evident in the difference in performance between the SI and SD systems which is around 15%. Applying feature normalisation and deltas is highly beneficial to both SD and SI word accuracy, especially with DAE features where the word accuracy improves by almost 20% compared to the original results. Next, we found that LDA/MLLT and fMLLR help minimise the speaker identity effect and enhance word accuracy which is shown by the small gap between the performance on seen and unseen speakers. The explanation for the LDA/MLLT is that it transforms the feature space to satisfy the class discrimination which indirectly reduces speaker identity effects.

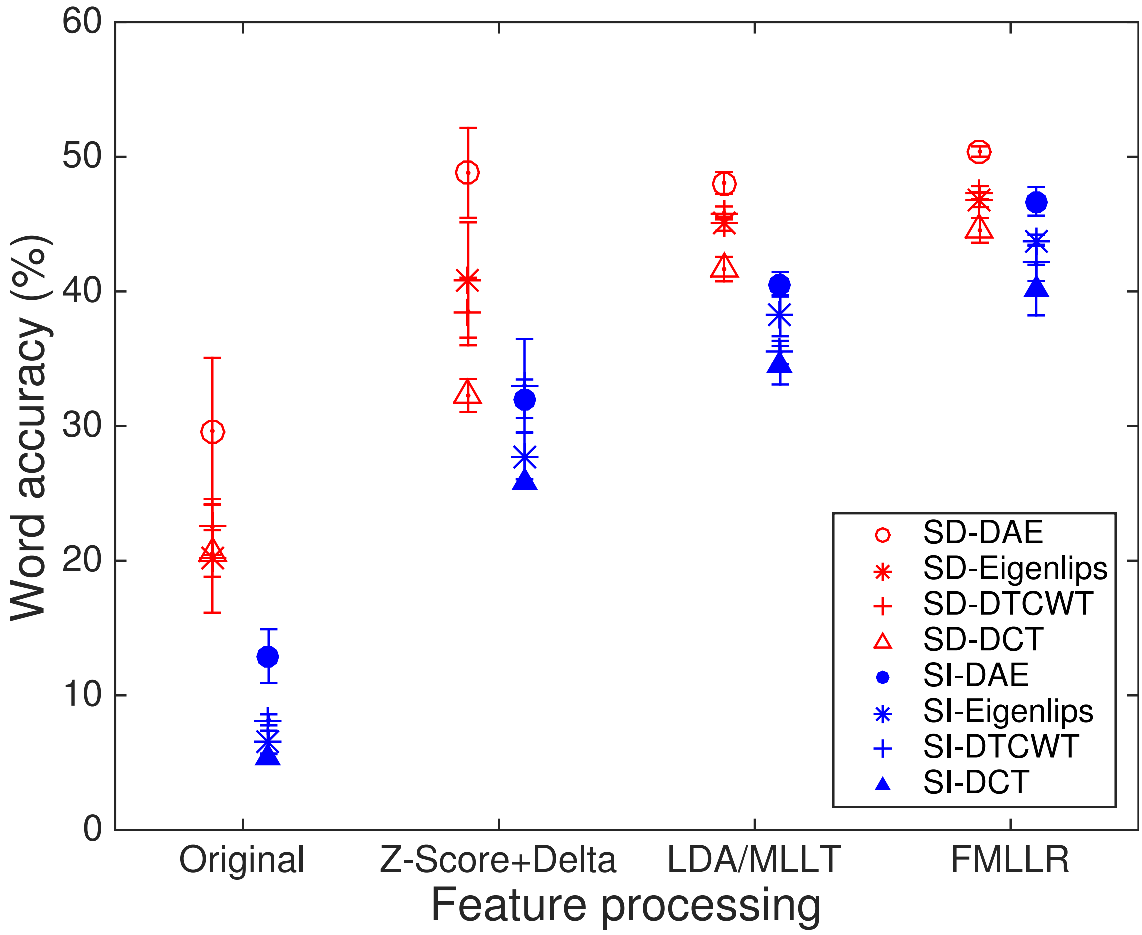


Fig. 7.3 Word accuracy (%) of DNN-HMM lipreading on speaker dependent (red) and speaker independent (blue) comparing four-types of feature representations as a function of feature transformation methods with ±1 standard error.

The fMLLR is directly proposed to solve the speaker identity effect. Overall the highest word accuracy of speaker-independent lipreading is 46.69% using the DAE feature with the fMLLR transformation method.

To conclude, in this task we found that using feature processing and transformation is useful. Although each DNN layer performs a unique representation, it is still unable to handle complexity in the original feature space. This is also due to the small size of the available training data. Thus, using the intermediate representation of a feature helps obtain a better result over the original feature. More detail of visualising features and transformations are in Section 7.3.1.

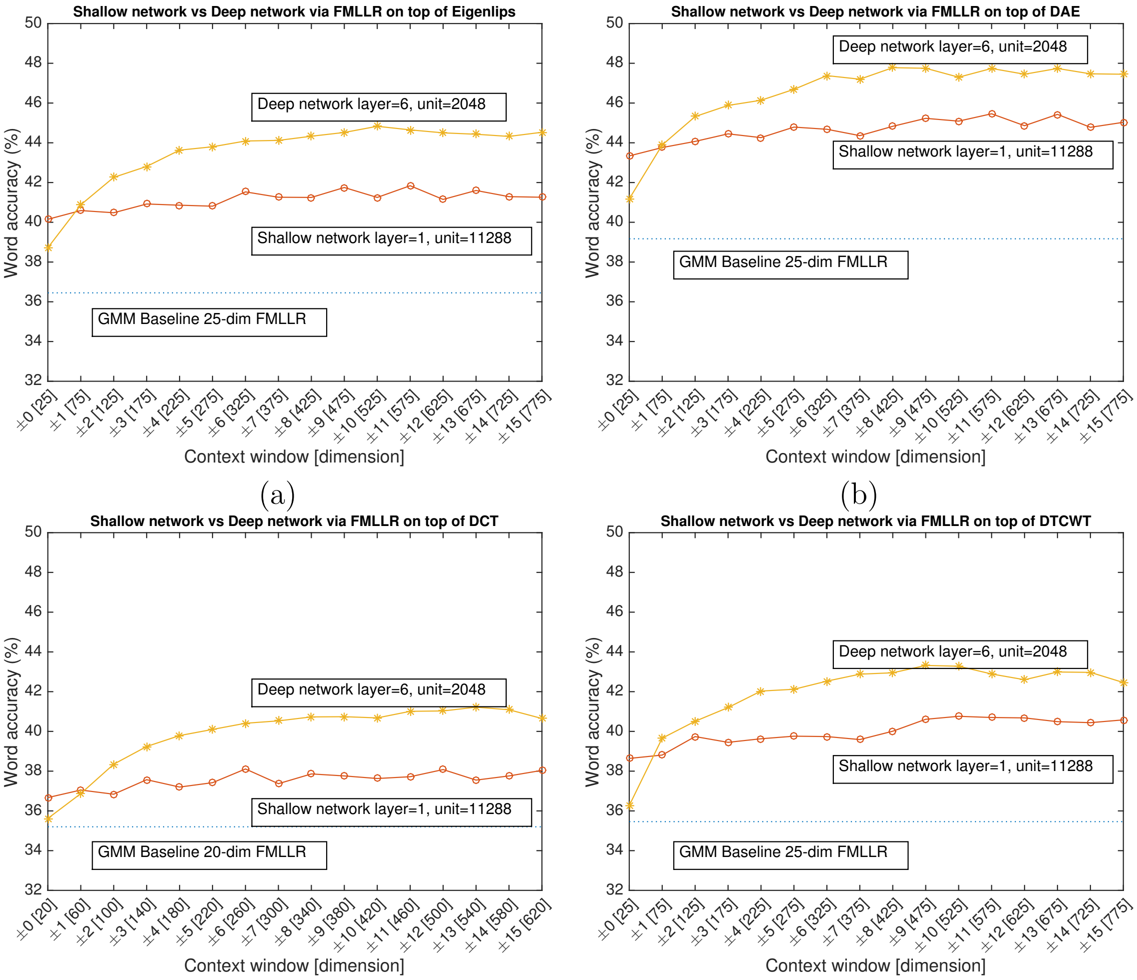
7.2.2 Effect of deep and shallow network

In the previous experiment we investigated the affect of feature transformation. We found that using fMLLR feature transform is useful and provides the best result in DNN-HMM training.

One interesting question concerning the DNN model is which property of the model makes it superior to other models. There is previous study that attempts to answer this question in speech recognition [103]. In [103], they compared speech recognition performance between GMMs and DNNs. For the DNN, they trained a one-layer neural network and a DNN model with six layers. They also varied the size of context input between 1 and 13 frames. Here we would like to consider the same question on lipreading. We study the difference between the shallow structure and deep structure of the DNN model. We compare the performance of a DNN with a shallow neural network model. We set up a shallow model using one hidden layer. We then increase the number of nodes in the shallow network to 12288 which is comparable to 6 layers × 2048 nodes in the deep structure. We vary the dimensionality of input features to observe the differences in learning ability between the shallow and the deep model. Here we splice the input features with $\pm N$ context frame where N={0,1,3,...,15}. We use fMLLR feature processing. The evaluation task is speaker independent (SI).

We compare the performance of lipreading between the shallow neural network and the deep neural network model in the hybrid configuration with HMMs. We evaluate on the speaker independent task (SI). Figure 7.4 provides the results from four feature types: Eigenlips (a), DAE (b), DCT (c), DTCWT (d). The graphs show word accuracy of the DNN and the Shallow NN as a function of context window (and corresponding input dimension).

Although we evaluate four different features, the performance trend is similar in many ways. Firstly the DNN and the shallow NN work better than the GMM. In fact, at the same input dimension ±0, the shallow network outperforms the deep network and the GMM. Second, the DNN and shallow NN do not suffer from the curse of dimensionality problem (explained in Chapter 4.2.1) and the features do not require to be decorrelated. These capabilities are seen from the improvement of accuracy even when we increased the feature dimension from 25 to 775. Third, with the same capacity, deeper is better. The results show that DNN performance improves more than the shallow NN when we increase the feature dimension. These results are consistent with [103] which concludes that the gain of the DNN is associated with the long context window of speech frames in the input feature vectors.



(a)

(b)

(c)

(d)

Fig. 7.4 Word accuracy (%) of speaker independent lipreading (SI) comparing the shallow network (one-layer) and deep network (multi-layer) as a function of context window. The shallow network has one-layer with 12288 nodes, and the deep network has six-hidden layers with 2048 nodes/layer. The graphs on the top are the results from (a) Eigenlips, and (b) DAE. The graphs at the bottom are results via (c) DCT, and (d) DTCWT. Also, plotted in each graph is the result of the GMM baseline.

One thing we would like to add to the conclusions of [103] is that the DNN, especially with the RBM pretraining, has representation learning in each hidden layer. By representation learning with the RBM, we mean that the DNN performs an unsupervised feature learning providing different feature transformations in its hidden layers. Moreover, the DNN model in general also has multiple feature transformations in the network via supervised fine-tuning with the backpropagation algorithm. This concept is obviously different from the GMM by its nature, where representation is changed mainly by feature engineering. The GMM tries to represent the distribution rather than modify the representation. More explanation of representation learning can be found in [13]. This might also explain why the shallow network performs slightly worse than the deeper one. We provide more discussion in this topic later in the analysis and discussion section 7.3.1.

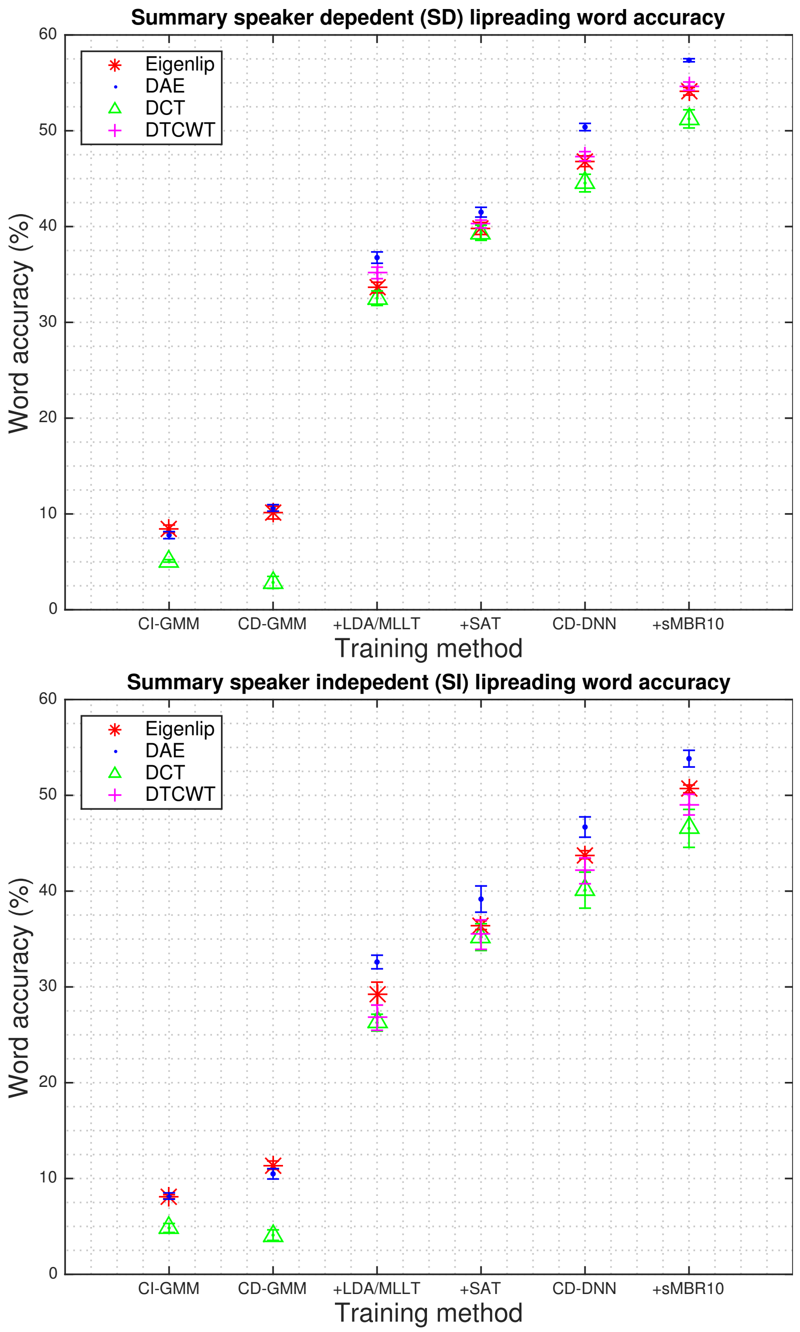
7.2.3 Comparing of representation summary results

The previous experiment demonstrates the effect of using the Deep neural network model and the shallow neural network model in lipreading. We found that the deep structure helps the model to achieve better performance than the shallow one. In this step we utilise the sequence discriminative training with sMBR in the DNN-HMM training. We present lipreading accuracy using our best training configuration.

This experiment aims to investigate the final result of lipreading in the large vocabulary task on the TCD-TIMIT. Here we demonstrate the summary results of visual speech modelling trained on four feature types: Eigenlips, DAE, DCT, and DTCWT. Results show in sequence of model training methods; starting from GMM to DNN with sMBR. The DNN-HMM with sMBR was initialised by RBM pretraining and CE then optimised via 10 iterations of sMBR (as explained in Chapter 4.2.3).

Here are the feature processing methods in each step. In CI-GMM and CD-GMM, we use delta features. In +LDA/MLLT, we use the CD-GMM model with the LDA/MLLT transformation. In +SAT, we use the CD-GMM with the LDA/MLLT and fMLLR feature transforms. Then the fMLLR feature is used in CD-DNN and CD-DNN with sMBR training.

Figure 7.5 summaries the results of our lipreading system trained on Eigenlips, DAE, DCT, and DTCWT features. Figure 7.5 (top) shows the SD results and Figure 7.5 (bottom) shows the SI results. Shown on the x-axis is the model training sequence that we used in the entire training process, from GMM-HMM to DNN-HMM with sMBR. As expected, DAE yields the best word accuracy. It obtained 57.36% in SD and 53.83% in SI via DNN-HMM on sBMR training. Compared to the DNN, sMBR offers a 6.98% gain in SD, and 7.13% gain in SI. If we compare only results from the sBMR, in the SD scenario DAE obtained 6.12% higher than the DCT, 3.25% higher than the Eigenlips, 2.70% higher than the DTCWT. In the SI scenario, DAE offers 7.28% higher than the DCT, 3.16% higher than the Eigenlips and 4.80% higher than the DTCWT.


Summary speaker depedent (SD) lipreading word accuracy

Summary speaker indepedent (SI) lipreading word accuracy

Fig. 7.5 Comparison of lipreading word accuracy (%) over four types of representations as a function of model training methods. Speaker dependent (SD) results are at the top, and speaker independent (SI) results at the bottom.

It can also be seen from Figure 7.5 that the majority of gains come from the training method rather than the features. Here we observe that the highest increase in the word accuracy is obtained from the transition between the CD-GMM with delta and

CD-GMM with LDA/MLLT with more than 20% gained in both scenarios. It implies that the information provided in the original feature with delta feature processing is too complex for visual speech modelling. We visualise the data distribution of original features in Figure 7.6 and 7.7 in the analysis section.

Significant tests

The present results are different in term of the standard errors across the three-fold cross-validation results. However, a speech recogniser has specific methods to compare performance between algorithms whether they produce identical results or not. Here, we report significant tests of lipreading on the DNN-HMM with sMBR using four techniques: Matched pairs sentence segment word error (MAPSSWE or MP), Signed paired comparison (SP), Wilcoxon signed rank (WI), and McNemar (MN). These methods compare error rate between algorithms in different levels. The MN test analyses the statistical difference between algorithms using the sentence error rate. The SP and WI compare word error rate of each speaker. The MP test, most used in ASR, compares word error rate of each sentence segment level. More detail of each algorithm can be found in Chapter 4.6.

Table 7.2 reports the SD lipreading results. It presents the significant measurement methods in each pair of features. The first column indicate the test methods: MP, SP, WI, MN. The second column and the first row show four feature types: Eigenlips, DAE, DTCWT, DCT. The significant tests show that the result of DAE features significantly outperforms other features with (p<0.001). Considering the MP test, there are no significant differences between results of Eigenlips compared with DTCWT and Eigenlips compared with DCT. The result from DTCWT is statistically better than the DCT result ($p = 0.011$).

Turning to the unseen speaker test in Table 7.3, the result of the DAE feature is not significantly better than the Eigenlips feature. The statistical tests of SI lipreading show that DAE and Eigenlips results are not so different in terms of word error rate in speaker level according to the MN test. The DAE and Eigenlips results are significantly better than DTCWT and DCT results. The DTCWT result significant outperforms the DCT result.

To summarise, this section presents the evaluation of lipreading systems on a large vocabulary continuous speech recognition task. We evaluate lipreading performance on the TCD-TIMIT database in two scenarios: SD and SI. We investigate the visual representations using four types of features: Eigenlips, DAE, DCT, and DTCWT.

Table 7.2 Significant tests on speaker dependent set (SD). Each cell shows the P-value for a pairwise comparison between the tests. An underline indicates P-value < 0.05 and a double underline indicates P-value < 0.01.

Test Abbrev.		Eigenlips	DAE	DTCWT	DCT
MP	Eigenlips		DAE 5.8497e-09	Eigenlips 0.472	Eigenlips 0.070
SP			DAE <0.001	DTCWT <0.001	Eigenlips <0.001
WI			DAE 1.3006e-05	Eigenlips 0.352	Eigenlips 0.162
MN			DAE 7.9784e-04	DTCWT 0.0337	DCT 0.8552
MP	DAE			DAE 1.4794e-06	DAE 5.3539e-13
SP				DAE <0.001	DAE <0.001
WI				DAE 1.5065e-04	DAE 2.4772e-06
MN				DAE 0.4594	DAE 0.0067
MP	DTCWT				DTCWT 0.011
SP					DTCWT <0.001
WI					DTCWT 0.112
MN					DTCWT 0.0289

Table 7.3 Significant tests on speaker independent set (SI). Each cell shows the P-value for a pairwise comparison between the test. An underline indicates P-value < 0.05 and a double underline indicates P-value < 0.01.

Test Abbrev.		Eigenlips	DAE	DTCWT	DCT
MP	Eigenlips		DAE 0.007	Eigenlips 1.7882e-06	Eigenlips 2.3742e-23
SP			DAE 0.049	Eigenlips 0.629	Eigenlips <0.001
WI			DAE 0.022	Eigenlips 0.028	Eigenlips 3.5698e-04
MN			DAE 0.0699	Eigenlips 0.0154	Eigenlips 2.7481e-04
MP	DAE			DAE 4.6638e-14	DAE 5.6543e-35
SP				DAE 0.002	DAE <0.001
WI				DAE 5.0141e-04	DAE 2.9460e-04
MN				DAE 4.8899e-05	DAE 4.1298e-07
MP	DTCWT				DTCWT 8.7470e-08
SP					DTCWT 0.049
WI					DTCWT 0.004
MN					DTCWT 0.3125

Here the best result of SD lipreading is 57.36% obtained on DAE using DNN-HMMs with sMBR sequence discriminative training. This result is statistically better than Eigenlips, DCT, and DTCWT, according to the MP test. The best result of SI lipreading is 53.83% which is obtained from the same configuration as the SD. This result significantly outperforms the Eigenlips, DCT and DTCWT.

7.2.4 Effect of language modelling

A language model (LM) can be used to constrain word combinations to form legitimate sentences or sentence fragments. It is usually learnt from the training text. We use five n-gram language models: zerogram; unigram; bigram (mainly used); trigram and 4gram. The term zerogram means that we use no language model (a unigram model with uniform probabilities). We show the word accuracy obtained from the 1-best and also show the corresponding lattice oracle which represents the quality of the lattices.

Table 7.4 Word accuracy (%) of lipreading system decoded with different language models.

Word-based n-gram	**Word accuracy (%)**		
language model (LM)	**SI testset**	**Lattice oracle**	**Guessing**
zerogram LM	6.24 ±0.1	48.54 ±0.6	1.63 ±0.7
unigram LM	10.69 ±0.3	60.10 ±0.5	2.04 ±0.6
(currently used) bigram LM	53.83 ±0.8	83.25 ±0.5	2.02 ±0.6
trigram LM	67.69 ±0.7	85.44 ±0.2	2.03 ±0.6
4gram LM	68.45 ±0.9	85.20 ±0.2	2.02 ±0.6

The results in Table 7.4 illustrate that the n-gram order of language modelling contributes to noticeable changes in lipreading performance. Lipreading performance gets below 10% without an LM, but the word accuracy increases significantly to about 68% when we use trigrams and 4grams. Although the lattices generated by bigram LM, trigram LM, and 4gram LM obtain similar quality observed by the oracle results, the word accuracy of a less LM-constrained system (bigram LM) is 15% lower than the higher LM-constrained systems (trigram and 4gram). These observations are consistent with what is known about human lipreaders who make considerable use of their linguistic and domain knowledge. We also evaluate if the language model dominates the lipreading performance by decoding a random noise vector. Results in the guessing column indicated that language modelling has successfully increased lipreading accuracy only in combination with a suitable associated visual input signal.

In the next section, we analyse lipreading results and provide visualisations of each representation. We also offer discussions and some practical suggestions to improve lipreading systems.

7.3 Analysis and discussion

The results so far indicate that extracting a good representation of visual speech is important to the improvement of lipreading accuracy. In the previous section we illustrate the increase in word accuracy when we utilise feature transformations and the DNN model. Moreover, we found that the deep layer in a DNN model yields further improvement than the shallow model. We reported that DAE significantly outperforms DCT and DTCWT features and achieved more than 50% word accuracy in both SD and SI scenarios.

In the next section, we aim to analyse lipreading results. First, we examine why the original visual representations are difficult to model. Second, we illustrate how feature transformations and DNN modeling affect visual representation.

7.3.1 Visualisation

As mentioned in the conclusion of the experiment in Section 7.2.1, feature processing methods are essential for visual speech modelling. This is because they transform the visual representation to associate with the linguistic unit of speech. Moreover, in Section 7.2.2 we conclude that DNNs cause feature transformations in each hidden layer. Here, we support our claim by visualising to see what occurred after transforming a feature inside the DNN layers. There are two questions we attempt to answer: first, what is the information contained in an original feature?; second, to explain why feature transformation is necessary. To find an answer, we analyses visual representations using t-SNE visualisation techniques, explained in Section 4.7.

Original representation

For answering the first question, we plot t-SNE of four feature types from their original representation. They are 30-d Eigenlips, 30-d DAE, 44-d DCT and 66-d DTCWT. We plot the distribution of each feature collected from multiple speakers. We then observe a cluster of data points at speaker level and word level. There are 2841 data points of utterance *Don't ask me to carry an oily rag like that* extracted from six speakers; three male and three female. To clarify, the t-SNE algorithm does not require the label of each data point. The data points are clustered by similarity. We then use the label to verify the clustered data points whether they are grouped by speaker similarity or word similarity.

We measure the t-SNE output by computing the magnitude of the class discriminant ratio using Fisher's Ratio analysis by considering the speaker class label and a linguistic class label. The class discriminant ratio is the ratio of the between-classes variance and the within-classes variance. The class discriminant ratio can be computed by

$$\text{Class F-ratio} = \frac{S_B}{S_W}, \tag{7.1}$$

where S_B is the between-class covariance matrix and S_W is the within-class covariance matrix. The definition of these covariance matrices are:

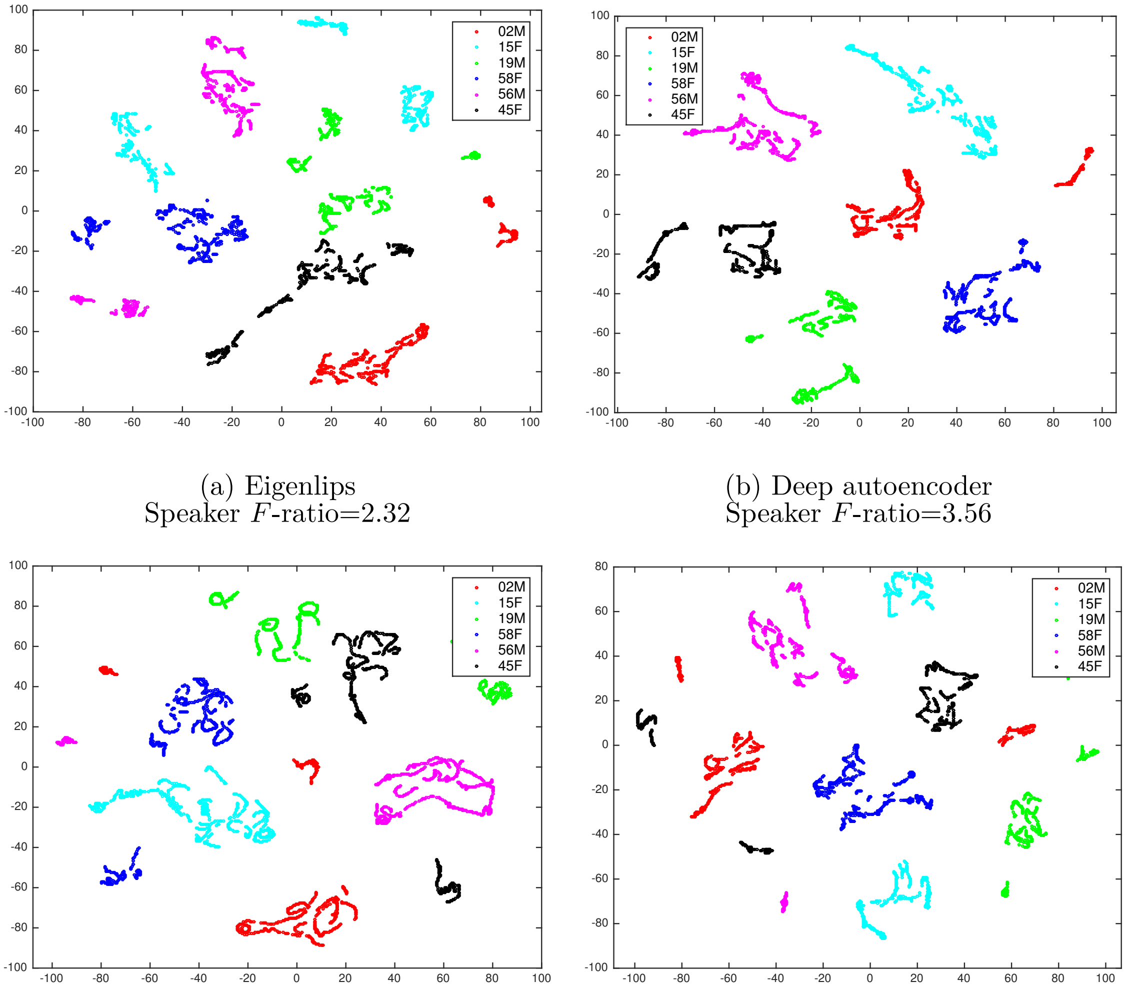
$$S_B = \sum_c (\mu_c - \bar{x})(\mu_c - \bar{x})^T, \tag{7.2}$$

$$S_W = \sum_c \sum_{i \epsilon c} (x_i - \mu_c)(x_i - \mu_c)^T, \tag{7.3}$$

where μ_c refers to the mean of each class and x_i refers to each data point. This F-ratio identifies the goodness of the data clustered regarding the provided class label. In our case we can also use this ratio to identify whether the data are clustered by speaker similarity or linguistic similarity. For instance, if the F-ratio of speakers is higher than the F-ratio of linguistic units, we assume that the feature represents speaker similarity and vice versa. Note that we compute three F-ratios using speaker label (spk), and two linguistic labels which are word label (w) and phonetic label (ph).

Figures 7.6 and 7.7 illustrate t-SNE visualisation of the data points obtained from four-types of original representations. In Figure 7.6 the data points are labelled by speakers. There are six speakers. The three male speakers are 02M labelled in red; 19M labelled in green; and 56M labelled in magenta. The three female speakers are 15F labelled in cyan; 58F labelled in blue; and 45F labelled in black. Here we observe that the data points are relatively separated according to speakers. These are also confirmed by the high value of speaker F-ratios. This is due to the fact that the original features directly capture variability existing in the data. These include environment, speaker, and phonological patterns variability. Here we use the controlled environment and the controlled sentences. The most variation of the data points is the speakers. Therefore, the representation in an original feature shows that it suffers from speaker dependencies. This finding is in agreement with Cox et al. [32] findings which showed a high sensitivity of visual features to speaker identity.

Figure 7.7 shows the same structure as Figure 7.6, but data points are coloured by words instead of speakers to reveal the linguistic representation of the data. There are



(a) Eigenlips
Speaker F-ratio=2.32

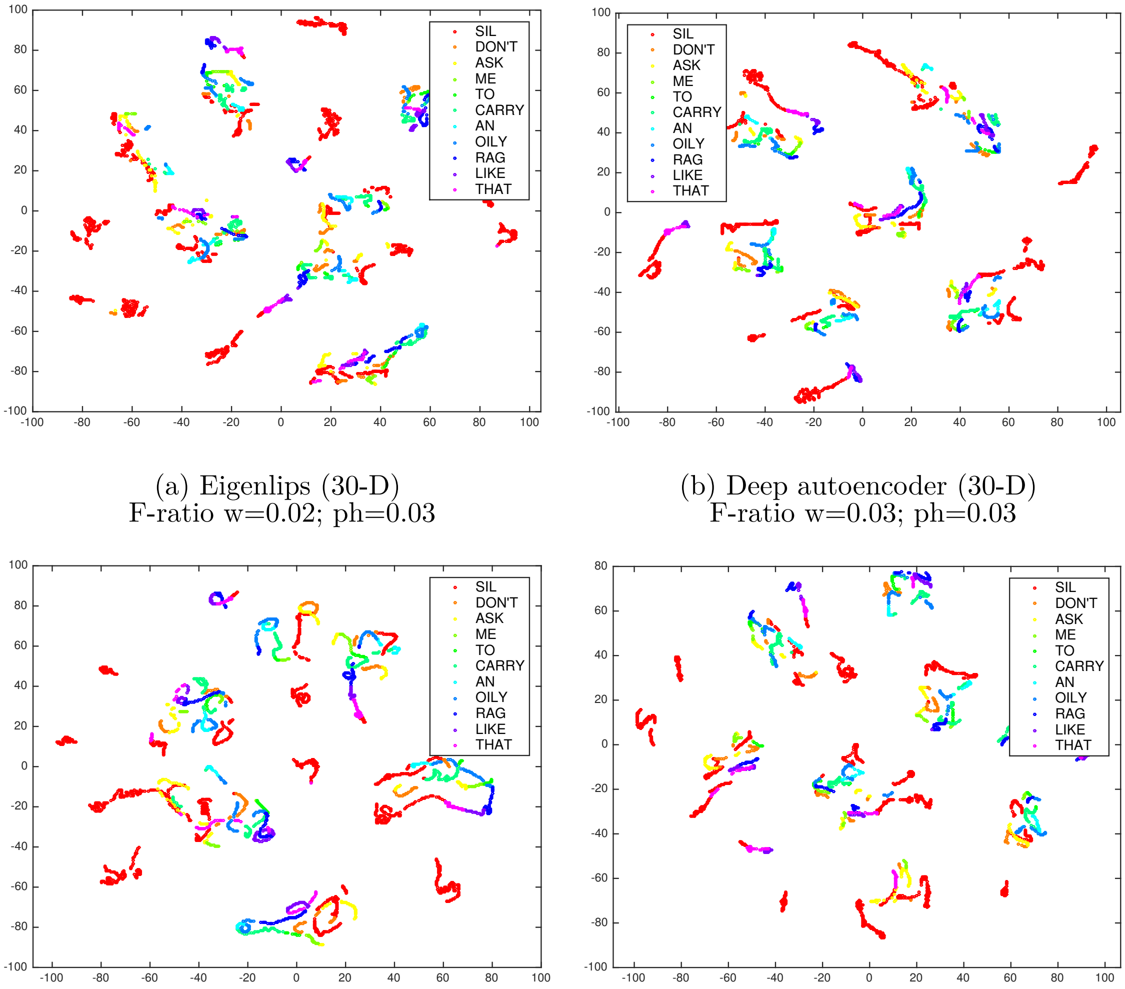
(b) Deep autoencoder
Speaker F-ratio=3.56

(c) DCT
Speaker F-ratio=3.20

(d) DTCWT
Speaker F-ratio=2.40

Fig. 7.6 T-SNE plots for the different types of features coloured by speaker. The speaker class discriminant ratio is provided underneath the plot of each feature.

11 words including silence. Each word is represented by a different colour. Here we plot the word level, instead of the phonetic level and the HMM state level, because it is much more apparent to see a change of linguistic representation. Here, we observed that the word and phonetic class discriminant ratios are much smaller than the speaker F-ratios. For example, the word F-ratio and speaker F-ratio of the Eigenlips features are 0.02 and 2.32 respectively. The low value of the F-ratio means that the between-classes variance of the data is smaller than the within-classes variance. In other words, it indicates that the distribution of the data is less dependent to the linguistic class than the speaker class.



(a) Eigenlips (30-D)
F-ratio w=0.02; ph=0.03

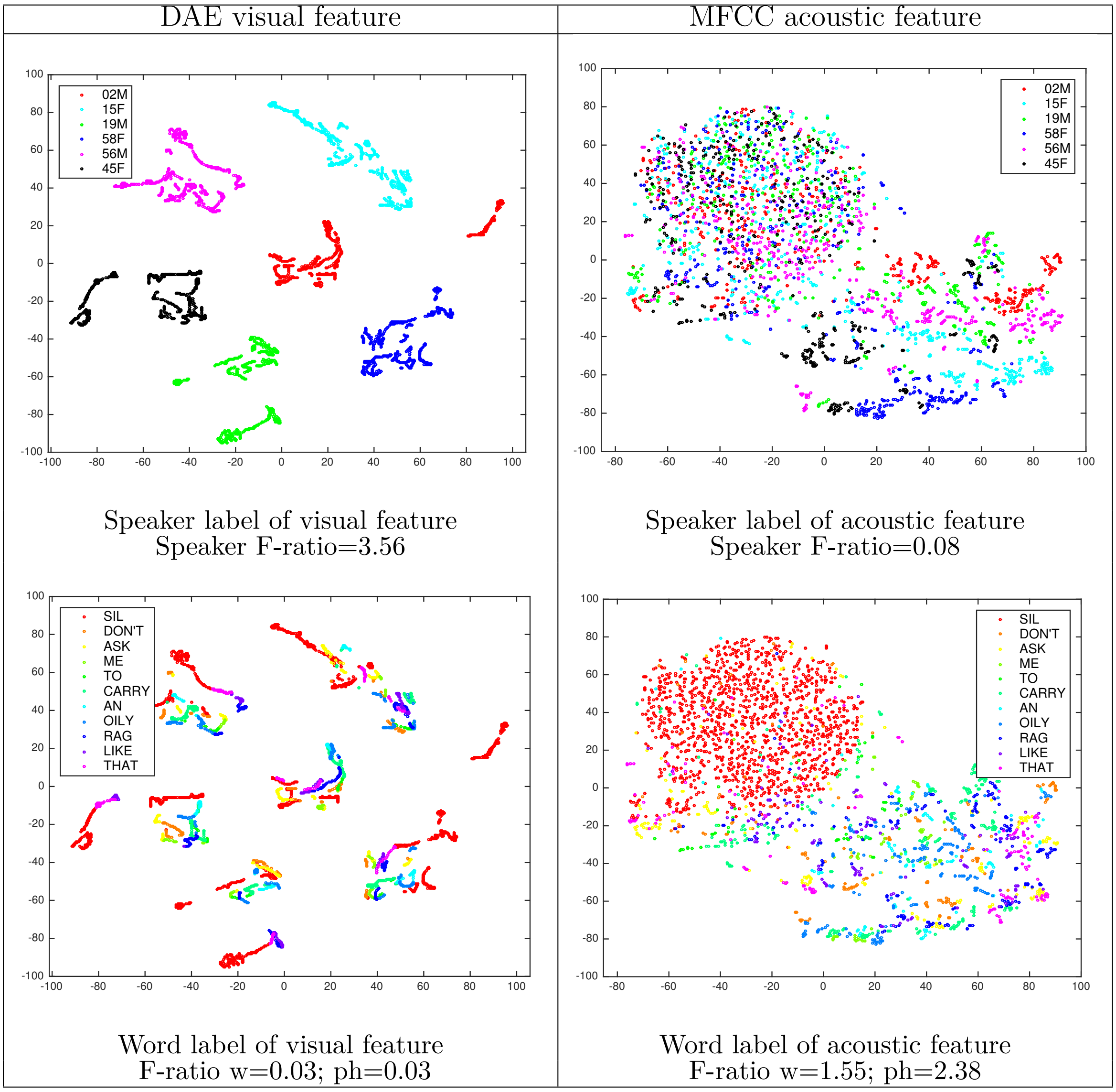
(b) Deep autoencoder (30-D)
F-ratio w=0.03; ph=0.03

(c) DCT (44-D)
F-ratio w=0.03; ph=0.02)

(d) DTCWT (126-D)
F-ratio w=0.02; ph=0.03

Fig. 7.7 T-SNE plots for the different types of features coloured by word. The linguistic F-ratio of word level (w) and phonetic level (ph) are provided underneath the plot of each feature.

Figure 7.8 illustrates the difference of the original representation between the visual (left) and acoustic (right) speech features. In Figure 7.8 on the top the data points are labelled (with colours) by speakers, while in the bottom the data points are labelled (coloured) by words. In these data, variations in the environment, and phonological patterns are controlled so variation is attributable to speaker identity. The – acoustic and visual features look quite different – in the visual we have clusters from different speaker identities, but the acoustic features have largely removed identity as we can see from the mixed-colour clusters. This is confirmed by the speaker F-ratio (3.56). In other words, features extracted from lip ROIs are highly speaker dependent. The red points in the bottom show silence which is well-clustered in the acoustic and, again, separated by identity in the visual case. In the acoustic data, a slight effect of speaker dependency can be found in the non-silence classes.


DAE visual feature

MFCC acoustic feature

Speaker label of visual feature
Speaker F-ratio=3.56

Speaker label of acoustic feature
Speaker F-ratio=0.08

Word label of visual feature
F-ratio w=0.03; ph=0.03

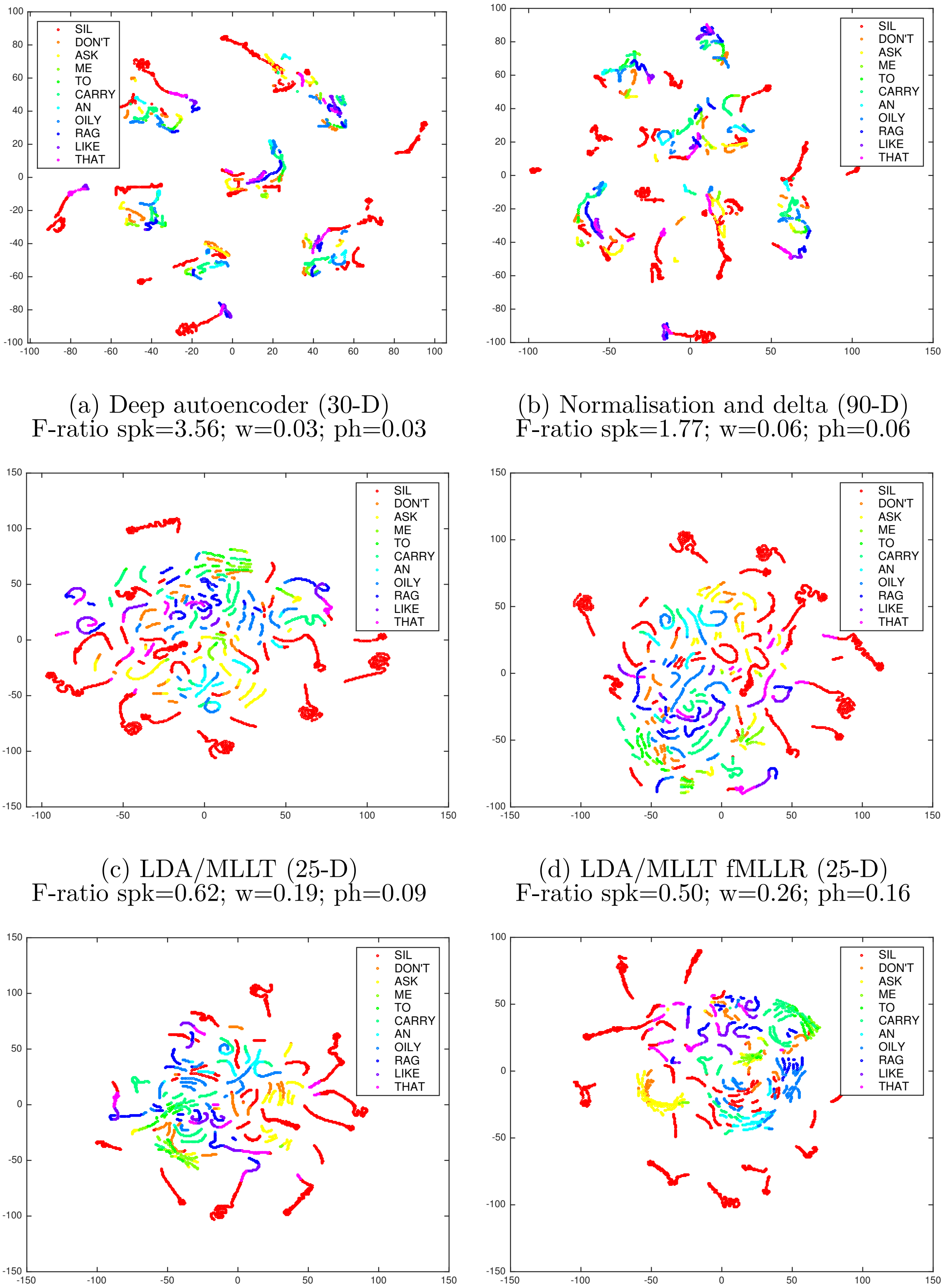
Word label of acoustic feature
F-ratio w=1.55; ph=2.38

Fig. 7.8 T-SNE plots show the comparison of the original representation of the DAE visual speech features (left) and the MFCC acoustic speech features (right).

Visual representation after applying a feature transformation method

As we show earlier, the original features, which are directly extracted from a video frame, represent other information rather than speech classes. The F-ratio reveals that the original space of all four features has higher speaker F-ratio than the linguistic F-ratio. In our lipreading system, we apply multiple feature transformation techniques to enhance the quality of features. These techniques are feature normalisation, adding delta, using LDA/MLLT, using fMLLR and using a DNN model. The goal of feature transformation is to provide a suitable structure of visual speech representation so that it is comprehensive and robust to predict a linguistic unit of speech. Here we use t-SNE visualisation to illustrate why these feature transformation methods play an important role to the improvement of the lipreading performance.

We run t-SNE on different feature transformation methods including a hidden layer in the DNN model. We compare the visual representation of DAE features in six steps of feature transforms: original, normalisation and delta, LDA/MLLT, fMLLR, DNN layer-1, and DNN layer-6. We extract features generated inside a trained DNN model by ignoring the weights from the last N components of the trained DNN model, where N refers to the number of layers. For example, if we would like to extract layer five from the six hidden layer model, we have to avoid the last two layers: the output layer and layer six. We then pass the input vector through the rest of the DNN model, and the output vector is the DNN feature generated from layer five. Note that this is the implementation on a DNN model generated from the Kaldi toolkit.



(a) Deep autoencoder (30-D)
F-ratio spk=3.56; w=0.03; ph=0.03

(b) Normalisation and delta (90-D)
F-ratio spk=1.77; w=0.06; ph=0.06

(c) LDA/MLLT (25-D)
F-ratio spk=0.62; w=0.19; ph=0.09

(d) LDA/MLLT fMLLR (25-D)
F-ratio spk=0.50; w=0.26; ph=0.16

(e) DNN layer-1 (2048-D)
F-ratio spk=0.28; w=0.71; ph=0.55

(f) DNN layer-6 (2048-D)
F-ratio spk=0.13; w=0.91; ph=0.61

Fig. 7.9 T-SNE plots of Deep autoencoder features with different feature transformation methods and transformation inside DNN layers. Three class discriminant ratios - speaker (spk), word (w), and phonetic (ph) - are provided underneath the plot of each feature transformation method.

We observe the change of the linguistic F-ratio and the speaker F-ratio throughout each feature processing step. We suspect if the data frames are more clearly clustered in their word group, the linguistic F-ratio will be increased, while the speaker F-ratio will be degraded. Figure 7.9 demonstrates that the structure of the data changed when different transformation methods had been applied. The data points are coloured in the word level. Figure 7.9 (a) is the original DAE feature. Figure 7.9 (b) shows the distribution when we applied utterance level mean-variance normalisation and delta as feature processing. This step contributes a tiny change in the data structure. But none of these words is clustered. The speaker F-ratio reduces from 3.56 to 1.77, although it is higher than the word F-ratio. This implies that the complexity of the original representation is slightly modified by using feature normalisation and delta.

The next step, Figure 7.9 (c), is the data distribution from the LDA/MLLT feature transformation. As described in Chapter 4.4.2, the LDA/MLLT method provides a discriminative feature associated with phonetic classes. It can be seen that the LDA/MLLT method enhances the linguistic representation of the features. The word F-ratio increases by 0.13 (from 0.06 to 0.19) from the previous step. The effect of speaker dependency has been reduced by 1.15 (from 1.77 to 0.62) observed from the speaker F-ratio. Here, the data points appear to connect into a small piece of words where joined data points mostly contain a single colour. The same word seems to be located in the similar region. Comparing to the previous step, this shows the significant changes in the data structure. This plot is the evidence to support why we got a considerable increase of word accuracy in Section 7.2.3. Comparing the LDA/MLLT in this plot, the fMLLR in Figure 7.9 (d) retains pretty much the same structure. The word F-ratio slightly increases from 0.19 to 0.26, while the speaker F-ratio reduces to 0.50.

Figure 7.9 (e) and Figure 7.9 (f) show the visualisation of the features extracted from the first hidden layer and the sixth hidden layer of the DNN model. As we explained earlier in Section 7.2.2, the capability of a DNN model is that it performs an intermediate representation through a hidden layer using a nonlinear function. In a definition of a DNN, the higher-level features present the hierarchy of concepts of the lower-level features [37]. It can be seen that the features in the sixth hidden layer have a better representation of the word classes than the first hidden layer. We observe that many words are well clustered. The word F-ratio in the first hidden layer increases by 0.45 (from 0.26 to 0.71) from the fMLLR step. And the word F-ratio of the sixth hidden layer additionally increases by 0.2 to become 0.91. Here the linguistic F-ratios of word and phonetic classes are higher than the speaker F-ratio which reduces to 0.13. This illustrates that the deep architecture of a DNN model represents a higher level in semantic concepts of the visual speech. This is the evidence to support the lipreading performance accomplishment of a DNN model.

Overall, these visualisations of visual speech representation confirm the importance of each step of feature transformation. They clearly reveal the different feature presentation produced inside a DNN model throughout its hidden layers. As a result, after enough feature transforms, the words are well defined and clustered. The effect of speaker dependency has been reduced, but is still higher than the speaker dependency effect found in the original MFCC feature. We observe that the silence representation of visual speech feature is not clustered even after feature transformation methods are applied. The silence phone (red) has a unique character: it is placed outside the word group and separated from one another. This raises a question about an issue of visual speech silence modelling which will be analysed in the following section and will be discussed in the investigation of the visual silence subsection.

7.4 Conclusion

This chapter demonstrates that lipreading systems can be built via the conventional techniques of acoustic speech recognition systems based-on DNN-HMMs and sMBR training. In a 6000-word vocabulary task, we achieved 57.36% word accuracy in the speaker dependent scenario and 53.83% word accuracy in the speaker independent scenario using DAE features and the fMLLR transformation method. The results and the visualisation of each feature indicate that feature processing steps are relevant to gain speaker-independent lipreading accuracy because they reduce the influence of speaker identity found in the original space of the DAE features.

We demonstrated that a feature transformation minimises the effect of decoding irrelevant information to predict a speech class. Such irrelevant information might exist in an original feature space. And a DNN model which has many transformed layers, is able to reduce these effects. However, a DNN model needs to be trained on a comprehensive training set. Unless, we provide them with a massive variety of data collected from a real situation. Otherwise, a step to pre-process features is still needed.

Although accomplishment of computer lipreading is dependent on the order of the n-gram language model, we observe that language modelling does not dominate the entire lipreading decoder a fact verified by the poor results of decoding the random signal. In the next Chapter, we provide an analysis of lipreading results.

Chapter 8

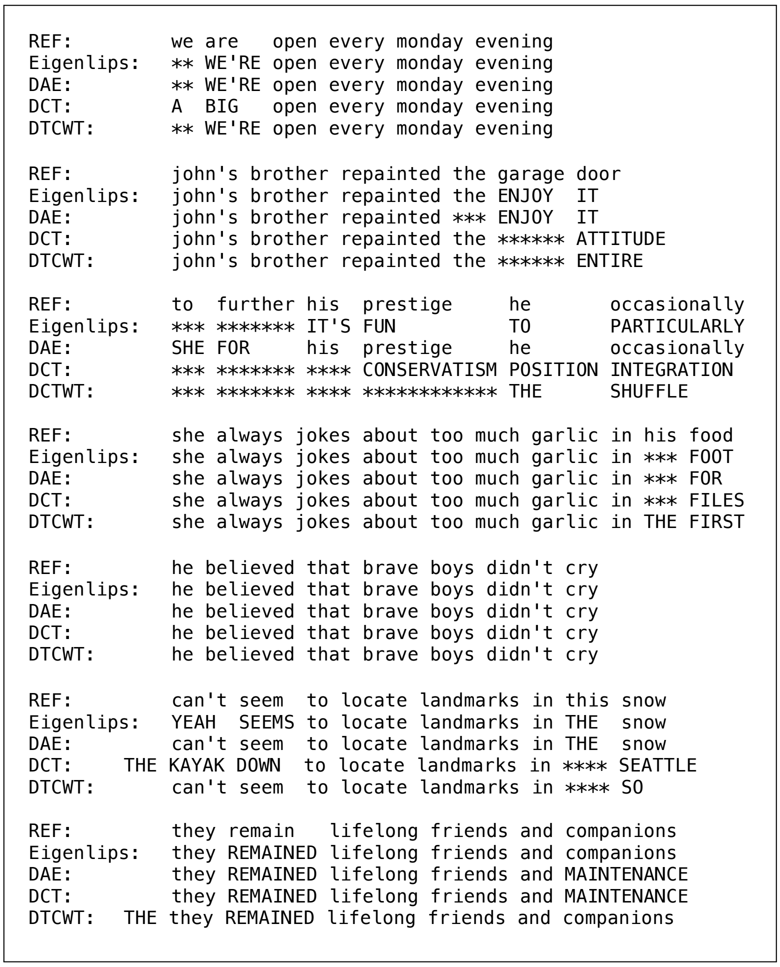
Results analysis

Developing of a lipreading system is a difficult task. Our best system for large vocabulary task yields 57.36% word accuracy in SD and 53.88% word accuracy in SI on the TCD-TIMIT dataset. This section aims to analyse lipreading results and understand lipreading errors. We investigate errors of lipreading transcriptions in the word level and the speaker level. For word level, we focus on error types, position, word length, and word frequency. For speaker level, we found high variation of word accuracy between talkers. Therefore, we investigate their general information and speaking styles involving five factors: age, gender, speaking rate, visual energy, and visual dynamics. Finally, we examine the further improvement of lipreading systems if we can handle visual silence.

The contributions of this chapter are:

- Word accuracy of a talker has a strong positive correlation between their SI accuracy and SD accuracy due to a speaker identity. We found a significant positive correlation ($p < 0.05$) between word accuracy and the rate of change in lip signal which we call a visual dynamic.

- We found that the complexity of lip movements in silence is a critical part which causes errors and is hard to predict. Our further investigation demonstarted that the word accuracy in speaker dependent task can be improved by 4.45% from 57.36% to 61.82% and by 5.59% absolute (from 53.83% to 59.42%) in a speaker independent task by simply eliminating the silence phone at the end of sentences. To date, this is the best word accuracy of lipreading reported on the TCD-TIMIT corpus which is a large vocabulary task covering 6000 words.



```
REF:         we are   open every monday evening
Eigenlips:   ** WE'RE open every monday evening
DAE:         ** WE'RE open every monday evening
DCT:         A  BIG   open every monday evening
DTCWT:       ** WE'RE open every monday evening

REF:         john's brother repainted the garage door
Eigenlips:   john's brother repainted the ENJOY  IT
DAE:         john's brother repainted *** ENJOY  IT
DCT:         john's brother repainted the ****** ATTITUDE
DTCWT:       john's brother repainted the ****** ENTIRE

REF:         to  further his  prestige     he       occasionally
Eigenlips:   *** ******* IT'S FUN          TO       PARTICULARLY
DAE:         SHE FOR     his  prestige     he       occasionally
DCT:         *** ******* **** CONSERVATISM POSITION INTEGRATION
DCTWT:       *** ******* **** ************ THE      SHUFFLE

REF:         she always jokes about too much garlic in his food
Eigenlips:   she always jokes about too much garlic in *** FOOT
DAE:         she always jokes about too much garlic in *** FOR
DCT:         she always jokes about too much garlic in *** FILES
DTCWT:       she always jokes about too much garlic in THE FIRST

REF:         he believed that brave boys didn't cry
Eigenlips:   he believed that brave boys didn't cry
DAE:         he believed that brave boys didn't cry
DCT:         he believed that brave boys didn't cry
DTCWT:       he believed that brave boys didn't cry

REF:         can't seem  to locate landmarks in this snow
Eigenlips:   YEAH  SEEMS to locate landmarks in THE  snow
DAE:         can't seem  to locate landmarks in THE  snow
DCT:     THE KAYAK DOWN  to locate landmarks in **** SEATTLE
DTCWT:       can't seem  to locate landmarks in **** SO

REF:         they remain   lifelong friends and companions
Eigenlips:   they REMAINED lifelong friends and companions
DAE:         they REMAINED lifelong friends and MAINTENANCE
DCT:         they REMAINED lifelong friends and MAINTENANCE
DTCWT:   THE they REMAINED lifelong friends and companions
```

Fig. 8.1 Seven examples of word transcriptions produced by the DNN-HMM sMBR model comparing four feature types. ‘REF’ refers to the reference sentence (ground-truth sentence). Words in capital letters refer to the misrecognised words.

Figure 8.1 shows examples of lipreading results obtained from Eigenlips, DAE, DCT, and DTCWT. There are seven examples of word transcriptions produced by the DNN-HMM sMBR model. ‘REF’ refers to the reference sentence (ground-truth sentence). Words in capital letters are the misrecognised words. These show that results obviously vary between sentences. Some sentences are easier to predict than others. An error can be caused by a different word form such as *we are* and *WE’RE*, *remain* and *REMAINED*, *seem* and *SEEMS*. It can also be caused by a similarity in a viseme group such as *food* and *FOOT*, *snow* and *SO*. It could also become a completely different word such as *garage* and *ENJOY*, *he* and *POSITION*. It is not easy to observe the word confusion matrix because many errors propagate from previous errors, and sometimes many words are missing. Thus we investigate the error type and error position instead.

8.1 Analysis of error type and position

Here we investigate three error types and five positions. The error types are insertion (I), deletion (D) and substitution errors (S) as explained in Chapter 4.6.

```
REF:  this IS no assignment for a frivolous girl she assures him
HYP:  this ** no assignment for a frivolous girl she assures him
Eval:      D

REF:  INTERNAL national responsibility now A truism need not be documented
HYP:  NEITHER  national responsibility now * truism need not be documented
Eval: S                                    D

REF:  the tooth fairy forgot to come when roger's tooth fell OUT
HYP:  the tooth fairy forgot to come when roger's tooth fell AXIS
Eval:                                                        S

REF:  *** twenty nine exhibits received AWARDS
HYP:  THE twenty nine exhibits received WORRIED
Eval: I                                 S

REF:  FOR roast insert meat thermometer diagonally so IT DOES NOT   REST ON BONE
HYP:  BUT roast insert meat thermometer diagonally so ** **** THERE WAS  A  BOWL
Eval: S                                               D  D    S     S    S  S

REF:  the source IS KNOWN SO THERE IS    no necessity to remove insecticide residues
HYP:  the source ** ***** ** OWNER KNOWS no necessity to remove insecticide residues
Eval:            D  D     D  S     S

REF:  YOU'D THINK HER STOMACH WOULD'VE GOT   USED TO    IT  IN  THREE WEEKS
HYP:  ***** ***** *** ******* COCONUT  CREAM WITH THOSE WHO HIS WORK  WITHOUT
Eval: D     D     D   D       S        S     S    S     S   S   S     S
```

Fig. 8.2 Seven examples of word transcriptions produced by the DNN-HMM sMBR model on DAE. 'REF' refers to the reference sentence (ground-truth sentence), 'HYP' is a hypothesis (lipreading result), 'Eval' shows the type of errors.

Figure 8.2 shows seven lipreading results obtained from the DNN-HMM sMBR model on DAE features. Also shown is the error evaluation (Eval). The error positions are classified into five areas: (1) error on the entire length of a sentence, (2) error in the middle, (3) error at the beginning, (4) error at the end, (5) error at the beginning and the end. We can match these error positions to our lipreading results by identifying if errors occur in the predefined area. For example, the first sentence and the sixth sentence have errors occurred in the middle. The seventh sentence has errors in the entire sentence. The reason why we try to analyse these error positions is because we have an assumption about the complexity of the visual silence model. In silence speech area, where there is no sound, sometimes the lips do not close completely. Therefore, there is high possibility to find an error occuring at the beginning and/or the end of a predicted sentence.

Table 8.1 Sentence level error analysis

Sentence level analysis		#Utt	%
Total utterances		1666	100.00
Correct utterances		390	23.41
Error type	Insertion	215	12.91
	Deletion	960	57.62
	Substitution	1159	69.57
Error position	Entire sentence	291	17.47
	Middle	91	5.46
	Begin	333	19.99
	End	306	18.37
	Begin and end	255	15.31

Here we analyse 1666 utterances of lipreading results from the first-fold of SI. Table 8.1 shows the percentage of specific types and positions of the sentence errors. The first column reveals the category of error. The second column shows the actual number of utterances and the third column shows its percentage. The first row lists the total number of utterances and the second row shows the number of correctly predicted sentences. There are 23.41% of utterances that we recognise correctly. The rest of the utterances contain an error. The error type shows how many sentences contain that kind of error. It shows that the main type of error found in our lipreading system is substitution error. We found that almost 70% of predicted sentences contain substitution errors. Deletion errors are also a major issue that is found in nearly 60% of predicted sentences. The insertion errors are a minor issue in our lipreading result; we found only 12.91% of sentences that contain an insertion error.

The lower part of Table 8.1 shows the position of sentence errors (number and percentage). Let's start with the error positions that we assume might relate to the difficulty in predicting a visual silence. As we can see, the majority number of results have an error occurring at the beginning or the end or both positions which totally amount to 53.67% of utterances. There are only 5.46% of predicted results that found an error appearing in the middle. These numbers of position errors are an indicator that guides to a problem of modelling visual speech in the visual silence area.

Table 8.2 Word level error analysis

Word level error analysis Total reference words = 13809		#Words							
		Correct		Insertion err		Deletion err		Substitution err	
		Actual	%	Actual	%	Actual	%	Actual	%
Sum		7480	54.17	250	1.81	2507	18.15	3822	27.68
Correct sentence		2857	20.69						
Error position	Entire sentence	0	0	23	0.17	804	5.82	1351	9.78
	Middle	588	4.26	7	0.05	122	0.88	130	0.94
	Begin	1708	12.37	71	0.51	411	2.98	694	5.03
	End	1475	10.68	45	0.33	484	3.50	654	4.74
	Begin and end	852	6.17	104	0.75	686	4.97	993	7.19

Table 8.2 shows the error types and positions in the word level. In this table, we investigate the area that generates a large number of each error type. The columns present the number of correct words and the number of words in each type of errors. Each line refers to an error position. The error position in the entire sentence means that all predicted words are incorrect. There are 13809 words in total, with about 54% of them correctly recognised. The majority of errors are substitution errors, totally 27.68% with 9.78% of them found in the entire incorrect sentence and 16.96% of error words found at the beginning and/or the end of the sentence area. The deletion errors are also considerably high. As we can see, deletion errors cover 18.15% of words where 11.45% of them occur in the area of the beginning and/or the end of sentences.

The analysis of error type and position leads to a further investigation in visual silence modelling. It can be seen that the majority of error occurs at the beginning and/or the end of utterances rather than in the middle of utterances. To understand further, we carry a further investigation in this issue in Section 8.4.

8.2 Analysis of word length and word frequency

This investigation analyses errors in terms of word length and word frequency. We define three groups of word length and three groups of word frequency. The groups of word length are short words (one to four characters), medium-length words (five to nine characters), and long words (⩾10 characters). The groups of word frequency are low frequency words (occur one or two times), middle frequency words (occur between three to nine times), and high frequency words (occur ⩾10 times).

Table 8.3 The statistical of word length and word frequency between correct and incorrect recognised words

Word frequency	total	correct			incorrect		
		short	medium	long	short	medium	long
high frequency word (10 times)	48.66	18.24	3.08	0.06	25.46	1.81	0.01
middle frequency word (3-9 times)	30.70	3.89	12.87	1.76	4.08	7.36	0.74
low frequency word (1-2 times)	20.64	1.24	7.50	1.83	1.68	7.24	1.15

Table 8.3 compares the statistical information of word length and word frequency between correct and incorrect recognised words. Most of the high-frequency words are short and harder to recognise than longer words. We get 25.46% false recognition in the short words that occur frequently. We tend to get more correct results if the words are longer than four characters even if they do not occur often.

Table 8.4 Examples of challenging to predict words and easy to predict words

Examples of difficult to predict words (< 10% correct)		
high frequency word	middle frequency word	low frequency word
A ->(ah)	YET ->(y eh t)	SITUATION ->(s ih ch uw ey sh ah n)
IT ->(ih t)	TELL ->(t eh l)	UNDERSTANDINGLY ->(ah n d er s t ae n d ih ng l iy)
I ->(ay)	DESSERT ->(d ih z er t)	ANALYSIS ->(ah n ae l ah s ah s)
IN ->(ih n)	ORDER ->(ao r d er)	BOARDINGHOUSES ->(b ao r d ih ng hh aw s ah z)
IS ->(ih z)	DOCTOR ->(d aa k t er)	LEATHER ->(l eh dh er)
YOU ->(y uw)	SURE ->(sh uh r)	FOOLING ->(f uw l ih ng)
HE -> (hh iy)	MUSTARD ->(m ah s t er d)	INSTANTANEOUS ->(ih n s t ah n t ae n iy ah s)
DOES ->(d ah z)	CHURCH ->(ch er ch)	RESULT ->(r ah z ah l t)
THEM ->(dh eh m)	MARINE ->(m er iy n)	ERRORS ->(eh r er z)
AT -> (ae t)	HOUSE ->(hh aw s)	HINT ->(hh ih n t)

Examples of easy to predict words (> 90% correct)		
high frequency word	middle frequency word	low frequency word
SHE ->(sh iy)	SHORTAGE ->(sh ao r t ah jh)	PANELIZATION ->(p ae n ah l ah z ey sh ah n)
CAN ->(k ae n)	BECOME ->(b ih k ah m)	INTELLIGIBLE ->(ih n t eh l ah jh ah b ah l)
YEAR ->(y ih r)	STEEP ->(s t iy p)	ILLUMINATING ->(ih l uw m ah n ey t ih ng)
WOULD ->(w uh d)	BOB ->(b aa b)	MERCILESSLY ->(m er s ah l ah s l iy)
OILY ->(oy l iy)	REDWOODS ->(r eh d w uh d z)	VIEWPOINT ->(v y uw p oy n t)
SMALL ->(s m ao l)	OUTDOORS ->(aw t d ao r z)	GIGANTIC ->(jh ay g ae n t ih k)
BROTHER ->(b r ah dh er)	BRIGHT ->(b r ay t)	WARDROBE ->(w ao r d r ow b)
MAKES ->(m ey k s)	WIRE ->(w ay er)	GARBAGE ->(g aa r b ih jh)
SELDOM ->(s eh l d ah m)	LET ->(l eh t)	CONFIRM ->(k ah n f er m)
OVER ->(ow v er)	INCREASES ->(ih n k r iy s ah z)	OPEN ->(ow p ah n)

Table 8.4 gives example words which are relatively easy or difficult to lipread. There are two effects at play: firstly there is homophene or the confusion of words because they have identical shapes on the lips and secondly there is the observation that certain sounds are more visible on the lips than others. Words such as “brother” and “makes” have bilabials at the start of the word which makes them easier to spot than “at” or “he”. Longer words are easier to lipread than shorter ones, and the homophene effect means the classifier has to guess from a considerable number of alternatives (which might explain why some of the low-frequency difficult words still contain bilabials –

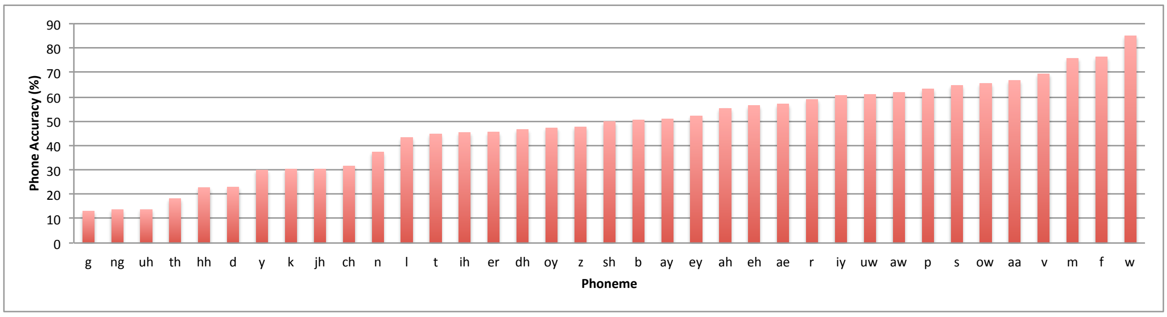


Fig. 8.3 The phoneme accuracy ranging from the lowest to the highest accuracy.

albeit weakly enunciated bilabials such as found in “marine” or “mustard”). Table 8.4 is also essentially an illustration of various effects - the more frontal-labial is, the easier it is to lipread; the more homophemes, the worse it is to lipread.

Additionally, in Figure 8.3, we plot the phoneme recognition performance by ranging from the worst to the best of phoneme accuracy. There is a relation between phoneme accuracy and the place of articulation in which it related to the level of visibility from the lips. The phoneme accuracy gives an idea to understand why some words are harder to predict than others. We suspect that many of the hard-to-predict words, especially short words, usually start with a vowel sound. This is because they contain less hints than words starting with a consonant, therefore it is obviously harder to distinguish between similar words such as A, I, IN and IT. However, words starting with some consonants, which are invisible on the lips, are also difficult to predict. For example, words started with /hh/ and /d/.

8.3 Analysis of speaker accuracy

As usual with lipreading systems the identity of the talkers can be very significant. As shown in Figure 8.4, we plot word accuracy between SI and SD from 56 speakers, word accuracies are in a wide range, ranging from 20% to 80%. It can be seen that there is a rough linear relationship between word accuracy of SI and SD (correlation coefficient $R = 0.82$). An easy-to-lipread speaker tends to get high word accuracy even if their data was unseen. In contrast, a hard to lipread speaker got low word accuracy even if some part of their data is in training. This suggests a link to the speaker identity effect that still exists in the data although we applied multiple transformations to handle it.

This section attempts to analyse factors from an individual speaker that affect their lipreading performance. Here we conduct a test to study whether word accuracy of

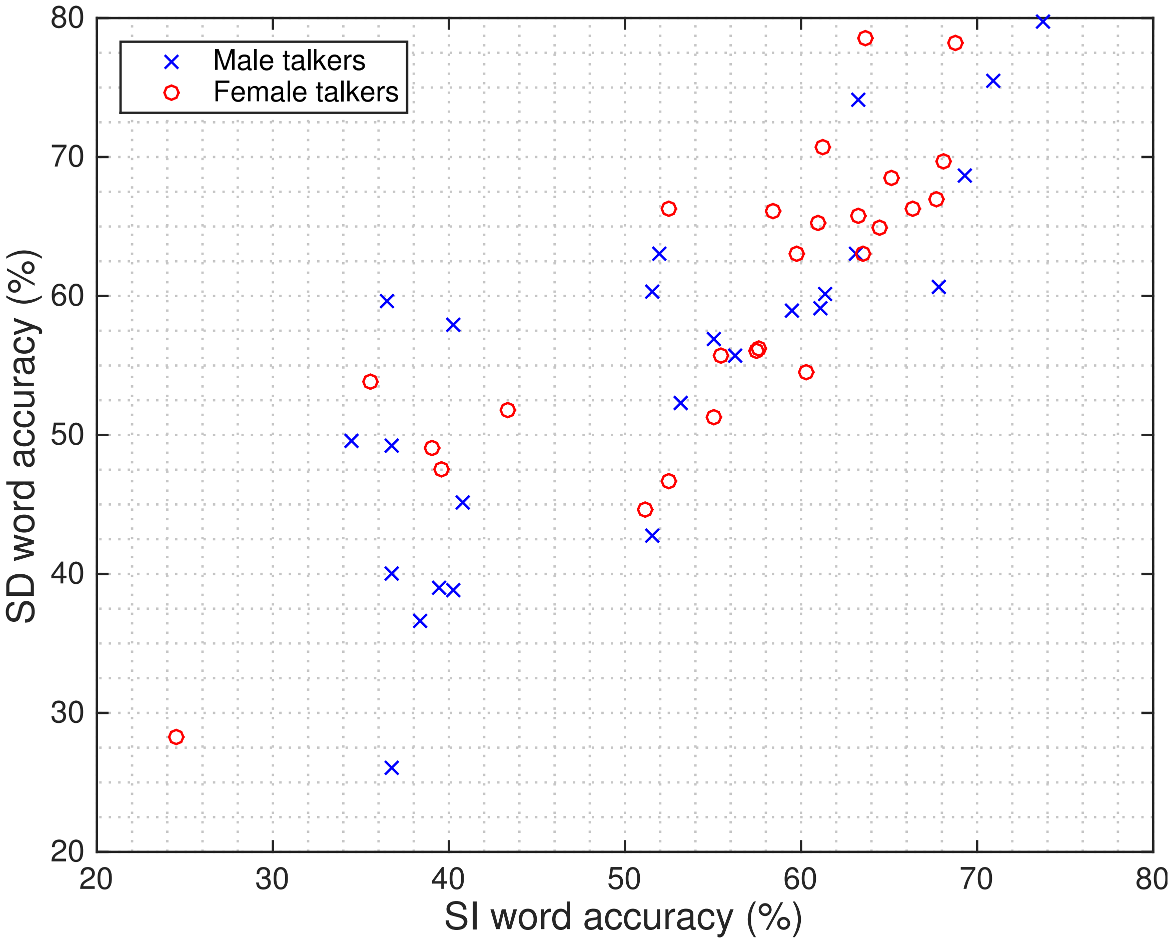


Fig. 8.4 Word accuracy in SD vs SI for a variety of talkers.

each speaker is correlated with speaking rate, age, gender, visual dynamics and visual energy. We investigate the correlation between these factors and SI word accuracy. We also determine the level of significance of each factor. We test the null hypothesis $H0 : R = 0$ where R is the correlation between word accuracy and speaking rate, word accuracy and gender, word accuracy and age, word accuracy and visual energy, word accuracy and visual dynamics. Speaking rate is the number of syllables per second. The mean of speaking rate of TCD-TIMIT corpus is 5.14 syllables per second. Visual dynamics refers to the rate of change of the lip signal while speaking. The visual dynamics of each speaker is calculated by averaging the gradient magnitude of the DAE features over time. The visual energy of each speaker is the mean energy of their DAE features.

Among many factors, results in Table 8.5 show that there is a significant correlation between speaker accuracy and the visual dynamics ($p < 0.05$). There is no significant

Table 8.5 Level of significance for the test $H0 : R = 0$. t is modeled by the t-student distribution with DF degrees of freedom.

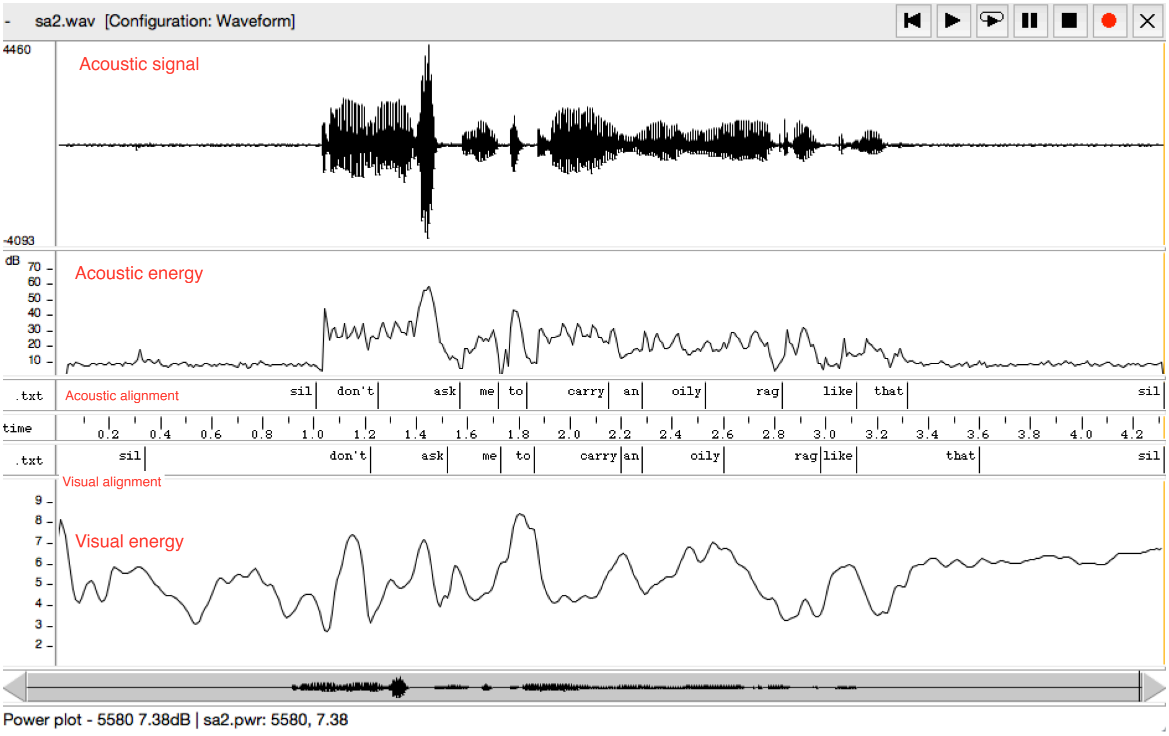
Speaker factors	R	t	DF	$p(\|t\|)$
word accuracy vs. speaking rate	-0.19	-1.4221	54	0.1607
word accuracy vs. gender	0.18	1.3447	54	0.1843
word accuracy vs. age	0.12	0.8882	54	0.3784
word accuracy vs. visual energy	-0.26	-1.9787	54	0.0530
word accuracy vs. visual dynamic	0.34	2.6568	54	0.0104

correlation between word accuracy and other factors. In particular, we have no evident to support the conventional wisdom which is that women are easier to lipread than men. A simple explanation of Table 8.5 might be that it is easier to lipread speakers that make high-energy motions in their mouth regions.

8.4 Investigation of visual silence

Visual silence is the visual signal present during acoustic silence. Visual silence strongly relates to speech co-articulation where the lips are getting into position for the next sound. Indeed the function of silences between words is to allow for breaths and for the articulators to reposition themselves in time for the next word. So lip movements in audio silence are caused by the future production of sound.

Our analysis in error position shows that 53% of the total sentence error rate occurs either at the start or end of a sentence due to lip movements of visual silence. Before a talker utters a sentence, at the sentence start, the talker has moved their lips once to take a breath and again to position their lips to wait for the ready-to-speak position. We analyse delays between acoustic and visual alignment in the TCD-TIMIT training data and found that there are 200-600 millisecond shifts from a starting point of the sound. Several observations reported that the audio lags behind the visual within the range of 100 to 300 milliseconds without disrupted speech perception [154, 93, 23]. However, the mismatch in the auditory and visual alignment in our training data is longer than that. Thus, the start of the breath generates extra movement in the lip region. At the end of a sentence, this process has happened again to move the lips back to the original shape and position which is taking another 200-300 milliseconds. The problem occurs when we train a model with a single silence model to handle all of this visual silence process. Our model can deal with stationary lip signals when the lips do not move. The rest of the process such as lip opening to take a breath and return to the original position causes errors due to the high energy motion of the lip signal.



Power plot - 5580 7.38dB | sa2.pwr: 5580, 7.38

Fig. 8.5 The difference between acoustic and visual alignments.

When this unexpected visual silence occurs in the model training, it causes false sentence alignment even with the training utterances. Figure 8.5 shows time alignment created by an acoustic signal and visual signal. The mismatch in time alignment between audio and visual is obviously seen at the start/end of the sentence between silence and first/last word. Even though the breath can also be observed via the acoustic signal in the same position in visual, it has very low energy. In visual speech, in contrast, mouth opening or closing causes significant temporal changes over frames.

To see this problem more clearly, we draw the visual alignment word boundary onto the video frame which we show in Figure 8.6. Looking at the figure we can see the first word *"don't"* in the magenta label. It is quite unrealistic that a monosyllabic word takes longer than 25 frames covering approximately 833 milliseconds. The issue comes from mislabeling of the visual silence area. More specifically, the mislabeling has happened due to the mouth opening to take a breath preparing to speak. This problem directly affects words at the start and the end of an utterance both in training and testing and still needs to be solved.

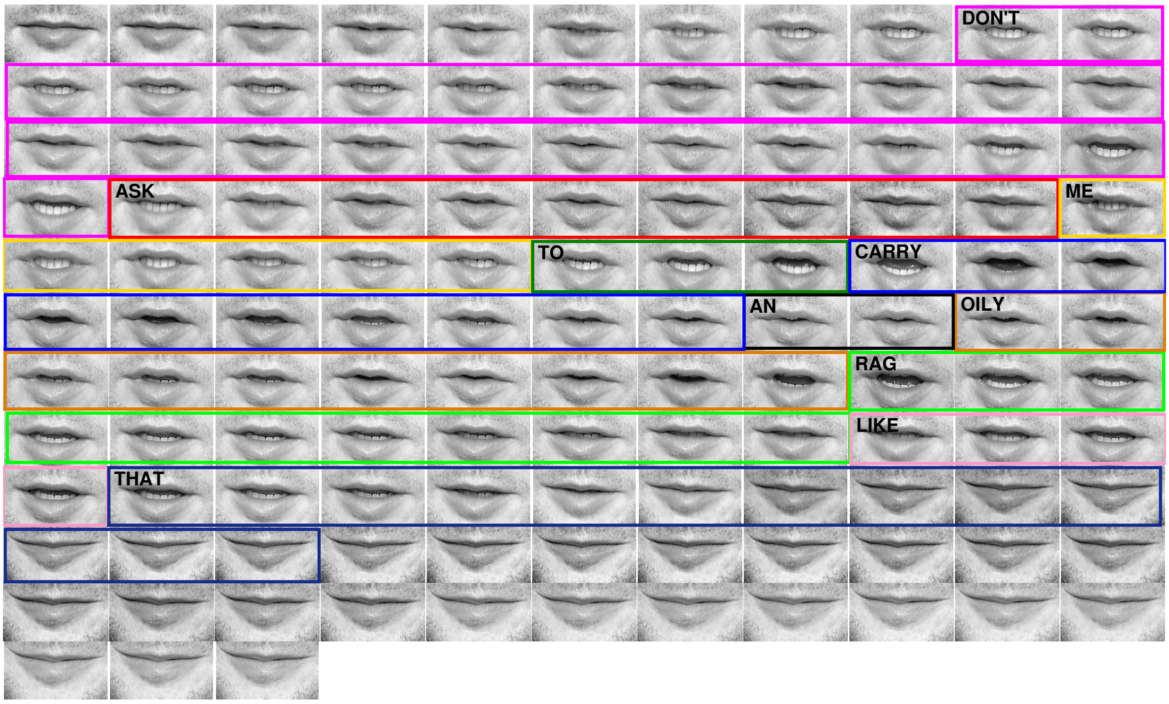


Fig. 8.6 Example of visual speech alignment of a sentence *"Don't ask me to carry an oily rag like that"* shown in word level. The word alignment shown here is obtained from visual speech DNN-HMM via sMBR model trained on deep autoencoder (DAE) features. Colour-boxes indicate word boundary while no box indicates no word or a silence phone.

Although the DNN-HMM is better at modelling a complicated visual signal compared to a conventional GMM-HMM, it is still unable to model the high variation that occurs within visual silence. As a consequence, word recognition results at the start and end of sentences are more likely to be incorrect compared to words in the middle. Furthermore, the misalignment in visual silence effect to the first and the last word that leads to unreliable modelling phonetic classes in that particular area. Our assumption is that there would be a further improvement if we can eliminate the visual silence problem. To support this idea, we reproduce the visual speech model using modified silence features where we remove the silence phone at the end of all TCD-TIMIT sentences. The alignment of the silence phone is obtained from the acoustic forced alignment. The results are shown in Figure 8.7.

Not surprisingly, removing the silence has benefits to system performance. This is seen in Figure 8.7, where the model trained on the modified silence yields better word accuracy than the original feature. We detailed the improvement and reduction of errors in Table 8.6. There is a 5.59% absolute increase in word accuracy which

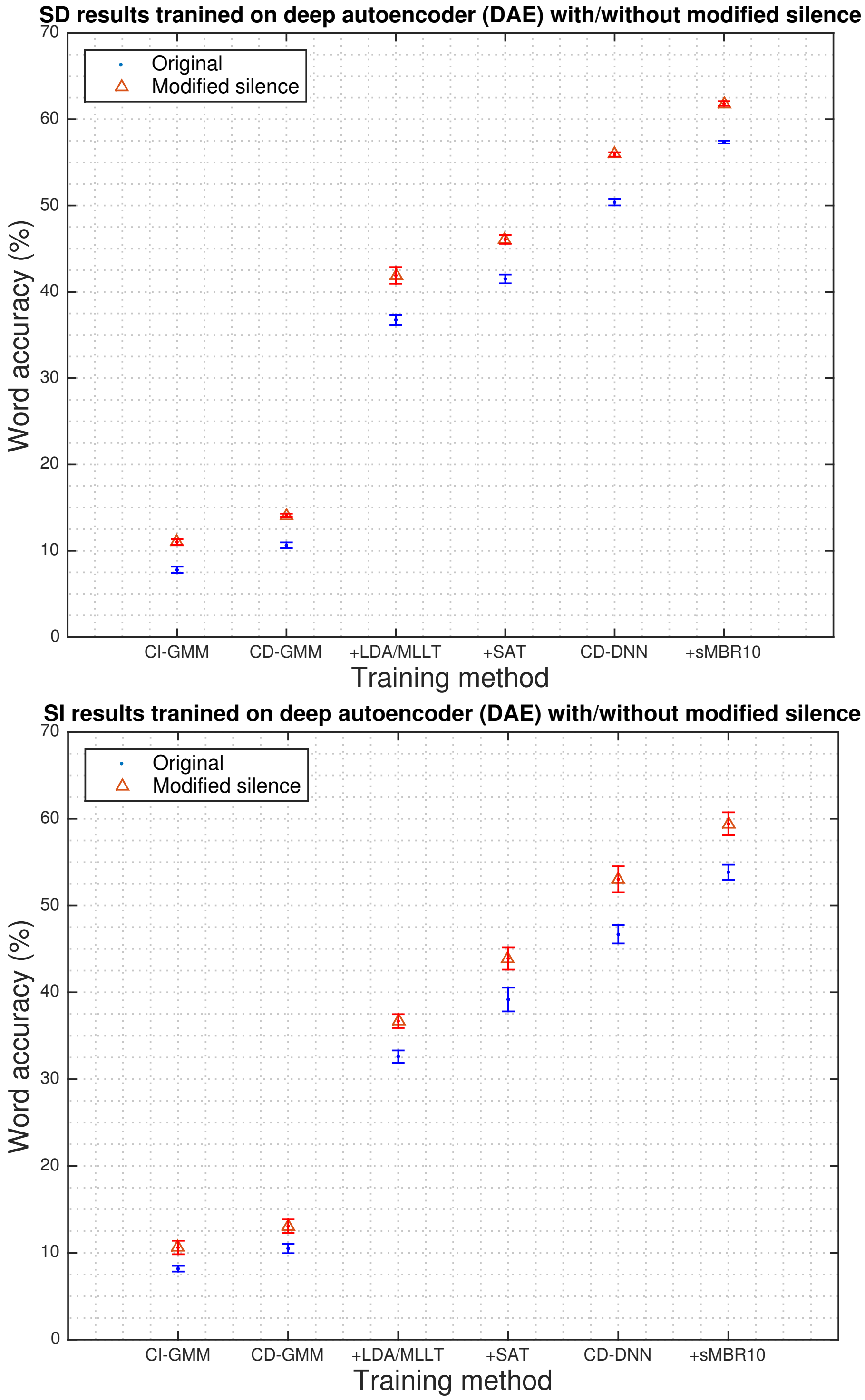


Fig. 8.7 Word accuracy (%) of speaker dependent lipreading (top) and speaker independent lipreading (bottom) comparing between original data (blue) and the modified silence data (red) as a function of model training methods.

Table 8.6 Word accuracy (%) and errors of speaker independent lipreading trained on DAE features with/without silence modification. (Corr: word correction, Sub: substitution error, Del: deletion error, Ins: insertion error, Err: all error and S.Err: sentence error)

Model	Feature	Silence	%Word acc	Corr	Sub	Del	Ins	Err	S.Err
DNN+SMBR10	DAE	original	53.83	55.87	27.90	16.23	2.03	46.17	75.17
	DAE	modified	59.42	60.73	24.07	15.13	1.33	40.60	68.30
Absolute change			+5.59	+4.87	-3.83	-1.10	-0.70	-5.57	-6.87
Relative change			+10.38	+8.71	-13.74	-6.78	-34.43	-12.06	-9.14

Table 8.7 Sentence level error analysis

Modified silence (sentence level error analysis)		#Utt Before	 After
Total		1666	1666
Correct		390	497
Error type	Insertion	215	185
	Deletion	960	908
	Substitution	1159	1046
Error position	Entire sentence	291	247
	Middle	91	109
	Begin	333	364
	End	306	248
	Begin and end	255	202

obtained almost 60% word accuracy in speaker independent lipreading when we train the model on the modified silence utterances. Indeed, we can obtain high relative error reduction for all three types of errors especially insertion and substitution. Also we observe error reduction in sentence level as shown in Table 8.7. However, we chose not to cut the silence phone at the start of a sentence since it still contains a clue to predict phonemes in the first word. We believe if we could identify the breath at the beginning of an utterance and remove it, we could further improve the word accuracy of the system.

Although we see improvement in overall word accuracy, there is some negative effect of removing a silence area. Figure 8.8 illustrates a positive and a negative effect of eliminating silence in terms of absolute change in word accuracy of each speaker. We plot the absolute change of word accuracy of speaker dependent against speaker independent results. Word accuracy has been improved in most cases in the speaker independent scenario, but word accuracy has also degraded in many cases in the speaker dependent scenario.

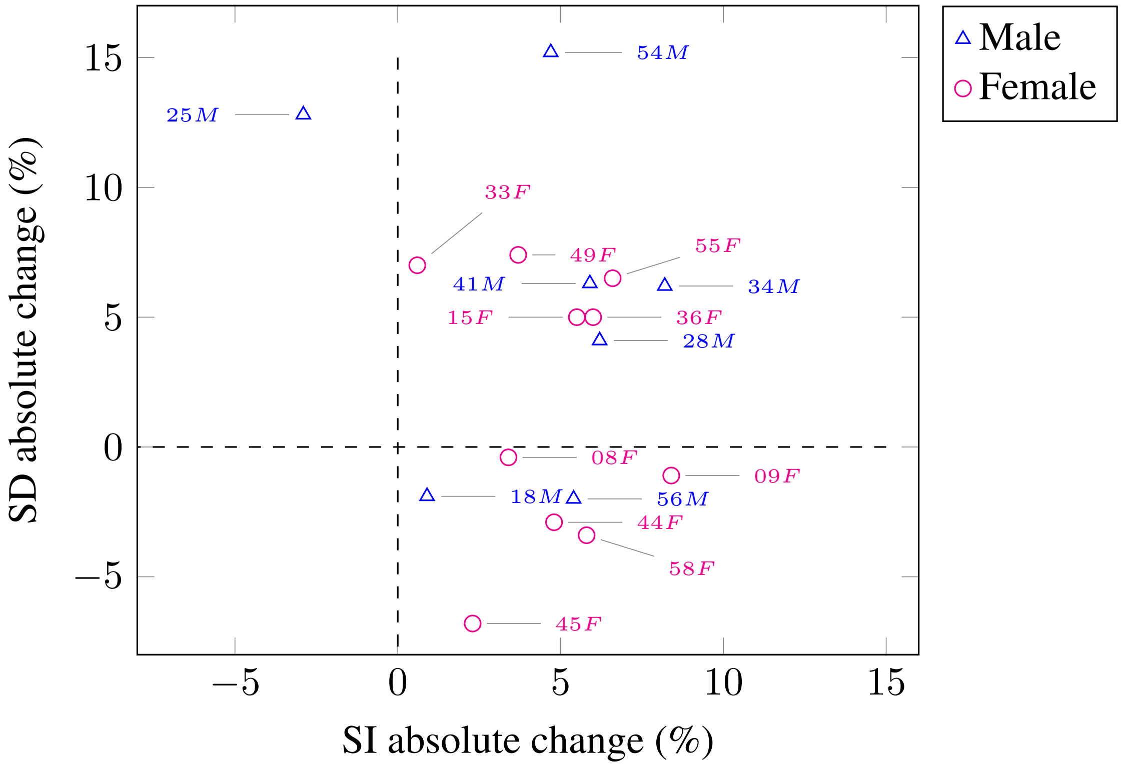


Fig. 8.8 Positive and negative effect of modifying visual silence in terms of absolute change in word accuracy of each speaker.

8.5 Conclusion

This Chapter presents results analysis of transcriptions generated from our best lipreading system. We found that long words are easier to lipread than short words. A word carrying more visibility-on-the-lip sound has higher chance to be predicted correctly than a word with invisible sound originating inside the mouth.

In terms of speaker identity, we found that the word accuracy of a talker has a strong positive correlation between their SI accuracy and SD accuracy. We found a significant positive correlation ($p < 0.05$) between word accuracy and the rate of change in lip signal which we call visual dynamics. The high volume of visual dynamics means a talker open their mouth wider.

Finally, we observe that lipreading transcriptions contain lots of substitution and deletion errors. These errors obviously occur mostly at the beginning and end of a sentence due to the misalignment of the silence model. We conduct a preliminary investigation on visual silence. We eliminate the silence area at the end of a sentence and retrain the model. We get around 5% improvement in word accuracy in both SD and SI scenarios.

Chapter 9

Summary and future work

9.1 Thesis summary

To return to the research question: can we simply employ the DNN-HMM hybrid approach, which is gaining usage in acoustic recognition, to improve visual speech model and computer lipreading? The answer is yes we can. We successfully built a computer lipreading system for the LVCSR task using the DNN-HMM hybrid approach proposed in speech recognition systems. Our approach is to tackle challenges in lipreading presented in Chapter 1. We built a word recogniser based on the phoneme unit. We use 30-dimensional deep autoencoder (DAE) features extracted from gray-scale lip ROIs. The features are then pre-processed via LDA-MLLT and fMLLR feature transformation methods. We train visual speech models on a six-hidden layer DNN-HMM optimised via the cross-entropy criterion and the sMBR sequence distriminative training method. In recognising a 6000-word vocabulary, we achieve word accuracy of 61.82% in a speaker dependent scenario and 59.42% in a speaker independent scenario.

We also attempt to answer how does a DNN-HMM work better than a GMM-HMM? and why? Conventional systems have shown speaker independence to be a challenge, here with a novel DNN-HMM architecture we have reduced the speaker effects. We speculate that the success of the DNN has likely to do with its ability to better model the effects of co-articulation which is a well-known bugbear of human and machine lipreaders. Furthermore, we found that visual speech features extracted from lip ROIs retain most of the variation regarding speaker identity information. Employing multiple feature transformation methods to pre-process features can reduce the speaker identity effect and enhance the performance in the speaker-independent scenario. Indeed, each DNN layer works as a feature transformation and also minimises this effect.

We observed evidence where a deeper model obtains higher word accuracy than a shallow model (one hidden layer) when the number of nodes in total was constrained to be the same. This evidence is further investigated and explained by feature space visualisation. We demonstrated that the speaker dependency effect reduces throughout feature pre-processing steps and DNN layers.

Our contributions of each chapter can be summarised as follows:

In Chapter 5, we incorporate visual information in an audiovisual speech recognition system. We have explored the use of DNNs in visual and audiovisual speech recognition. The experimental results obviously demonstrated that the DNN techniques, even in standard settings, can beat the conventional GMM-HMM speech recogniser (both the unimodal and bimodal speech recognition system). In a speaker-dependent visual speech (lipreading) experiment, DNNs gave 85% word accuracy, a huge improvement over the baseline HMM performance of 33%. Moreover, we found that DNNs improved the robustness of audio-only and audiovisual recognition tasks by approximately 10 and 12dB respectively. It is interesting to see that the performance is improved for matched-condition recognisers, where the performance of the audiovisual system closely followed the visual-only system. This highly suggests that DNNs have the power to model complex signals in very challenging conditions (lower than 0dB) if we provide enough information. The result of matched-condition recognisers is the primary motivation for us to start explore more on lipreading technology based on DNNs.

Chapter 6 presents the development of LVCSR computer lipreading on the TCD-TIMIT corpus. We have built a successful lipreading system using DNNs and sequence discriminative training. Comparing our result with a conventional HMM, we see that performance has increased from around 25% word accuracy to around 48% in speaker independent mode. Looking in more detail, large improvements are obtained using fMLLR, the DNN rather than a GMM, some temporal stacking and the use of sequence discriminative training. The best word accuracies are 52.88% in speaker dependent mode and 48.71% in speaker independent mode obtained on the phoneme unit rather than the viseme unit. We have added more evidence to the argument that phoneme classifiers can outperform those of visemes. Based on evidence showing an improvement in word accuracy from the reduction in homophene words in a pronunciation dictionary, we suggest that phonemes are the current best class labels for lipreading.

In Chapter 7, we show that the original space of visual features is influenced by speaker identity. This problem induces a difficulty to train visual speech classification to match to a linguistic unit (phoneme/viseme) even with DNN models. We apply several feature processing steps that are relevant to reduce this effect and gain speaker-independent lipreading accuracy. We achieved 53.83% word accuracy in the speaker independent scenario using DAE features. Furthermore, we illustrate benefits of DNN models in term of providing extra feature transformations inside their hidden layers. Thus, the deeper layers of the representation tend to be more abstract (less affected by the environment).

Chapter 8 presents a results analysis of transcriptions generated from our best lipreading system. We investigate the visual silence problem where we eliminate the silence area at the end of a sentence and retrain the model. We get around 5% improvement in word accuracy in both SD and SI scenarios. Our best system for the large vocabulary task yields 61.82% word accuracy in SD and 59.42% word accuracy in SI on the TCD-TIMIT dataset.

In this work, we confirm that computer lipreading an unseen speaker in LVCSR is possible. According to advances in machine learning, more specifically deep learning and a high growth rate of video data, we expect that computer lipreading is soon to be a viable product. There are some limitations in our work discussed in the following section. We also plan further investigation for future work to enhance lipreading systems discussed a later section.

9.2 Limitations

In this thesis, we evaluated lipreading in a studio environment which is the most simple condition concerning challenges to video processing. Also, this system still works as an offline system where we have enough time to process complex features and use a two-pass decoder. Indeed, the decoding process starts after receiving the whole utterance, because the feature pre-processing steps are computed in an utterance basis.

A limitation in terms of the training method is that DNN-HMM training strongly relies on a state level alignment and the fMLLR features generated from a GMM-HMM system. Therefore, the limitations of the GMM-HMM system somehow transfer to the DNN-HMM system. It can be seen that the DNN-HMM is not yet an ultimate answer for every problem since it cannot handle the visual silence model properly. Although a deep neural network visual speech model has more potential modelling a complex signal, this model still needs correct labels to train. And the quality of labels heavily relies on the performance of a GMM-HMM.

9.3 Future work

So far, this thesis shows that computer lipreading an unseen speaker for LVCSR is possible. The visual speech model via DNN-HMMs improves lipreading accuracy over the GMM-HMM system in both speaker dependent and speaker independent scenarios.

We also plan further investigation for future work to enhance lipreading system as follows:

- We see a wide range of word accuracy among different talkers due to the effect of speaker identity. We can improve the DNN-HMM by incorporating a speaker-informed feature such as i-vectors into the input layer. In ASR, this technique is known as a speaker adaptation of DNN acoustic models [129, 70, 69] that offers 5-10% relative improvement in ASR performance.

- In terms of a visual speech unit, there are other choices of speech units that can be used to model visual speech, for example, syllable, subsyllable such as onset-rime [75], and dynamic visemes [146]. According to a study by Chandrasekaran et al. [23], visual speech moves at a similar time scale as the syllabic level of speech and could provide a cue to identify a syllable segment. Therefore, we can consider a longer unit to model visual speech instead of a short unit as a phoneme.

- In robust speech recognition system, an acoustic model is trained on thousands of hours of speech data. One technique to increase training data size is to use data augmentation. Generative Adversarial Networks (GANs) [45] have been a powerful method to generate images in different styles. We could apply GANs to augment the visual speech corpus by transforming an appearance of a speaker and use this data as an additional training set.

- There are many advanced techniques in deep learning that we could also consider, for example, TDNN, LSTM, and the end-to-end approach.

Appendix A

Optimising GMM-HMM visual speech modelling

The idea of tuning GMM-HMM parameters is inspired by Joy et al. [65], in work in TORGO [125] for a Dysarthric speech recognition task. They trained an acoustic model for dysarthric speech recognition system using GMM-HMMs and DNN-HMMs with careful tuning of speaker-specific parameters. They reported a new state-of-the-art result with a relative WER reduction of 17.62% from the baseline system trained on a more complex model. There is an assumption here that dysarthric speech is similar in some way to visual speech [60].

This set of experiments aims to examine the optimal parameters of visual speech modelling via the conventional GMM-HMM training on the Kaldi toolkit. Since Kaldi recipes are developed for acoustic speech modelling, most of the parameters are tuned on acoustic features with a large speech dataset. In contrast, our available training dataset is small and the visual speech signal is complex. Thus, we fine-tune parameters in GMM-HMM visual speech modelling most suitable for each feature.

Training the GMM-HMM visual speech model involves these steps: (1) context-independent training; (2) context-dependent training, (3) context-dependent training with LDA/MLLT, (4) context-dependent training with speaker adaptive training. The model obtained in the last step often outperforms the models trained from the previous steps.

Effect of adding delta and normalisation

This experiment investigates the effect of feature processing for context-independent GMM-HMM (CI-GMM) visual speech modelling. CI-GMM is used as an initialising step for context-dependent GMM-HMM. It also generates phonetic-state labelling for the entire training set.

Here we explore feature processing parameters including the Delta windowing and feature normalisation. The default setting uses two context-windows (w=2) for calculating the velocity (Δ) and the acceleration (ΔΔ) which is a standard setting for acoustic modelling. Feature normalisation is optional since acoustic speech has many choices to normalise features for example, vocal tract length normalisation (VTLN). For visual features there may be a large scale difference between the dimensions of the feature vector. Therefore, we apply feature mean and variance normalization (FMVN) to rescale the visual features (Chapter 4.4.2).

In this experiment, we train the CI-GMM with the standard three-state left-to-right topology. We set the maximum number of Gaussian components to 1000. There are ten variations of feature processing. The variations include using a raw static feature, adding FMVN, adding derivatives up to third order, using w context window to calculate the derivative, where $w = 1, 2, 3, 4, 5$.

Table A.1 Word accuracy (%) of speaker dependent lipreading using CI-GMMs.

Feature processing	Speaker dependent (SD)			
	Eigenlips 30 dim	DAE 30 dim	DCT 44 dim	DTCWT 66 dim
Raw + Δ + ΔΔ (default $w = 2$)	6.78 ±0.20	7.63 ±0.18	3.90 ±0.05	-1.82 ±0.14
Raw	2.89 ±0.14	3.65 ±0.13	2.23 ±0.09	1.74 ±0.09
Raw + FMVN	3.59 ±0.19	4.29 ±0.20	2.38 ±0.08	2.13 ±0.08
Raw + FMVN + Δ	9.95 ±0.61	8.38 ±0.28	5.29 ±0.25	2.22 ±0.16
Raw + FMVN + Δ + ΔΔ ($w = 1$)	9.62 ±0.24	7.81 ±0.40	4.97 ±0.38	1.40 ±0.26
Raw + FMVN + Δ + ΔΔ ($w = 2$)	8.26 ±0.17	7.65 ±0.45	4.93 ±0.10	0.92 ±0.38
Raw + FMVN + Δ + ΔΔ ($w = 3$)	8.44 ±0.40	7.79 ±0.37	5.10 ±0.13	0.98 ±0.31
Raw + FMVN + Δ + ΔΔ ($w = 4$)	9.39 ±0.71	8.78 ±0.29	4.76 ±0.05	1.53 ±0.06
Raw + FMVN + Δ + ΔΔ ($w = 5$)	8.57 ±0.21	8.54 ±0.30	4.67 ±0.01	1.38 ±0.07
Raw + FMVN + Δ + ΔΔ + ΔΔΔ	7.41 ±0.28	6.80 ±0.34	4.65 ±0.32	-0.10 ±0.01

Table A.1 shows the mean word accuracy with ±1 standard error for the speaker-dependent (SD) system. The column on the left presents ten variations of feature processing. The raw feature refers to the original form of a static feature and (+) means augmenting the feature vector. The FMVN is equivalent to z-score normalisation, Δ

Table A.2 Word accuracy (%) of speaker independent lipreading using CI-GMMs.

Feature processing	Speaker independent (SI)			
	Eigenlips 30 dim	DAE 30 dim	DCT 44 dim	DTCWT 66 dim
Raw + Δ + $\Delta\Delta$ (default $w = 2$)	6.57 ±0.10	7.04 ±0.28	3.47 ±0.08	-1.62 ±0.48
Raw	2.53 ±0.04	3.21 ±0.21	2.11 ±0.07	1.99 ±0.13
Raw + FMVN	3.01 ±0.12	3.82 ±0.25	2.47 ±0.02	1.58 ±0.48
Raw + FMVN + Δ	8.44 ±0.41	8.16 ±0.55	4.55 ±0.07	2.19 ±0.39
Raw + FMVN + Δ + $\Delta\Delta$ ($w = 1$)	9.03 ±0.56	7.60 ±0.46	4.87 ±0.09	1.23 ±0.08
Raw + FMVN + Δ + $\Delta\Delta$ ($w = 2$)	7.89 ±0.19	7.40 ±0.76	4.64 ±0.06	0.87 ±0.30
Raw + FMVN + Δ + $\Delta\Delta$ ($w = 3$)	8.10 ±0.21	8.17 ±0.33	4.84 ±0.47	0.69 ±0.61
Raw + FMVN + Δ + $\Delta\Delta$ ($w = 4$)	8.37 ±0.35	7.82 ±0.47	4.78 ±0.26	1.32 ±0.33
Raw + FMVN + Δ + $\Delta\Delta$ ($w = 5$)	7.98 ±0.03	7.85 ±0.34	4.72 ±0.22	1.49 ±0.32
Raw + FMVN + Δ + $\Delta\Delta$ + $\Delta\Delta\Delta$	6.77 ±0.06	6.71 ±0.41	4.84 ±0.56	0.09 ±0.35

and $\Delta\Delta$ are usually known as velocity and acceleration respectively. w is the size of window for computing Δ. Word accuracy for the four feature types is shown in the columns on the right. Looking at the first row (default setting), the best word accuracy is 7.63%, obtained using DAE features; followed by 6.78%, obtained by Eigenlips; and 3.90% obtained by DCT. The word accuracy of lipreading trained on the DCTWT features is down to -1.82%. The negative number indicates large numbers of insertions (detailed in Chapter 4.6). Applying FMVN increases word accuracy in all features. Comparing systems without FMVN ($\Delta + \Delta\Delta$, $w = 2$) and with FMVN (FMVN + $\Delta + \Delta\Delta$, $w = 2$), the latter yield +1.48% for Eigenlips; +0.02% for DAE; +1.03% for DCT; and +2.74% for DTCWT. The best word accuracy of SD CI-GMM is 9.95% obtained from Eigenlips features with FMVN + Δ. Followed by DAE features, there is 8.74% word accuracy obtained from FMVN + $\Delta + \Delta\Delta$ ($w = 4$).

Table A.2 shows the mean word accuracy of speaker-independent (SI) systems obtained from four features with ten variations of feature processing. The results between SD and SI are similar, although the SI results are worse than SD, as SI is regarded as a tougher problem. The best result of SI is 9.03% also obtained by Eigenlips, but obtained for different feature processing which is FMVN + $\Delta + \Delta\Delta$ ($w = 1$).

These results show that improvements due to feature processing tend to be maintained irrespective of whether the system is trained in the SI or SD configurations. The results suggest that adding dynamic information is useful since the information provided in a static feature has insufficient cues to predict a linguistic unit. The support evident is the increasing of word accuracy from 2-5% in each feature by adding Δ. This result is consistent with the previous work reported on TCD-TIMIT [49]. Second, setting the window size $w = 2$ is not the optimal solution in lipreading. We found that the word accuracy can still increase when we change the window size, i.e. at $w = 1$ and $w = 3$. Third, adding more dynamic features by using the third-order of derivative does not improve accuracy. Finally, applying FMVN enhances word accuracy.

To select feature processing for CI-GMMs and context-dependent GMM-HMMs in the next step, we choose $\Delta + \Delta\Delta$ at $w = 3$ with FMVN. We observe that using FMVN and adding up to second-order derivatives significantly outperforms the default setting, but changing the size of the window does not gain very much. Note that, $w = 3$ is equivalent to using information from the actual video frame after applying feature interpolation for upsampling 29.97fps to 100fps. The next experiment moves on to the context-dependent GMM-HMM system.

Tuning of context-dependent tied-states clustering

The previous experiment demonstrates the effect of feature processing for CI-GMMs. The best word accuracy of the SD system is 9.95%, and of the SI is 9.03% obtained using the Eigenlips features. Here we use a context-dependent model to improve word accuracy. In this experiment, we train a context-dependent GMM (CD-GMM). We construct a triphone-context model by adding context information over three phonemes: the middle phone, and the phonemes to the left and right.

An important parameter in this step is to set the number of triphone tied-state clusters as described in Section 4.5. This experiment determines the proper number of the tied-state clusters for CD-GMM training. In Chapter 6, we set the number of the tied-states clusters to 2000. However, this number was tuned for the Wall Street Journal (WSJ) speech corpus that is a large speech corpus containing 400 hours. In this experiment, we reduce the number of context-dependent tied-states from 2000 to between 50 and 1500. It is important to note that these numbers refer to the maximum number of the tied-state cluster. The actual cluster size can be slightly less depending on the variance of each feature. Here we use $\Delta + \Delta\Delta$ and $w = 3$ for feature processing and set the maximum number of Gaussian component to 15000.

Table A.3 compares the word accuracy of SD lipreading among four types of features obtained by CD-GMMs with seven different sets of tied-state clusters. The best result is 13.62% obtained from a CD-GMM with 100 state-clusters trained on DAE features.

Table A.3 Word accuracy (%) of speaker dependent lipreading using context dependent GMMs (tri-phone models) with various numbers of tied-states.

# Context dependent states	Speaker dependent (SD)			
	Eigenlips 30 dim	DAE 30 dim	DCT 44 dim	DTCWT 66 dim
50	10.28 ±0.17	12.26 ±0.32	4.27 ±0.14	-4.24 ±0.18
100	12.13 ±0.36	13.62 ±0.14	5.16 ±0.25	-1.84 ±0.47
200	11.27 ±0.37	12.18 ±0.06	4.26 ±0.48	-3.94 ±0.37
500	10.14 ±0.67	10.63 ±0.34	2.85 ±0.63	-8.27 ±0.32
1000	9.14 ±0.57	9.41 ±0.22	1.18 ±0.27	-10.05 ±0.55
1500	8.64 ±0.65	8.90 ±0.37	0.35 ±0.18	-11.70 ±0.09
2000 (default)	7.81 ±0.61	7.14 ±0.48	-0.71 ±0.38	-12.29 ±0.41

Table A.4 Word accuracy (%) of speaker independent lipreading using context dependent GMMs (tri-phone models) with various numbers of tied-states.

# Context dependent states	Speaker independent (SI)			
	Eigenlips 30 dim	DAE 30 dim	DCT 44 dim	DTCWT 66 dim
50	7.97 ±0.28	10.50 ±0.49	3.90 ±0.21	-0.75 ±0.42
100	10.48 ±0.47	12.46 ±0.86	5.18 ±0.51	0.84 ±0.45
200	11.23 ±0.30	12.36 ±0.25	5.11 ±0.53	-0.67 ±0.50
500	11.39 ±0.44	10.49 ±0.54	4.09 ±0.55	-4.26 ±0.32
1000	9.97 ±0.33	9.28 ±0.46	2.79 ±0.52	-6.36 ±0.44
1500	9.83 ±0.29	8.13 ±0.64	1.91 ±0.57	-8.08 ±0.54
2000 (default)	9.46 ±0.14	6.98 ±0.60	1.38 ±0.52	-9.08 ±0.69

For the Eigenlips features, the word accuracy is 12.13% obtained on 100 clusters. For the DCT and the DTCWT features, we observe no significant gain.

If we now turn to the SI result in Table A.4, we observe similar results. Increasing the number of clusters tends to decrease word accuracy, and 100 state-clusters seem to be optimal. The best word accuracy is 12.46% at 100 state-clusters trained on DAE features, followed by Eigenlips features that archive 10.48% word accuracy. We found no significant gain for DCT and DTCWT features.

Increasing the context-dependent state clusters reveals more benefits when more training data is available. As reported in a speaker-independent continuous speech recognition task [79], they found that increasing the number of triphone clusters reduces WER only when they use more data. For example, the optimal cluster for 30 speakers (1200 utterances) is 300 clusters, where 1000 clusters are optimal for 105 speakers (4200 utterances). Therefore, in our task with this particular feature processing, 100 clusters are the optimal number that gives the best word accuracy in SD and SI lipreading.

However, the proposal of applying CD-GMMs is motivated by two reasons: first, to model the phone contextual variations dealing with the visual coarticulation effect; second, to share the training data within the phone cluster. Moreover, the tuned CD cluster is further used for LDA/MLLT estimation in the next step. Therefore, we choose 500 clusters which are high enough to separate the phone group while still yield an equivalent or better result than the monophone (significantly better is only the DAE feature set ($p < 0.01$)).

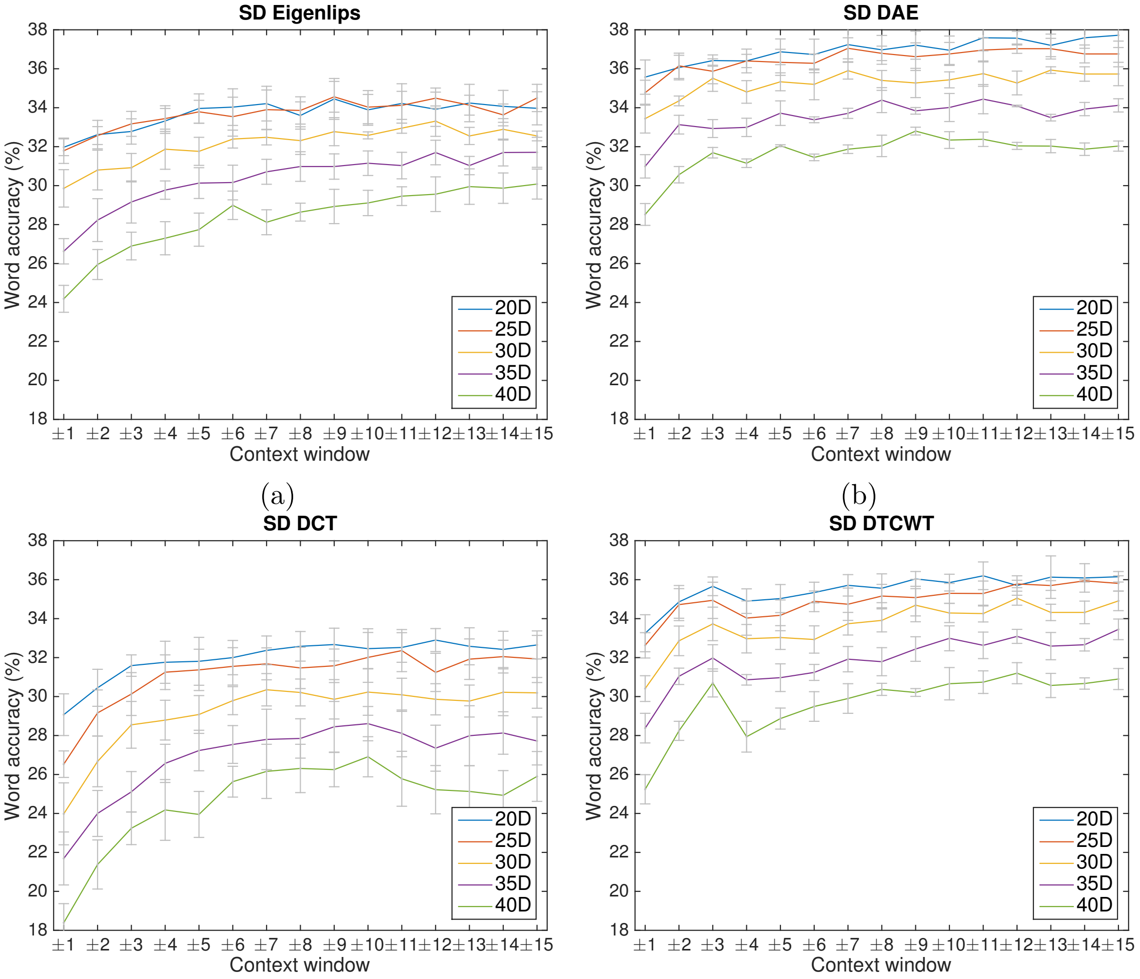
Tuning of window length and dimension of LDA/MLLT transformation

In the previous experiment, we showed that using CD-GMMs with the optimal number of cluster increases lipreading performance. However, the result obtained from the CD-GMM with the $\Delta + \Delta\Delta$ feature processing is not very high. In this step, we use a well-known feature processing method, called LDA/MLLT, to find discriminant features associating with phonetic-state clusters.

A successful lipreading system that employs LDA/MLLT is reported in [113, 114]. As a supervised feature dimensionality reduction method, LDA/MLLT has been applied to reduce dimensionality for both intra-frame to retain class discriminant features and inter-frame to embed dynamic information associated with a speech class. The retained LDA/MLLT dimension is usually 40-41, but the number of concatenated frames differs. In [114, 148], they use ±7 frames, but [4, 1] use only ±3 frames which is equivalent to the setting in acoustic speech recognition.

In this experiment, we investigate LDA/MLLT feature processing over a varying window size starting from ±1 to ±15 frames (10 milliseconds/frame). And we also vary the retained LDA dimension between 20 and 40 dimensions for the five-dimension interval. Here we train CD-GMM, with 500 clusters, and we use 15000 components for the number of Gaussians.

We compare lipreading performance using various configurations of the LDA/MLLT feature transformation to train CD-GMMs for visual speech modelling. Figure A.1 presents the speaker dependent lipreading results. The four sub-figures present the results by the four-types of features: (a) Eigenlips, (b) DAE, (c) DCT, and (d) DTCWT. Each sub-figure presents word accuracy of lipreading (y-axis) as a function of context window size (x-axis) ranging from ±1 to ±15. Here the number following the ± sign indicates the number of frames added. For example, the ±1 case computes LDA/MLLT



(a)

(b)

(c)

(d)

Fig. A.1 Word accuracy (%) of speaker dependent lipreading (SD) using various LDA/MLLT dimensions (ranging from 20 to 40) as a function of context window ($\pm N$) where $N = \{1, 2, ..., 15\}$. The graphs on the top are the results from utilising LDA-MLLT on (a) Eigenlips, (b) DAE; at the bottom are the results on (c) DCT, (d) DTCWT.

features by splicing three static vectors: the current frame, one from the preceding frame, and one from the succeeding frame. Thus, the spliced vector will cover 30-ms. Furthermore, each line in the graph indicates the result obtained from different retained LDA dimensions. Also shown in each line is the ±1 standard error.

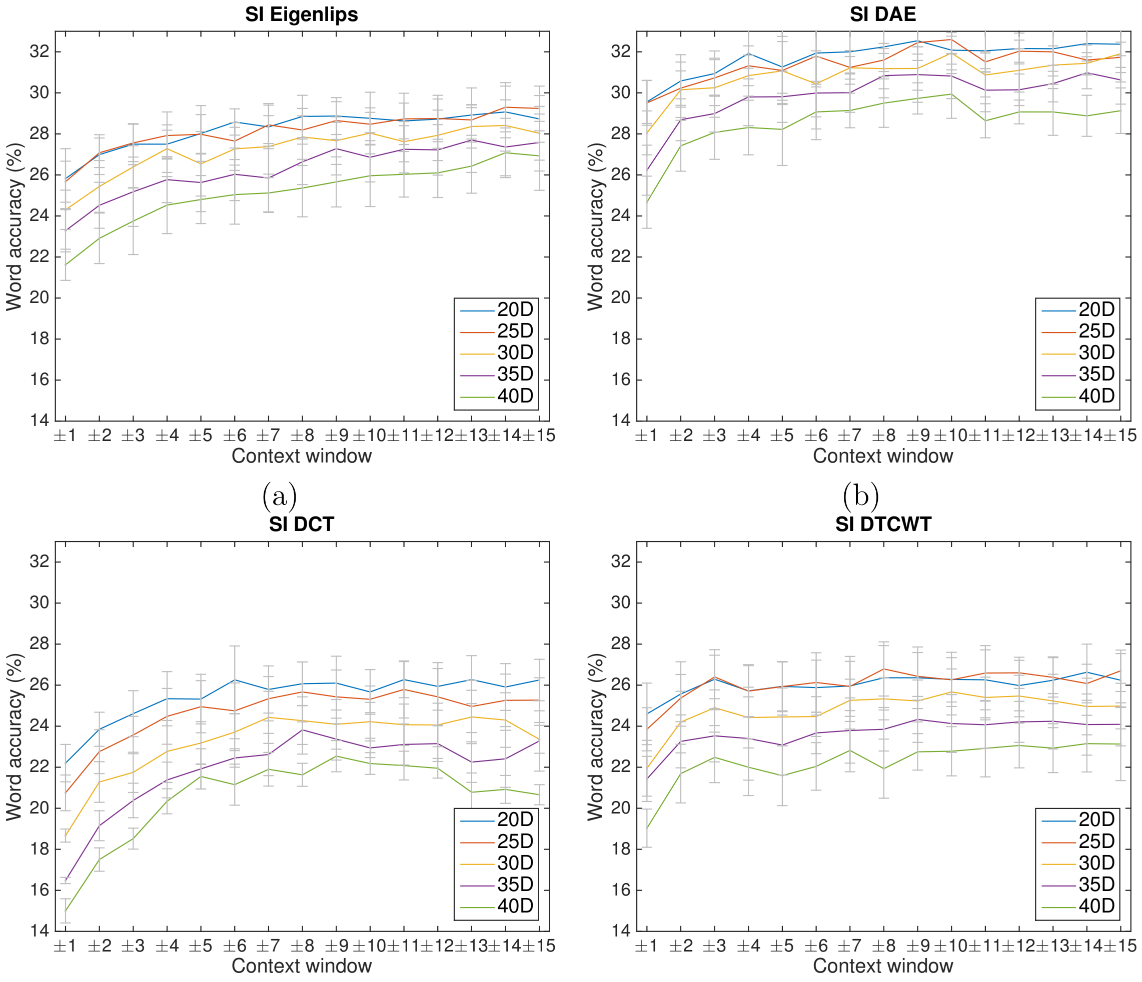
Here we report the best result of each feature set using their best configuration of the LDA/MLLT transformation. For Eigenlips (a), the best result is 34.56% with ±9 context window and retained 25 dimensions. For the DAE (b), the best result is 37.72% with a ±15 context window and 20 retained dimensions. For the DCT (c), the best result is 32.90% with a ±12 context window and 20 retained dimensions. For the DTCWT (d), the best result is 36.20% with a ±11 context window and 20 retained dimensions.

Figure A.2 presents the speaker independent lipreading results. Here word accuracies are reported, ranging from the highest to the lowest performance. First, the best result of SI CD-GMM with LDA/MLLT feature transform is 32.60% word accuracy obtained from the DAE feature with a ±10 context window and 25 retained dimensions. Second, the Eigenlips (a) feature obtained 29.30% word accuracy using a ±14 context window and 25 retained dimensions. Next, the DTCWT (d) result is 26.79% using a ±8 context window and 25 retained dimensions. Last, the DCT (c) feature obtained 26.27% with a ±11 context window and 20 retained dimensions.

Using the LDA/MLLT transformation to find linear discriminant features highly impacts the performance of computer lipreading, resulting in the improvement of word accuracy of all features in both SD and SI conditions. We report the absolute word accuracy gained by comparing to the previous step. In the SD scenario, the absolute performance increases are 22.43% for Eigenlips, 24.10% for DAE, 27.74% for DCT, and 56.54% for DTCWT features. The absolute gain in the SI scenario is also high, namely 17.91% for Eigenlips, 20.14% for DAE, 21.09% for DCT, and 44.94% for DTCWT features.

There is no evidence to retain up to 40 dimensions of LDA/MLLT features. It appears that retaining too many LDA dimensions is unnecessary. We find that reducing the number of dimensions to 20 to 25 dimensions gives the highest word accuracy. This result is in agreement with the findings of Joy et al. [65] on the TORGO dysarthric speech data. They found that reducing LDA dimensions to 25–30 outperforms the 40-dimensional LDA. For context windows, the results show that using ±3 consecutive frames is not optimal. We found that increasing the context window can improve lipreading performance and ±7 seems to be the best. However, if we look at the error bar in the graph, we hardly see the significant improvement among different window sizes. Also there must be a trade-off between accuracy and ability to design online lipreading systems: large context windows increase latency.

It is relevant to note that the best window size differs across features and scenarios. For example, DCT and DTCWT seem to give good results when increasing the window length. However, if we look at the SI scenario where we test on unseen speakers, the best solution is located in the middle, between ±6 and ±8. Therefore, we select LDA/MLLT dimension and window length for each feature set from the best result in



(a) (b)

(c) (d)

Fig. A.2 Word accuracy (%) of speaker independent lipreading (SI) using various LDA/MLLT dimensions as a function of context window ($\pm N$) where $N = \{1, 2, ..., 15\}$. The graph on the top are the results from utilising LDA-MLLT on (a) Eigenlips, (b) DAE; at the bottom are the results on (c) DCT, (d) DTCWT.

their SI scenario. In the next step we train speaker adaptive CD-GMMs using fMLLR features. We estimate the speaker adaptive transform on top of the LDA/MLLT feature.

Tuning the number of context-dependent states on CD-GMM-SAT

The previous experiment applied the LDA/MLLT feature transformation resulting in noticeable gains in CD-GMM lipreading. The best results are 37.72% and 32.60% on the SD and SI tasks respectively, both obtained from the DAE features with LDA/MLLT transformation. Here is the last step of the GMM-HMM training. We use the SAT method to reduce speaker variability and to enhance lipreading performance. We estimate speaker normalised features by applying fMLLR on the extracted LDA/MLLT features. Note that each feature set uses different LDA dimensions and context windows depending on its best configuration.

In this experiment, we investigate the context-dependent state cluster to finalise an appropriate state cluster for each feature set. This fine-tuned cluster will carry on to the DNN-HMM training as a class label, called senone label. We evaluate six different sizes of tied-states clusters: 100, 200, 500, 1000, 1500, and 2000. Here we compare the model on four-types of features with fMLLR feature processing. We train CD-GMM-SAT visual speech model with 15000 Gaussian components.

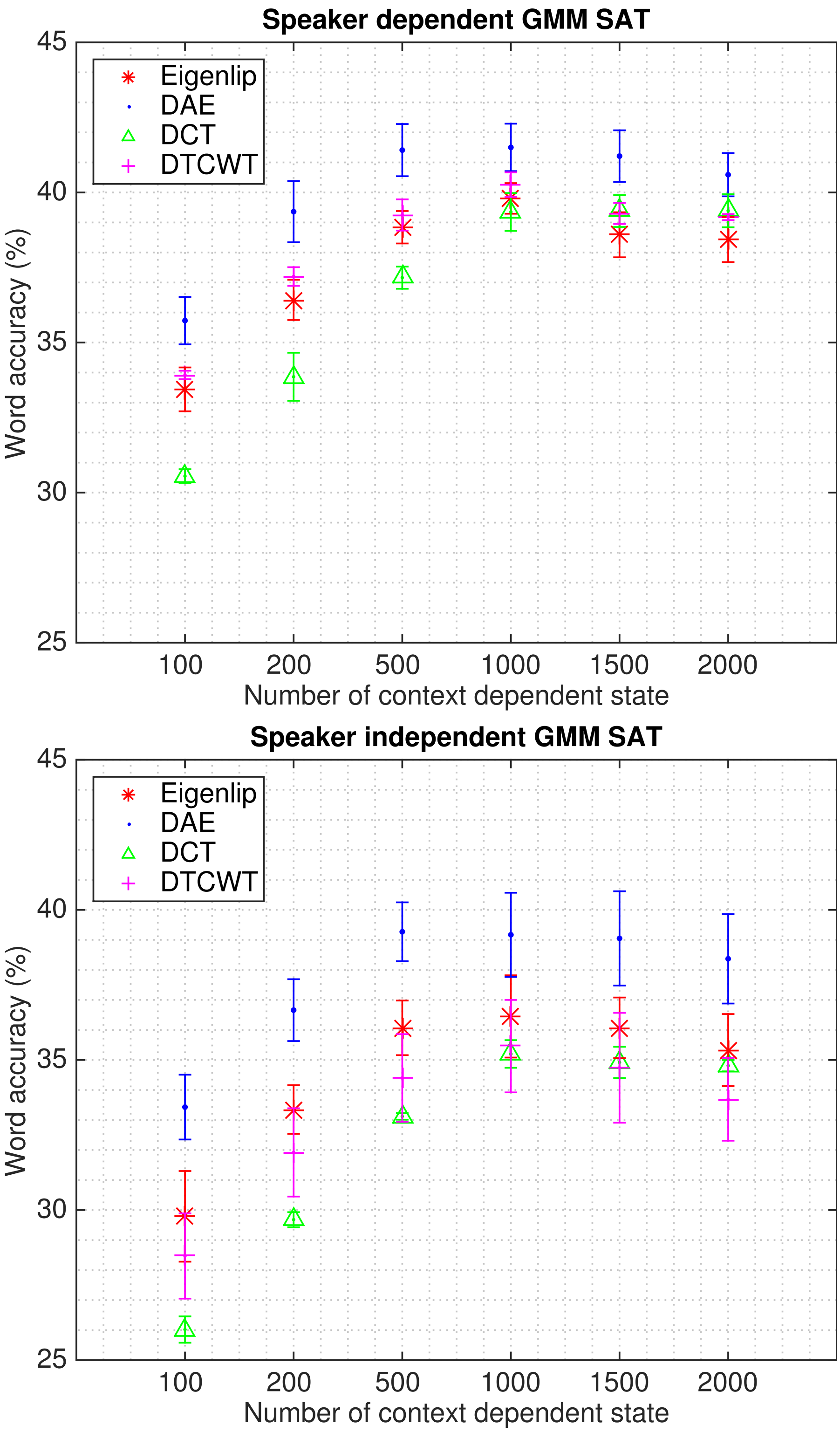


Fig. A.3 Tuning the number of context dependent states of the CD-GMM SAT method. The top graph shows speaker dependent (SD) and the bottom speaker independent (SI) results.

Figure A.3 presents the results obtained from the CD-GMM-SAT system on the SD (top) and the SI (bottom) scenarios. We report word accuracy (y-axis) as a function of the number of tied-state clusters (x-axis). The colours in the bar chart refer to the four feature types with LDA/MLLT and fMLLR feature transforms.

Here we investigate six different numbers of tied-state clusters: 100, 200, 500, 1000, 1500, 2000. In contrast to the results obtained from the CD-GMM system with Delta features, in this CD-GMM-SAT system we found that increasing state clusters improves lipreading performance. The results indicate that 1000 is the optimal number of state clusters for all four feature types and scenarios.

The speaker adaptive training method offers a significant improvement in GMM-HMM training over the LDA/MLLT features. Here we report performance of the SD scenario on the CD-GMM-SAT with fMLLR in word accuracy and the absolute gain compared to the LDA/MLLT results. The DAE features achieved 41.50% word accuracy which is significantly better than other features. It yields a 3.78% gain over the LDA/MLLT feature transformation. There are no significant differences among DTCWT, Eigenlips, and DCT performance. The DTCWT obtains 40.26% word accuracy with 4.06% increase over LDA/MLLT. The Eigenlips features obtain 39.80% word accuracy with a 5.24% improvement. The DCT features obtain 39.35% word accuracy with a 6.45% enhancement.

There is a more substantial improvement in word accuracy of the SI scenario compared to the CD-GMM with LDA/MLLT system. The SI results on CD-GMM-SAT are shown in Figure A.3 (b) and the LDA/MLLT results in Figure A.2. Here DAE still obtains the highest performance but smallest gain compared to other features. The word accuracy of DAE features is 39.17% with a 6.57% absolute increase. The Eigenlips, DCT, and DTCWT results are 36.45%, 35.20% and 35.46% respectively. Compared to the LDA/MLLT, there are 7.15% gains for Eigenlips, 8.93% for DCT, and 8.67% for DTCWT.

From both graphs, it can be seen that by far the highest performance of the GMM-HMM lipreading system is obtained using the DAE feature. The optimal number of clusters is 1000. The best result for SD is 41.50% and for SI is 39.17%. In addition, DAE performance has the highest consistency with the smallest performance differences between scenarios. Comparing word accuracy between SD and SI, the performance differences are 2.33% for DAE, 3.35% for Eigenlips, 4.15% for DCT, and 4.80% for DTCWT features.

In summary, the results in each GMM-HMM training step indicate that applying feature transformation methods, such as LDA/MLLT and fMLLR, contribute directly to the word accuracy of the lipreading system. Careful fine-tuning of the feature extraction parameters that suit each static feature set is necessary. However, we want to note that this fine-tuning over three-fold cross-validation can induce an optimistic bias specially if the dataset is small [22]. To avoid this bias, a better method such as nested cross-validation [138] has been suggested. The final GMM-HMM-SAT model, the context-dependent state clustering, the phonetic time alignment, and the fMLLR features are carried on to use in DNN-HMM training.

Bibliography

[1] Abdelaziz, A. H. (2017). NTCD-TIMIT: A new database and baseline for noise-robust audio-visual speech recognition. In *Proc. Interspeech 2017*, pages 3752–3756.

[2] Ahmed, N., Natarajan, T., and Rao, K. R. (1974). Discrete Cosine Transform. *IEEE Transactions on Computers*, C-23(1):90–93.

[3] Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F., Bayer, J., Belikov, A., Belopolsky, A., Bengio, Y., Bergeron, A., Bergstra, J., Bisson, V., Bleecher Snyder, J., Bouchard, N., Boulanger-Lewandowski, N., Bouthillier, X., de Brébisson, A., Breuleux, O., Carrier, P.-L., Cho, K., Chorowski, J., Christiano, P., Cooijmans, T., Côté, M.-A., Côté, M., Courville, A., Dauphin, Y. N., Delalleau, O., Demouth, J., Desjardins, G., Dieleman, S., Dinh, L., Ducoffe, M., Dumoulin, V., Ebrahimi Kahou, S., Erhan, D., Fan, Z., Firat, O., Germain, M., Glorot, X., Goodfellow, I., Graham, M., Gulcehre, C., Hamel, P., Harlouchet, I., Heng, J.-P., Hidasi, B., Honari, S., Jain, A., Jean, S., Jia, K., Korobov, M., Kulkarni, V., Lamb, A., Lamblin, P., Larsen, E., Laurent, C., Lee, S., Lefrancois, S., Lemieux, S., Léonard, N., Lin, Z., Livezey, J. A., Lorenz, C., Lowin, J., Ma, Q., Manzagol, P.-A., Mastropietro, O., McGibbon, R. T., Memisevic, R., van Merriënboer, B., Michalski, V., Mirza, M., Orlandi, A., Pal, C., Pascanu, R., Pezeshki, M., Raffel, C., Renshaw, D., Rocklin, M., Romero, A., Roth, M., Sadowski, P., Salvatier, J., Savard, F., Schlüter, J., Schulman, J., Schwartz, G., Serban, I. V., Serdyuk, D., Shabanian, S., Simon, E., Spieckermann, S., Subramanyam, S. R., Sygnowski, J., Tanguay, J., van Tulder, G., Turian, J., Urban, S., Vincent, P., Visin, F., de Vries, H., Warde-Farley, D., Webb, D. J., Willson, M., Xu, K., Xue, L., Yao, L., Zhang, S., and Zhang, Y. (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

[4] Almajai, I., Cox, S., Harvey, R., and Lan, Y. (2016). Improved speaker independent lip reading using speaker adaptive training and deep neural networks. In *2016 IEEE*

International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2722–2726.

[5] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., Chen, J., Chen, J., Chen, Z., Chrzanowski, M., Coates, A., Diamos, G., Ding, K., Du, N., Elsen, E., Engel, J., Fang, W., Fan, L., Fougner, C., Gao, L., Gong, C., Hannun, A., Han, T., Johannes, L. V., Jiang, B., Ju, C., Jun, B., LeGresley, P., Lin, L., Liu, J., Liu, Y., Li, W., Li, X., Ma, D., Narang, S., Ng, A., Ozair, S., Peng, Y., Prenger, R., Qian, S., Quan, Z., Raiman, J., Rao, V., Satheesh, S., Seetapun, D., Sengupta, S., Srinet, K., Sriram, A., Tang, H., Tang, L., Wang, C., Wang, J., Wang, K., Wang, Y., Wang, Z., Wang, Z., Wu, S., Wei, L., Xiao, B., Xie, W., Xie, Y., Yogatama, D., Yuan, B., Zhan, J., and Zhu, Z. (2016). Deep Speech 2: End-to-end speech recognition in English and Mandarin. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 173–182. JMLR.org.

[6] Anastasakos, T., McDonough, J., and Makhoul, J. (1997). Speaker adaptive training: a maximum likelihood approach to speaker normalization. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1043–1046 vol.2.

[7] Anina, I., Zhou, Z., Zhao, G., and Pietikäinen, M. (2015). OuluVS2: A multi-view audiovisual database for non-rigid mouth motion analysis. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 1, pages 1–5.

[8] Association, I. P. (1999). *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet.* Cambridge University Press.

[9] Bahl, L., Brown, P., de Souza, P., and Mercer, R. (1986). Maximum mutual information estimation of Hidden Markov Model parameters for speech recognition. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86.*, volume 11, pages 49–52.

[10] Bauman, N. (2011). Speechreading (lip-reading). http://hearinglosshelp.com/blog/speechreading-lip-reading. Accessed on 23rd March 2018.

[11] Bear, H. L. and Harvey, R. (2016). Decoding visemes: Improving machine lip-reading. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 2009–2013. IEEE.

[12] Bear, H. L., Harvey, R. W., Theobald, B.-J., and Lan, Y. (2014). Which phoneme-to-viseme maps best improve visual-only computer lip-reading? In *Advances in Visual Computing*, pages 230–239. Springer.

[13] Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

[14] Bernstein, L. E., Demorest, M. E., and Tucker, P. E. (1998). *What makes a good speechreader? First you have to find one.* Hove, United Kingdom: Psychology Press Ltd. Publishers.

[15] Biswas, A., Sahu, P., and Chandra, M. (2015). Multiple camera in car audio-visual speech recognition using phonetic and visemic information. *Comput. Electr. Eng.*, 47(C):35–50.

[16] Bouveyron, C. and Brunet-Saumard, C. (2014). Model-based clustering of high-dimensional data: A review. *Comput. Stat. Data Anal.*, 71:52–78.

[17] Bregler, C. and Konig, Y. (1994). Eigenlips for robust speech recognition. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, volume ii, pages II/669–II/672 vol.2.

[18] Bridle, J. S. (1990). Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In Soulié, F. F. and Hérault, J., editors, *Neurocomputing*, pages 227–236, Berlin, Heidelberg. Springer Berlin Heidelberg.

[19] Brooke, N. M. and Scott, S. D. (2012). *Visual and audiovisual synthesis and recognition of speech by computers*, pages 159—-192. Cambridge University Press.

[20] Brooke, N. M. and Templeton, P. D. (1990). Visual speech intelligibility of digitally processed facial images. *Proceedings of the Institute of Acoustics*, 12(10):483–490.

[21] Burnham, D., Ambikairajah, E., Arciuli, J., Bennamoun, M., Best, C., Bird, S., Butcher, A., Cassidy, S., Chetty, G., Cox, F., Cutler, A., Dale, R., Epps, J., Fletcher,

J., Goecke, R., Grayden, D., Hajek, J., Ingram, J., Ishihara, S., Kemp, N., Kinoshita, Y., Kuratate, T., Lewis, T., Loakes, D., Onslow, M., Powers, D., Rose, P., Togneri, R., Tran, D., and Wagner, M. (2009). *A Blueprint for a comprehensive Australian English auditory-visual speech corpus*, pages 96–107. Cascadilla Proceedings Project.

[22] Cawley, G. C. and Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(Jul):2079–2107.

[23] Chandrasekaran, C., Trubanova, A., Stillittano, S., Caplier, A., and Ghazanfar, A. A. (2009). The natural statistics of audiovisual speech. *PLOS Computational Biology*, 5(7):1–18.

[24] Chen, W., Er, M. J., and Wu, S. (2005). PCA and LDA in DCT domain. *Pattern Recogn. Lett.*, 26(15):2474–2482.

[25] Chollet, F. et al. (2015). Keras. https://github.com/keras-team/keras.

[26] Chung, J. S., Senior, A., Vinyals, O., and Zisserman, A. (2017). Lip reading sentences in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[27] Chung, J. S. and Zisserman, A. (2016). Lip reading in the wild. In *Asian Conference on Computer Vision*.

[28] Chung, J. S. and Zisserman, A. (2017). Lip reading in profile. In *2017 British Machine Vision Conference (BMVC)*. BMVC.

[29] Cohen, M. M., Massaro, D. W., et al. (1993). Modeling coarticulation in synthetic visual speech. *Models and Techniques in Computer Animation*, 92:139–156.

[30] Cooke, M., Barker, J., Cunningham, S., and Shao, X. (2006). An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 120(5):2421–2424.

[31] Cootes, T. F., Edwards, G. J., and Taylor, C. J. (2001). Active Appearance Models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):681–685.

[32] Cox, S. J., Harvey, R. W., Lan, Y., Newman, J. L., and Theobald, B.-J. (2008). The challenge of multispeaker lip-reading. In *Auditory-Visual Speech Processing (AVSP)*, pages 179–184.

[33] Dahl, G., Yu, D., Deng, L., and Acero, A. (2012a). Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing (receiving 2013 IEEE SPS Best Paper Award)*, 20(1):30–42.

[34] Dahl, G. E., Yu, D., Deng, L., and Acero, A. (2012b). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42.

[35] Deena, S. P. (2012). *Visual speech synthesis by learning joint probabilistic models of audio and video*. PhD thesis, School of Computing Sciences, The University of Manchester.

[36] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal sSatistical Society. Series B (methodological)*, pages 1–38.

[37] Deng, L. and Yu, D. (2014). Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7(3–4):197–387.

[38] Feng, X. and Wang, W. (2008). DTCWT-based dynamic texture features for visual speech recognition. In *APCCAS 2008 - 2008 IEEE Asia Pacific Conference on Circuits and Systems*, pages 497–500.

[39] Fiscus, J. (1997). A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER). In *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 347–354.

[40] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Human Genetics*, 7(2):179–188.

[41] Furui, S. (1986). Speaker-independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(1):52–59.

[42] Gales, M. (1998). Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12(2):75–98.

[43] Gales, M. and Young, S. (2008). The application of hidden markov models in speech recognition. *Foundations and trends in signal processing*, 1(3):195–304.

[44] Gillick, L. and Cox, S. J. (1989). Some statistical issues in the comparison of speech recognition algorithms. In *International Conference on Acoustics, Speech, and Signal Processing,*, pages 532–535 vol.1.

[45] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

[46] Gopinath, R. A. (1998). Maximum likelihood modeling with Gaussian distributions for classification. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 2, pages 661–664. IEEE.

[47] Graves, A. and Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1764–II–1772. JMLR.org.

[48] Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664.

[49] Harte, N. and Gillen, E. (2015). TCD-TIMIT: An audio-visual corpus of continuous speech. *IEEE Transactions on Multimedia*, 17(5):603–615.

[50] Hazen, T. J. (2006). Visual model structures and synchrony constraints for audio-visual speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(3):1082–1089.

[51] Hazen, T. J., Saenko, K., La, C.-H., and Glass, J. R. (2004). A segment-based audio-visual speech recognizer: Data collection, development, and initial experiments. In *Proceedings of the 6th International Conference on Multimodal Interfaces*, ICMI '04, pages 235–242, New York, NY, USA. ACM.

[52] Heckmann, M., Berthommier, F., and Kroschel, K. (2002a). Noise adaptive stream weighting in audio-visual speech recognition. *EURASIP J. Appl. Signal Process.*, 2002(1):1260–1273.

[53] Heckmann, M., Kroschel, K., Savariaux, C., and Berthommier, F. (2002b). DCT-based video features for audio-visual speech recognition. In *Seventh International Conference on Spoken Language Processing.*

[54] Hickey, R. (2005). *Dublin English: Evolution and Change*. Varieties of English around the world: General series. J. Benjamins Publishing Company.

[55] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97.

[56] Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800.

[57] Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554.

[58] Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

[59] Hong, X., Yao, H., Wan, Y., and Chen, R. (2006). A PCA based visual DCT feature extraction method for lip-reading. In *2006 International Conference on Intelligent Information Hiding and Multimedia*, pages 321–326.

[60] Howell, D. (2015). *Confusion modelling for lip-reading*. PhD thesis, University of East Anglia.

[61] Howell, D., Cox, S., and Theobald, B. (2016). Visual units and confusion modelling for automatic lip-reading. *Image Vision Comput.*, 51(C):1–12.

[62] Huang, J. and Kingsbury, B. (2013). Audio-visual deep learning for noise robust speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7596–7599.

[63] Huang, J., Potamianos, G., Connell, J., and Neti, C. (2004). Audio-visual speech recognition using an infrared headset. *Speech Communication*, 44(1):83–96. Special Issue on Audio Visual speech processing.

[64] Jeffers, J. and Barley, M. (1971). *Speechreading (lipreading)*. Thomas books. Charles C Thomas Publisher, Springfield, Illinois, USA.

[65] Joy, N. M., Umesh, S., and Abraham, B. (2017). On improving acoustic models for torgo dysarthric speech database. In *Proc. Interspeech 2017*, pages 2695–2699.

[66] Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition.

[67] Kapadia, S., Valtchev, V., and Young, S. J. (1993). MMI training for continuous phoneme recognition on the timit database. In *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 491–494 vol.2.

[68] Kaplan, H., Bally, S., and Garretson, C. (1985). *Speechreading: A Way to Improve Understanding*. Way to Improve Understanding. Gallaudet University Press.

[69] Karanasou, P., Gales, M., and Woodland, P. (2015). I-vector estimation using informative priors for adaptation of deep neural networks. In *Proc. Interspeech 2015*, pages 2872–2876. ISCA.

[70] Karanasou, P., Wang, Y., Gales, M. J., and Woodland, P. C. (2014). Adaptation of deep neural network acoustic models using factorised i-vectors. In *Proc. Interspeech 2014*, pages 2180–2184.

[71] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

[72] Kingsbury, B. (2009). Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3761–3764.

[73] Kingsbury, N. (2001). Complex wavelets for shift invariant analysis and filtering of signals. *Applied and Computational Harmonic Analysis*, 10(3):234–253.

[74] Kirby, M., Weisser, F., and Dangelmayr, G. (1993). A model problem in the representation of digital image sequences. *Pattern Recognition*, 26(1):63–73.

[75] Kirtley, C., Bryant, P., MacLean, M., and Bradley, L. (1989). Rhyme, rime, and the onset of reading. *Journal of Experimental Child Psychology*, 48(2):224–245.

[76] Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86.

[77] Lan, Y., Harvey, R., and Theobald, B. J. (2012). Insights into machine lip reading. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4825–4828.

[78] Lee, B., Hasegawa-johnson, M., Goudeseune, C., Kamdar, S., Borys, S., Liu, M., and Huang, T. (2004). AVICAR: Audio-visual speech corpus in a car environment. In *in Proc. INTERSPEECH*, pages 2489–2492.

[79] Lee, K. F. (1990). Context-dependent phonetic hidden Markov models for speaker-independent continuous speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(4):599–609.

[80] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

[81] Luettin, J. and Thacker, N. A. (1997). Speechreading using probabilistic models. *Computer Vision and Image Understanding*, 65(2):163–178.

[82] Marschark, M., LePoutre, D., and Bement, L. (1998). *Mouth movement and signed communication*. Hove, United Kingdom: Psychology Press Ltd. Publishers.

[83] Massaro, D. W. and Stork, D. G. (1998). Speech recognition and sensory integration: a 240-year-old theorem helps explain how people and machines can integrate auditory and visual information to understand speech. *American Scientist*, pages 236–244.

[84] Matthews, I., Bangham, J. A., and Cox, S. (1996). Audiovisual speech recognition using multiscale nonlinear image decomposition. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 1, pages 38–41 vol.1.

[85] Matthews, I., Cootes, T., Bangham, J., Cox, S., and Harvey, R. (2002). Extraction of visual features for lipreading. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 24(2):198–213.

[86] McClain, M., Brady, K., Brandstein, M., and Quatieri, T. (2004). Automated lip-reading for improved speech intelligibility. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 1, pages I–701–4 vol.1.

[87] McGurk, H. and MacDonald, J. (1976). Hearing lips and seeing voices. *Nature Publishing Group.*

[88] McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

[89] Mohamed, A. R., Dahl, G. E., and Hinton, G. (2012). Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):14–22.

[90] Mohri, M., Pereira, F., and Riley, M. (2008). *Speech Recognition with Weighted Finite-State Transducers*, pages 559–584. Springer Berlin Heidelberg, Berlin, Heidelberg.

[91] Movellan, J. R. (1995). Visual speech recognition with stochastic networks. In *Advances in neural information processing systems*, pages 851–858.

[92] Mroueh, Y., Marcheret, E., and Goel, V. (2015). Deep multimodal learning for audio-visual speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 2130–2134.

[93] Munhall, K. G., Gribble, P., Sacco, L., and Ward, M. (1996). Temporal constraints on the McGurk effect. *Perception & Psychophysics*, 58(3):351–362.

[94] Navarra, J. and Soto-Faraco, S. (2007). Hearing lips in a second language: Visual articulatory information enables the perception of second language sounds. *Psychological research*, 71(1):4–12.

[95] Neti, Potamianos, Luettin, Matthews, Glotin, Vergyri, Sison, Mashari, and Zhou (2000). Audio-visual speech recognition. Technical report, Center for Language and Speech Processing, The Johns Hopkins University, Baltimore.

[96] Neti, C., Potamianos, G., Luettin, J., Matthews, I., Glotin, H., and Vergyri, D. (2001). Large-vocabulary audio-visual speech recognition: a summary of the Johns Hopkins Summer 2000 Workshop. In *2001 IEEE Fourth Workshop on Multimedia Signal Processing*, pages 619–624.

[97] Newman, J. L., Theobald, B.-J., and Cox, S. J. (2010). Limitations of visual speech recognition. In *Auditory-Visual Speech Processing (AVSP) 2010*.

[98] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011). Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696.

[99] Nicholls, G. H. and Mcgill, D. L. (1982). Cued speech and the reception of spoken language. *Journal of Speech, Language, and Hearing Research*, 25(2):262–269.

[100] Ninomiya, H., Kitaoka, N., Tamura, S., Iribe, Y., and Takeda, K. (2015). Integration of Deep Bottleneck Features for Audio-Visual Speech Recognition. *Interspeech*, pages 563–567.

[101] Noda, K., Yamaguchi, Y., Nakadai, K., Okuno, H. G., and Ogata, T. (2015). Audio-visual speech recognition using deep learning. *Applied Intelligence*, 42(4):722–737.

[102] Ortiz, I. d. l. R. R. (2008). Lipreading in the prelingually deaf: what makes a skilled speechreader? *The Spanish Journal of Psychology*, 11(2):488–502.

[103] Pan, J., Liu, C., Wang, Z., Hu, Y., and Jiang, H. (2012). Investigation of deep neural networks (DNN) for large vocabulary continuous speech recognition: Why DNN surpasses GMMs in acoustic modeling. In *The 8th International Symposium on Chinese Spoken Language Processing 2012*, pages 301–305.

[104] Papandreou, G., Katsamanis, A., Pitsikalis, V., and Maragos, P. (2009). Adaptive multimodal fusion by uncertainty compensation with application to audiovisual speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(3):423–435.

[105] Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition. In Xianghua Xie, M. W. J. and Tam, G. K. L., editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 41.1–41.12. BMVA Press.

[106] Parsons, T. (1987). *Voice and speech processing.* McGraw Hill Series in Electrical and Computer Engineering. McGraw-Hill.

[107] Patterson, E. K., Gurbuz, S., Tufekci, Z., and Gowdy, J. N. (2002). CUAVE: A new audio-visual database for multimodal human-computer interface research. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 2017–2020.

[108] Peeling, S., Moore, R., and Tomlinson, M. (1986). The multi-layer perceptron as a tool for speech pattern processing research. In *Proceedings of the 10th Autumn Conference on Speech and Hearing.*

[109] Pei, Y., Kim, T. K., and Zha, H. (2013). Unsupervised random forest manifold alignment for lipreading. In *2013 IEEE International Conference on Computer Vision*, pages 129–136.

[110] Petajan, E. D. (1984). *Automatic Lipreading to Enhance Speech Recognition (Speech Reading)*. PhD thesis, Champaign, IL, USA. AAI8502266.

[111] Petridis, S., Li, Z., and Pantic, M. (2017). End-to-end visual speech recognition with LSTMS. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2592–2596.

[112] Potamianos, G. and Graf, H. P. (1998). Linear discriminant analysis for speechreading. In *1998 IEEE Second Workshop on Multimedia Signal Processing*, pages 221–226.

[113] Potamianos, G., Luettin, J., and Neti, C. (2001). Hierarchical discriminant features for audio-visual LVCSR. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 1, pages 165–168. IEEE.

[114] Potamianos, G., Neti, C., Gravier, G., Garg, A., and Senior, A. (2003). Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 91(9):1306–1326.

[115] Potamianos, G., Neti, C., Luettin, J., and Matthews, I. (2012). Audiovisual automatic speech recognition. In Bailly, G., Perrier, P., and Vatikotis-Bateson, E., editors, *Audiovisual Speech Processing*, page 193–247. Cambridge University Press, Cambridge, UK.

[116] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The Kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*.

[117] Povey, D., Hannemann, M., Boulianne, G., Burget, L., Ghoshal, A., Janda, M., Karafiát, M., Kombrink, S., Motlíček, P., Qian, Y., Riedhammer, K., Veselý, K., and Vu, N. T. (2012). Generating exact lattices in the WFST framework. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4213–4216.

[118] Povey, D. and Saon, G. (2006). Feature and model space speaker adaptation with full covariance Gaussians. In *INTERSPEECH 2006 - ICSLP, Ninth International Conference on Spoken Language Processing, Pittsburgh, PA, USA, September 17-21, 2006*.

[119] Povey, D. and Woodland, P. C. (2002). Minimum phone error and i-smoothing for improved discriminative training. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 105–108.

[120] Price, P., Fisher, W. M., Bernstein, J., and Pallett, D. S. (1988). The DARPA 1000-word resource management database for continuous speech recognition. In *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pages 651–654. IEEE.

[121] Puviarasan, N. and Palanivel, S. (2011). Lip reading of hearing impaired persons using HMM. *Expert Syst. Appl.*, 38(4):4477–4481.

[122] Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

[123] Rao, K. R. and Yip, P. (1990). *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press Professional Inc., San Diego, CA, USA.

[124] Rosen, S. (1992). Temporal information in speech: acoustic, auditory and linguistic aspects. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 336(1278):367–373.

[125] Rudzicz, F., Namasivayam, A. K., and Wolff, T. (2012). The TORGO database of acoustic and articulatory speech from speakers with dysarthria. *Lang. Resour. Eval.*, 46(4):523–541.

[126] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

[127] Sak, H., Vinyals, O., Heigold, G., Senior, A., McDermott, E., Monga, R., and Mao, M. (2014). Sequence discriminative distributed training of long short-term memory recurrent neural networks. *Entropy*, 15(16):17–18.

[128] Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409.

[129] Saon, G., Soltau, H., Nahamoo, D., and Picheny, M. (2013). Speaker adaptation of neural network acoustic models using i-vectors. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 55–59.

[130] Scanlon, P., Potamianos, G., Libal, V., and Chu, S. M. (2004). Mutual information based visual feature selection for lipreading. In *in Proc. of ICSLP 2004, South Korea*, pages 4–8.

[131] Seltzer, M., Yu, D., and Wang, Y. (2013a). An investigation of deep neural networks for noise robust speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7398–7402.

[132] Seltzer, M. L., Yu, D., and Wang, Y. (2013b). An investigation of deep neural networks for noise robust speech recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7398–7402.

[133] Seymour, R., Stewart, D., and Ming, J. (2008). Comparison of image transform-based features for visual speech recognition in clean and corrupted videos. *J. Image Video Process.*, 2008:14:1–14:9.

[134] Southwick, L. and Vacala, M. (2008). Chapter 31 - patients with disabilities. In Ballweg, R., Sullivan, E. M., Brown, D., and Vetrosky, D., editors, *Physician Assistant (Fourth Edition)*, pages 593–606. W.B. Saunders, Philadelphia, fourth edition.

[135] Srivastava, N. and Salakhutdinov, R. R. (2012). Multimodal learning with deep boltzmann machines. In *Advances in Neural Information Processing Systems*, pages 2222–2230.

[136] Stolcke, A. and Droppo, J. (2017). Comparing human and machine errors in conversational speech transcription. In *Proc. Interspeech*, pages 137–141. ISCA - International Speech Communication Association.

[137] Stolcke, A. et al. (2002). SRILM-an extensible language modeling toolkit. In *7th International Conference on Spoken Language Processing (ICSLP)*, pages 901–904.

[138] Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society. Series B (Methodological)*, pages 111–147.

[139] Su, H., Li, G., Yu, D., and Seide, F. (2013). Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription. In *ICASSP 2013*, pages 6664–6668.

[140] Sui, C., Bennamoun, M., and Togneri, R. (2015). Listening with your eyes: Towards a practical visual speech recognition system using deep Boltzmann machines. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 154–162.

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

[142] Summerfield, Q. (1992). Lipreading and audio-visual speech perception. *Philosophical Transactions: Biological Sciences*, 335(1273):71–78.

[143] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9.

[144] Takashima, Y., Kakihara, Y., Aihara, R., Takiguchi, T., Ariki, Y., Mitani, N., Omori, K., and Nakazono, K. (2015). Audio-visual speech recognition using convolutive bottleneck networks for a person with severe hearing loss. *IPSJ Transactions on Computer Vision and Applications*, 7:64–68.

[145] Tamura, S., Ninomiya, H., Kitaoka, N., Osuga, S., Iribe, Y., Takeda, K., and Hayamizu, S. (2015). Audio-visual speech recognition using deep bottleneck features and high-performance lipreading. pages 575–582.

[146] Taylor, S. L., Mahler, M., Theobald, B.-J., and Matthews, I. (2012). Dynamic units of visual speech. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '12, pages 275–284, Goslar Germany, Germany. Eurographics Association.

[147] Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

[148] Thangthai, K. and Harvey, R. (2017). Improving computer lipreading via DNN sequence discriminative training techniques. In *Proc. Interspeech 2017*, pages 3657–3661.

[149] Thangthai, K., Harvey, R. W., Cox, S. J., and Theobald, B. (2015). Improving lip-reading performance for robust audiovisual speech recognition using DNNs. In *The 1st Joint Conference on Facial Analysis, Animation and Auditory-Visual Speech Processing FAAVSP 2015, Vienna, Austria, September 11-13, 2015*, pages 127–131.

[150] Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

[151] Vertanen, K. (2006). Baseline WSJ acoustic models for HTK and Sphinx: Training recipes and recognition experiments. Technical report, Cambridge, United Kingdom: Cavendish Laboratory.

[152] Veselỳ, K., Ghoshal, A., Burget, L., and Povey, D. (2013). Sequence-discriminative training of deep neural networks. In *Interspeech*, pages 2345–2349.

[153] Voigtlaender, P., Doetsch, P., Wiesler, S., Schlüter, R., and Ney, H. (2015). Sequence-discriminative training of recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2100–2104.

[154] Wassenhove, V. V., Grant, K. W., and Poeppel, D. (2007). Temporal window of integration in auditory-visual speech perception. *Neuropsychologia*, 45(3):598–607. Advances in Multisensory Processes.

[155] Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

[156] Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1-3):37–52.

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

[158] Xilinx, I. (2002). 2d discrete cosine transform (dct) v2.0 logicore product specification.

[159] Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M. L., Stolcke, A., Yu, D., and Zweig, G. (2017). Toward human parity in conversational speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(12):2410–2423.

[160] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3320–3328, Cambridge, MA, USA. MIT Press.

[161] Young, S. J., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. (2006). *The HTK Book Version 3.4*. Cambridge University Press.

[162] Young, S. J., Odell, J. J., and Woodland, P. C. (1994). Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 307–312, Stroudsburg, PA, USA. Association for Computational Linguistics.

[163] Yu, D. and Deng, L. (2014). *Automatic Speech Recognition - A Deep Learning Approach*. Springer.

[164] Zhao, G., Barnard, M., and Pietikainen, M. (2009). Lipreading with local spatiotemporal descriptors. *IEEE Transactions on Multimedia*, 11(7):1254–1265.

[165] Zhou, Z., Zhao, G., and Pietikäinen, M. (2011). Towards a practical lipreading system. In *CVPR 2011*, pages 137–144.

[166] Zimmermann, M., Mehdipour Ghazi, M., Ekenel, H. K., and Thiran, J.-P. (2017). Visual speech recognition using PCA networks and LSTMs in a tandem GMM-HMM system. In Chen, C.-S., Lu, J., and Ma, K.-K., editors, *Computer Vision – ACCV 2016 Workshops*, pages 264–276, Cham. Springer International Publishing.