Visual speech recognition: from traditional to deep learning frameworks

THÈSE N^{O} 8799 (2018)

PRÉSENTÉE LE 31 AOÛT 2018

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE TRAITEMENT DES SIGNAUX 5
PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Marina ZIMMERMANN

acceptée sur proposition du jury:

Dr M. Mattavelli, président du jury
Prof. J.-Ph. Thiran, directeur de thèse
Prof. G. Potamianos, rapporteur
Prof. T. Schultz, rapporteuse
Dr J.-M. Odobez, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2018

Acknowledgements

This work would not have been possible without the support of many others. Therefore, I would like to acknowledge their contributions at this point.

First of all, I would like to thank my thesis director Prof. Jean-Philippe Thiran. Thank you for always being supportive and encouraging, especially in moments where no solution seemed to be in sight. In the last two years I was also very lucky to work together with Prof. Hazım Kemal Ekenel. Thank you for all the detailed discussions and the scientific supervision. I would also like to thank the jury members of my thesis oral examination, Dr Marco Mattavelli, Prof. Gerasimos Potamianos, Prof. Tanja Schultz and Dr Jean-Marc Odobez, for their detailed feedback and insightful comments.

Along this journey I have worked with many people: Thank you to Estelle Chin and Olivier Payot for the collaboration with PSA in the beginning. Thank you also to Dr Mathew Magimai-Doss for several discussions on speech recognition when I was diving deeper into these topics.

I would also like to thank all the current and former members of LTS5, for common lunches, matches of 'baby-foot' or other activities: Alessandra, Alessandro, Alia, Alice, Anıl, Anna, Audrey, Behzad, Carlos, Christina, Christophe E., Christophe P., Damien, David, Didrik, Dimitri, Elda, Elena, Frank, Gabriel C., Gabriel G., Gaëtan, Hua, Jelena, Jonathan, Marco, Mário, Mina, Ming, Muhamed, Murat, Rafael, Ricardo, Saeed, Saleh, Tom, Vijay and others. Since we share the corridor with the other signal processing labs, I was also happy to meet Anne-Flore, Eda, Hamed, Raphaël, Sasan, Sibylle and many others there and spend time with them on and off campus. I am also very glad I could work within the 'Face group', a sub-group of LTS5: thank you Anıl, Behzad, Damien, Hazım, Hua, Gabriel, Murat, Saeed, Saleh. Thank you also to my office mates over the years: Gabriel, Alessandro, Anna, Tim and Damien.

Without the organisational skills of Rosie, and lately Anne, the labs would surely have trouble finishing all the administrative work on time. Thank you for all your help and nice chats.

Since a PhD is not only work, I would also like to thank all those who made lunches, dinners and trips in and outside Lausanne more fun: Audrey, Carlos, Deniz, Didrik, Elena, Gabriel, Hyunjin, Irene, Marili, Mário, Maya, Mina, Murat, Olivia, Ricardo, Sasan, Tom and Vijay. Thank you also to all my friends outside EPFL who reminded me that there is indeed a world out there: Akshara, Alexandra F., Alexandra P., Amandine, Ana, Angela, Bity, Eva, Laura, Linn, Lisa, Kathi, Kerstin, Khushboo, Simone, Sophie

and Vivienne.
A heartfelt thank you to Vijay, who accompanied me along this journey and lent a helping hand where necessary and skipped along the path with me during happy moments.
Finally, I would like to thank my family for their continuous support and words of encouragement. Thank you in particular to my parents who showed me never to give up.

Lausanne, July 2018 Marina Zimmermann

Abstract

Speech is the most natural means of communication for humans. Therefore, since the beginning of computers it has been a goal to interact with machines via speech. While there have been gradual improvements in this field over the decades, and with recent drastic progress more and more commercial software is available that allow voice commands, there are still many ways in which it can be improved.

One way to do this is with visual speech information, more specifically, the visible articulations of the mouth. Based on the information contained in these articulations, visual speech recognition (VSR) transcribes an utterance from a video sequence. It thus helps extend speech recognition from audio-only to other scenarios such as silent or whispered speech (e.g. in cybersecurity), mouthings in sign language, as an additional modality in noisy audio scenarios for audio-visual automatic speech recognition, to better understand speech production and disorders, or by itself for human machine interaction and as a transcription method.

In this thesis, we present and compare different ways to build systems for VSR: We start with the traditional hidden Markov models that have been used in the field for decades, especially in combination with handcrafted features. These are compared to models taking into account recent developments in the fields of computer vision and speech recognition through deep learning. While their superior performance is confirmed, certain limitations with respect to computing power for these systems are also discussed.

This thesis also addresses multi-view processing and fusion, which is an important topic for many current applications. This is due to the fact that a single camera view often cannot provide enough flexibility with speakers moving in front of the camera. Technology companies are willing to integrate more cameras into their products, such as cars and mobile devices, due to lower hardware cost for both cameras and processing units, as well as the availability of higher processing power and high performance algorithms. Multi-camera and multi-view solutions are thus becoming more common, which means that algorithms can benefit from taking these into account. In this work we propose several methods of fusing the views of multiple cameras to improve the overall results.

We can show that both, relying on deep learning-based approaches for feature extraction and sequence modelling, as well as taking into account the complementary information contained in several views, improves performance considerably. To further improve the results, it would be necessary to move from data recorded in a lab environment, to multi-view data in realistic scenarios. Furthermore, the findings and models could be

transferred to other domains such as audio-visual speech recognition or the study of speech production and disorders.

Key words: visual speech recognition; automatic lip-reading; multi-view processing; GMM-HMM; deep learning

Résumé

La parole est la forme de communication la plus naturelle des humains. C'est la raison pour laquelle, depuis le début des ordinateurs, on a cherché à interagir avec les machines à travers la parole. Bien qu'il y ait eu des améliorations graduelles dans ce domaine pendant des décennies, et qu'il existe de plus en plus de programmes commerciaux qui permettent des commandes vocales, bon nombre de points doivent encore être améliorés. Pour ce faire, une méthode utilise les informations de la parole visuelle, ou plus particulièrement, les articulations visibles de la bouche.

Basée sur les informations contenues dans ces articulations, la reconnaissance de la parole visuelle transcrit un énoncé depuis une séquence vidéo. Ainsi, elle permet d'étendre la reconnaissance de la parole de l'audio uniquement à d'autres scénarios, tels que : la parole silencieuse ou chuchotée utile pour la sécurité informatique, les « mouthings » (articulations des lèvres) de la langue des signes, une modalité de plus dans les scénarios d'audio bruité pour la reconnaissance de la parole audio-visuelle, une meilleure compréhension de la production de la parole ou des troubles du langage, l'interaction homme-machine ou une méthode de transcription.

Dans cette thèse, nous présentons et comparons des manières différentes de développer des systèmes de reconnaissance de la parole visuelle. Nous commençons par étudier les modèles de Markov cachés traditionnels, utilisés dans ce domaine pendant des décennies, en particulier en combinaison avec des caractéristiques choisies manuellement. Puis, nous comparons ces systèmes à des modèles qui prennent en compte les développements récents, par l'apprentissage profond, dans les domaines de la vision par l'ordinateur et de la reconnaissance de la parole. Même si leur performance supérieure est confirmée, il est important de souligner certaines limites concernant leur besoin de puissance de calcul.

Cette thèse développe aussi le traitement et la fusion de plusieurs angles de vues, ce qui est un sujet important pour beaucoup d'applications récentes. Cela est dû au fait qu'une seule caméra ne peut pas donner assez de flexibilité au locuteur qui bouge devant celle-ci. Les entreprises de technologie sont prêtes à intégrer plusieurs caméras dans leurs appareils, comme les voitures ou les appareils portables, suite à la baisse des prix des caméras et des processeurs, aux puissances de calcul plus élevées et aux algorithmes plus performants. Des solutions multi-caméra et multi-vue deviennent ainsi plus communes, ce que requièrent les algorithmes pour les prendre en compte. Dans ce travail, nous proposons plusieurs méthodes pour la fusion de vues de plusieurs caméras, afin d'améliorer les résultats finaux.

Nous pouvons constater que les deux approches : se baser sur l'apprentissage profond pour l'extraction de caractéristiques et la modélisation de séquences, ainsi que prendre en compte l'information complémentaire contenue dans plusieurs vues, améliore fortement la performance. Afin d'améliorer les résultats, il serait nécessaire de changer les données enregistrées dans un environnement de laboratoire, en données de plusieurs vues dans les scénarios réalistes. En outre, les résultats et les modèles pourraient être transposés à quelques autres domaines comme la reconnaissance de la parole audio-visuelle ou l'investigation de la production de la parole et des troubles du langage.

Mots clefs : reconnaissance de la parole visuelle ; lecture labiale automatique ; traitement de plusieurs vues ; MMG-MMC ; apprentissage profond

Zusammenfassung

Sprache ist die natürlichste Form der menschlichen Kommunikation. Daher existiert seit der Erfindung des Computers das Ziel, mit den Maschinen über Sprache zu interagieren. Auch wenn auf diesem Gebiet in den letzten Jahrzehnten kontinuierliche Fortschritte erreicht wurden und immer mehr Computerprogramme Sprachkommandos ermöglichen, gibt es weiterhin viele Verbesserungsmöglichkeiten.

Eine Möglichkeit der Optimierung besteht in den visuellen Sprachinformationen, präziser, in den sichtbaren Mundbewegungen. Gestützt auf die darin enthaltenen Informationen transkribiert die visuelle Spracherkennung die Äußerung einer Videosequenz. Dies erlaubt, die Spracherkennung von reinem Audio auf andere Szenarien auszuweiten, wie beispielsweise auf lautlose oder geflüsterte Sprache (z. B. in der Computersicherheit), „Mouthings“ (Lippenbewegungen) in der Gebärdensprache, als zusätzliche Modalität in geräuschvollen Audioszenarien für audio-visuelle Spracherkennung, zum besseren Verständnis von Sprachproduktion und Sprechstörungen oder alleine für die Mensch-Computer-Interaktion und als Methode zur Transkription von Videos.

In dieser Doktorarbeit präsentieren und vergleichen wir verschiedene Methoden zur Entwicklung eines visuellen Spracherkennungssystems: Wir beginnen mit den traditionellen Hidden-Markov-Modellen, die in diesem Bereich jahrzehntelang eingesetzt waren, vor allem in Kombination mit handverlesenen Merkmalen. Diese werden mit Modellen verglichen, die die neuesten Entwicklungen auf dem Gebiet des maschinellen Sehens und der Spracherkennung durch das Deep Learning einbeziehen. Deren bessere Leistung wird bestätigt, jedoch werden auch bestimmte Einschränkungen in Bezug auf die Rechenleistung für diese Systeme besprochen.

Diese Doktorarbeit behandelt auch die Verarbeitung mehrerer Kameraansichten und deren Fusion, die ein wichtiges Thema für viele aktuelle Anwendungen sind, weil eine einzige Kameraansicht nicht genügend Flexibilität bietet, wenn die Sprecher sich vor der Kamera bewegen. Technologieunternehmen integrieren inzwischen mehrere Kameras in ihre Produkte, wie Autos, mobile Geräte usw., da die Hardwarekosten sowohl für Kameras als auch für Prozessoren gesunken und gleichzeitig auch Rechenleistungen und die Performance der Algorithmen gestiegen sind. Multikamera- und Multiansicht-Lösungen verbreiten sich dadurch stärker, und daher sollten Algorithmen diese berücksichtigen. In dieser Arbeit stellen wir mehrere Methoden für die Fusion verschiedener Kameraansichten vor, um die Gesamtergebnisse zu verbessern.

Wir können aufzeigen, dass sowohl die Deep Learning-Methoden zur Extraktion von

Merkmalen und zur Modellierung von Sequenzen als auch die Nutzung von ergänzenden Informationen aus mehreren Kameraansichten die Leistung erheblich steigern. Um die Ergebnisse weiter zu optimieren, wäre es nötig, von Videoaufnahmen im Labor hin zu Multiansichts-Daten aus realistischen Szenarien zu wechseln. Außerdem sollten die Erkenntnisse und Modelle in anderen Bereichen wie der audio-visuellen Spracherkennung oder der Untersuchung von Sprachproduktion und Sprechstörungen angewandt werden.

Stichwörter: visuelle Spracherkennung; automatisches Lippenlesen; Verarbeitung mehrerer Kameraansichten; GMM-HMM; Deep Learning („tiefgehendes Lernen“)

Contents

Acknowledgements iii

Abstract (English/Français/Deutsch) v

List of figures xiii

List of tables xv

List of acronyms xvii

Introduction 1

Motivation . . . 1
Thesis outline . . . 2
Contributions . . . 3

1 Background 5

1.1 Visual speech classes . . . 5
1.2 Video preprocessing: face tracking . . . 6
1.3 Traditional approaches . . . 9
1.3.1 Visual feature extraction . . . 9
1.3.2 Sequence modelling . . . 11
1.4 Combined approaches . . . 13
1.4.1 Visual feature extraction . . . 14
1.4.2 Sequence modelling . . . 16
1.5 Deep learning approaches . . . 17
1.5.1 Feature extraction . . . 17
1.5.2 Sequence modelling . . . 17
1.6 Performance metrics . . . 23
1.7 Multi-view visual speech recognition . . . 23
1.8 Databases . . . 25
1.8.1 TCD-TIMIT . . . 25
1.8.2 OuluVS2 . . . 28

2 Traditional approach **31**
2.1 Motivation . . . 31
2.2 Proposed method . . . 32
2.3 Performance analysis . . . 33
2.3.1 The datasets . . . 33
2.3.2 Experimental results . . . 33
2.4 Summary . . . 37

3 Combined approach **41**
3.1 Motivation . . . 41
3.2 Proposed method . . . 42
3.3 Performance analysis . . . 44
3.3.1 The dataset . . . 44
3.3.2 Experimental results . . . 44
3.4 Summary . . . 47

4 Deep learning approach **51**
4.1 Motivation . . . 51
4.2 Proposed method . . . 52
4.2.1 Feature extraction using convolutional neural networks . . . 52
4.2.2 Sequence modelling with recurrent neural networks . . . 55
4.2.3 Decoding with connectionist temporal classification . . . 57
4.3 Performance analysis . . . 57
4.3.1 The dataset . . . 58
4.3.2 Experimental results . . . 59
4.4 Summary . . . 70

5 Multi-view visual speech recognition **73**
5.1 Motivation . . . 73
5.2 Proposed method . . . 74
5.3 Performance analysis . . . 75
5.3.1 The dataset . . . 75
5.3.2 Experimental results . . . 76
5.4 Summary . . . 83

Conclusion **85**
Perspectives . . . 86

Bibliography **89**

Curriculum vitae **99**

List of figures

1.1 Flowchart of the visual speech recognition chain. 5
1.2 Face tracking examples using the SDM-based face tracker. 8
1.3 Sequence modelling with a GMM-HMM system. 13
1.4 Schematic of an MLP. 14
1.5 Schematic of the basic building blocks of a CNN. 16
1.6 Schematic of an LSTM cell. 19
1.7 Schematic of a GRU. 20
1.8 Schematic of a bidirectional RNN 20
1.9 Examples from the TCD-TIMIT database. 26
1.10 Examples from the OuluVS2 database. 28

2.1 Flowchart of the GMM-HMM system used. 32
2.2 Phrase recognition results on OuluVS2 using a simple GMM-HMM model. 34
2.3 Viseme recognition results on TCD-TIMIT using a simple GMM-HMM model. 37
2.4 Confusion matrices of our simple GMM-HMM model for the frontal and 30° views. 38

3.1 The proposed tandem system with PCA network-LSTMs and GMM-HMMs for VSR. 43
3.2 The PCA network used in the first stage of the proposed tandem system. 44
3.3 Phrase recognition results on OuluVS2 using the proposed tandem system with cross-validation. 46
3.4 Phrase recognition results on OuluVS2 using the proposed tandem system on the test set. 47

4.1 Flowcharts of the different CNN architectures. 54
4.2 Flowchart of the RNN network integrated with the CNN. 56
4.3 Histogram of the visemes in the TCD-TIMIT database. 60
4.4 Evolution of loss during training. 61
4.5 Evolution of categorical accuracy during training. 61
4.6 Confusion matrices of the two best CNNs. 64
4.7 Confusion matrix of the best RNN. 66
4.8 Viseme recognition results on TCD-TIMIT using our CNN-RNN with CTC. 68

4.9 Confusion matrices of our CNN-RNN with CTC for the frontal and 30° views. 69

5.1 The proposed feature fusion multi-view tandem system with PCA network-LSTMs and GMM-HMMs for VSR. 75
5.2 Multi-view phrase recognition results on OuluVS2 using feature fusion in the proposed tandem system with cross-validation. 79
5.3 Multi-view phrase recognition results on OuluVS2 using feature fusion in the proposed tandem system on the test set. 80
5.4 Best multi-view phrase recognition results on OuluVS2 using the proposed tandem system. 82

List of tables

1.1 Viseme mapping used in this work. 7
1.2 Overview over the TCD-TIMT and OuluVS2 databases. 26
1.3 Example sentences from the TIMIT dataset. 27
1.4 Training and test splits for the TCD-TIMIT dataset. 27
1.5 List of short phrases from the OuluVS2 dataset. 28
1.6 Training and test splits for the OuluVS2 dataset. 29

2.1 Phrase recognition results on OuluVS2 using a simple GMM-HMM model with silence label. 35
2.2 Phrase recognition results on OuluVS2 using a simple GMM-HMM model without silence label. 36
2.3 Viseme recognition results on TCD-TIMIT using a simple GMM-HMM model. 40

3.1 Baseline sentence recognition results on OuluVS2 by the authors of the database. 45
3.2 Phrase recognition results on OuluVS2 using the proposed tandem system on the test set. 48
3.3 Frame recognition results on OuluVS2 using the LSTM output. 49

4.1 Comparison of model sizes for the different CNN architectures. 53
4.2 Comparison of model sizes for the different RNN architectures. 55
4.3 Framewise viseme accuracy validation results on the frontal view of TCD-TIMIT using the different CNN architectures. 62
4.4 Framewise viseme accuracy test results on the frontal view of TCD-TIMIT using the different CNN architectures. 62
4.5 Framewise viseme accuracy validation results on the frontal view of TCD-TIMIT using the different RNN architectures. 65
4.6 Framewise viseme accuracy test results on the frontal view of TCD-TIMIT using the different RNN architectures. 65
4.7 Viseme recognition test accuracy on the frontal and 30° views of TCD-TIMIT using different RNN architectures with CTC. 66
4.8 Viseme recognition results on TCD-TIMIT using our CNN-RNN with CTC. 71

5.1 Multi-view phrase recognition results on OuluVS2 using feature fusion in the proposed tandem system on the test set. 77
5.2 Optimal weights for the proposed multi-view tandem system. 78
5.3 Best multi-view phrase recognition results on OuluVS2 using the proposed tandem system. 81

List of acronyms

AAM	active appearance model
ANN	artificial neural network
ASR	automatic speech recognition
AVASR	audio-visual automatic speech recognition
BGRU	bidirectional GRU
BLSTM	bidirectional LSTM
BRNN	bidirectional RNN
CNN	convolutional neural network
CTC	connectionist temporal classification
DCT	discrete cosine transform
DNN	deep neural network
DWT	discrete wavelet transform
GMM	Gaussian mixture model
GPU	graphics processing unit
GRU	gated recurrent unit
HCI	human computer interaction
HiLDA	hierarchical LDA
HMM	hidden Markov model
LBP-TOP	local binary patterns from three orthogonal planes
LDA	linear discriminant analysis
LGO	local gradient orientation
LSTM	long short-term memory
MFCC	mel-frequency cepstral coefficient

MLLT maximum-likelihood linear transform

MLP multilayer perceptron

PCA principle component analysis

PDM point density model

RBM restricted Boltzmann machine

RNN recurrent neural network

ROI region of interest

SDM supervised descent method

SIFT scale-invariant feature transform

VSR visual speech recognition

WER word error rate

Introduction

Speech is an important means of human communication, and at its level of complexity it is often considered to be one of the distinctive characteristics between humans and other animals. Speech is also one of the most natural ways of communication for humans, and thus has long been dreamt up in human computer interaction (HCI) to facilitate the interaction between humans and machines and to make it more natural. With advances in signal processing and increasing computational power, algorithms were developed that could transcribe an audio signal to a sequence of phonetically distinct units. However, only with further developments in both technology and algorithms, could the machines recognise a larger vocabulary and eventually react to the commands. Still, the interaction is often frustrating when there are misunderstandings, which can happen particularly in noisy situations. Or it might be undesirable to have your neighbour listen in on your commands, e.g. when spelling out your password. Furthermore, hearing impaired people communicate via sign language, rather than using spoken words. For all these, another branch of speech recognition, using the visible articulations of the mouth, has developed: visual speech recognition (VSR) or automatic lip-reading.

There are many applications to this field, including the examples mentioned above, such as support for the audio in noisy situations through audio-visual automatic speech recognition [Potamianos et al., 2004], silent or whispered speech, e.g. for pronouncing passwords in cybersecurity [Denby et al., 2010, Hassanat, 2014, Petridis et al., 2018], for the mouthings in sign language recognition [Schmidt and Koller, 2013], as well as understanding speech production better [Badin et al., 2002] or for direct HCI.

Motivation

The interest in visual speech recognition is motivated by the way humans work when confronted with a listening task or conversation in a very noisy environment. In situations such as a noisy restaurant humans usually resort to lip reading to improve their understanding. Humans make use of the additional information to distinguish different sounds. When simultaneously listening and lip reading, a sound can even be confused if the wrong mouth movement is shown: this is the McGurk effect [McGurk and MacDonald, 1976].

We can exploit this extra information in audio-based automatic speech recognition as well by combining audio and video modalities.

In this thesis the focus is on pure VSR which can then be used as a starting point for other research problems like the ones mentioned above. The aim of this work is to show the improvements that can be obtained starting from the traditional hand-crafted feature and GMM-HMM systems, via combined approaches using neural networks to extract features while still maintaining the GMM-HMM system, all the way to fully deep learning-based methods with CNNs and RNNs. It also shows the limitations of certain methods, considering the data size and using only a single computer and graphics processing unit (GPU).

A remaining constraint of VSR is the inability to deal with videos coming from different view angles. This is a major problem for applying these algorithms in real-life situations, since in most cases the speaker is free to move his head or even the entire body. For these reasons and due to the availability of cheaper equipment, it is becoming more popular nowadays to increase the number of cameras on a device or in a certain environment. For example, it is becoming more common to integrate at least two cameras into a car to monitor the driver. Here the question of how to treat these different video streams and how to combine their information comes into play. This topic is treated in this thesis, albeit for static views. The choice of databases analysed is also determined by this factor, whether it provides several simultaneously recorded views.

Thesis outline

The thesis is structured in the following way:

- To begin, **Chapter 1: Background** provides an overview of the methods needed and used in VSR. This ranges from the definition of the speech classes, over the face tracking needed to preprocess the videos, to approaches used for visual feature extraction and sequence modelling. In the latter, three types of approaches are elaborated: the traditional approaches, the combined, and the most recent deep learning-based methods. This is followed by an overview of the performance metrics used in this thesis. Finally, a short introduction to multi-view visual speech recognition and the databases explored in this work are given.

- **Chapter 2: Traditional approach** outlines the setup and presents some baseline results performed with handcrafted features (DCT coefficients) with standard GMM-HMM systems.

- In **Chapter 3: Combined approach** a feature extraction system consisting of a PCA network followed by an LSTM is presented. In a tandem system these features are

then passed into a GMM-HMM which models the time evolution. It is shown that this network outperforms the traditional methods presented as a baseline.

- Following the advances in deep learning, **Chapter 4: Deep learning approach** provides a systematic step-by-step approach to developing a deep learning system. It is shown that this method outperforms the previous approaches.
- Taking into account the views from different camera angles, **Chapter 5: Multi-view visual speech recognition** uses the tandem approach to show that the combination can improve the results by integrating the complementary information.
- Finally, **Conclusion** summarises the results presented in this thesis and provides an outlook on future research.

Contributions

The main contributions of this thesis are summarised below:

- Provide an overview over the different approaches from the traditional to deep learning methods.
- Design a novel tandem system composed of PCA networks with LSTM and a GMM-HMM suitable for small databases [Zimmermann et al., 2017a].
- Propose a systematic approach to developing a deep learning system for continuous sequence-to-sequence visual speech recognition.
- Implement a new method to weight the contributions of different views for multi-view visual speech recognition with a tandem system [Zimmermann et al., 2017b].

1 Background

This chapter provides background information on visual speech recognition (VSR) and the methods used in the field, coming from both the speech recognition and computer vision domains. First the visual speech classes used in this thesis are elaborated. This is followed by an overview of face tracking, needed to find the relevant regions of the face. The subsequent sections present various approaches to VSR: from the traditional approaches with handcrafted features and GMM-HMMs via combined approaches to end-to-end deep learning models. For each approach the processing steps for VSR shown in Figure 1.1 are elaborated: first the features are extracted from the input video frames and then the feature sequence is modelled in time to obtain the sequence of output labels. Next, the performance metrics applied in this work are presented and an overview of multi-view lip reading is given. Finally, existing (audio-)visual speech databases are discussed and those used in this thesis are described in more detail.

1.1 Visual speech classes

One of the first decisions to take when building a speech recognition system involving video is the choice of classes to distinguish different sounds or rather articulations for VSR. In audio-based speech recognition the classes are usually phonemes, defined as the smallest distinctive linguistic unit, or phones, the unit of speech sound independent of the language [Coxhead, 2006]. However, in video-based speech recognition several

Parts of this chapter have been published by Zimmermann et al. [2017a,b].

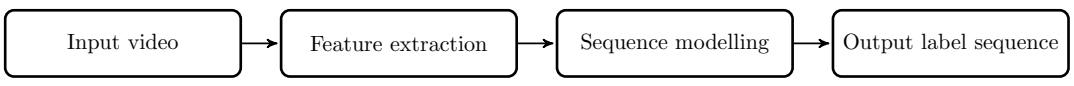


Figure 1.1 – Flowchart of the visual speech recognition chain.

phonemes look the same or similar, so they are usually grouped into visually distinct units called visemes [Cappelletta and Harte, 2012]. Several many-to-one mappings exist from phonemes to visemes. These are usually made up of around 8 to 17 different viseme classes [Coxhead, 2006, Turkmani, 2007, Souviraà-Labastie and Bimbot, 2013]. Even though some research has suggested that visemes provide a sub-optimal classification of speech data [Bear et al., 2014, Yu et al., 2011], some type of visemes or even phonemes are still generally preferred, since they easily relate to phonemes in audio recognition and are thus easy to understand for humans and easily fused in audio-visual automatic speech recognition (AVASR).

In this work we do VSR for English and use the same viseme set as the one used by [Harte and Gillen, 2015], initially proposed in [Jeffers and Barley, 1971]. The choice is based on two factors: first, this mapping has been shown to be reliable [Cappelletta and Harte, 2012]. Secondly, it is used in the baseline results for the database TCD-TIMIT proposed in [Harte and Gillen, 2015], which will be used for part of this work. Using the same viseme set allows better comparisons between different approaches.

However, other types of classes have also been used in the literature. Some recent studies use graphemes rather than phonemes or visemes, which are the different letters, numbers, characters and punctuation marks that appear in a written sentence [Chung and Zisserman, 2017]. For smaller datasets sometimes whole words are used as the smallest unit [Wand et al., 2016]. Finally, some research only tries to distinguish between a set of predefined sentences [Lee et al., 2017, Saitoh et al., 2017].

In contrast to these, in this thesis generally visemes are used as smallest units and sequence-to-sequence decoding is performed, rather than classifying whole utterances. On the smaller one of the two datasets treated in this work, these viseme sequences are then regrouped to word-level models which are then decoded to a sequence of words. For the larger dataset the smallest unit in the sequence decoding are visemes.

1.2 Video preprocessing: face tracking

When treating facial video, it is important to focus on the particular region of interest (ROI) on the face. For VSR, this means extracting the areas which are active during articulation and which thus provide the largest amount of information about the utterance: the mouth, in particular the lips, and possibly other regions such as the cheeks and jaws.

To find these ROIs, newer studies generally apply face trackers that detect the face and find specific landmarks on the face and track these over consecutive frames (an example is shown in Figure 1.2), while older works rely on manually labelling or on markers such as lipstick. The most common choice nowadays for detecting the face is still the Viola-Jones face detector, using features similar to the Haar basis functions [Viola and Jones, 2001].

Table 1.1 – Viseme mapping for English by Jeffers and Barley [1971] as presented by Harte and Gillen [2015] used in this work.

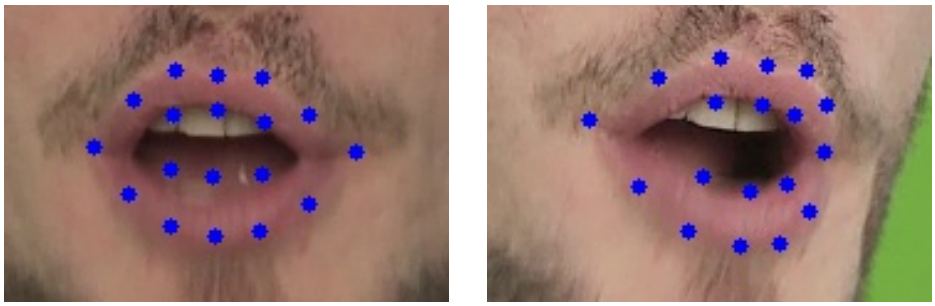
Viseme	TIMIT phonemes	Description
/A	/f/, /v/	Lip to teeth
/B	/er/, /ow/, /r/, /q/, /w/, /uh/, /uw/, /axr/, /ux/	Lips puckered
/C	/b/, /p/, /m/, /em/	Lips together
/D	/aw/	Lips relaxed-moderate opening to Lips puckered-narrow
/E	/dh/, /th/	Tongue between teeth
/F	/ch/, /jh/, /sh/, /zh/	Lips forward
/G	/oy/, /ao/	Lips rounded
/H	/s/, /z/	Teeth approximated
/I	/aa/, /ae/, /ah/, /ay/, /ey/, /ih/, /iy/, /y/, /eh/, /ih/, /iy/, /y/, /eh/, ax-h/, /ax/, /ix/	Lips relaxed narrow opening
/J	/d/, /l/, /n/, /t/, /el/, /nx/, /en/, /dx/	Tongue up or down
/K	/g/, /k/, /ng/, /eng/	Tongue back
/S	/sil/, /pcl/, /tcl/, /kcl/, /bcl/, /dcl/, /gcl/, /h#/, /#h/, /pau/, /epi/	Silence

For subsequent tracking, various types of face trackers exist. One of the most commonly used trackers in VSR is the active appearance model (AAM) [Cootes et al., 2001], since its parameters are sometimes directly used as features.

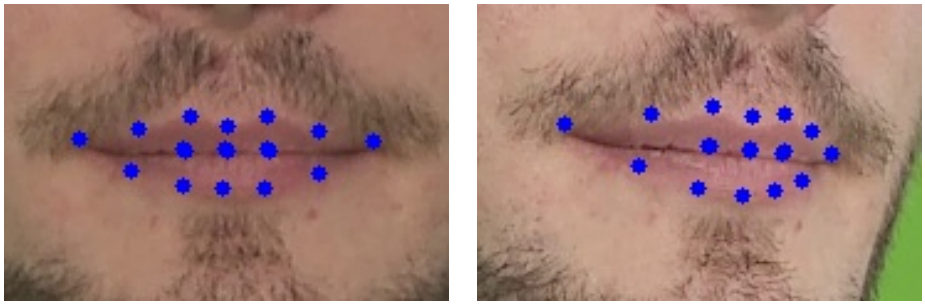
AAMs model both facial shape and appearance. The shape of the AAM is described by the sequence of (x, y) coordinates of the landmark locations in the model: $\mathbf{s} = (x_1, y_1, \ldots, x_n, y_n)$

This shape is parametrised with a principle component analysis (PCA), so that it can be represented as a sum of the mean shape $\mathbf{s_0}$ and its eigenvectors $\mathbf{s_i}$ multiplied by the shape parameters p_i [Lan et al., 2010]:

$$\mathbf{s} = \mathbf{s_0} + \sum_{i=1}^{m} p_i \mathbf{s_i}. \tag{1.1}$$



(a) Frontal view, mouth open. (b) Side view, mouth open.



(c) Frontal view, mouth closed. (d) Side view, mouth closed.

Figure 1.2 – Face tracking and mouth cropping examples with the facial landmarks indicated on frames from the TCD-TIMIT database using the SDM-based face tracker.

Here the m largest eigenvectors are kept, where

$$\mathbf{p} = \mathbf{s}^T(\mathbf{s} - \mathbf{s_0}). \tag{1.2}$$

Similarly, the appearance $\mathbf{A}$ within this region can be described as a linear combination of a decomposition by PCA into mean appearance and eigenvectors with the corresponding appearance parameters q_i:

$$\mathbf{A} = \mathbf{A_0} + \sum_{i=1}^{l} q_i \mathbf{A_i}. \tag{1.3}$$

This is applied to shape-normalised and reshaped images and the l largest eigenvectors are stored. Again,

$$\mathbf{q} = \mathbf{A}^T(\mathbf{A} - \mathbf{A_0}). \tag{1.4}$$

However, more recent face trackers, like the regression-based one using the supervised descent method (SDM) [Xiong and De la Torre, 2013] for fitting, show better performance results for facial landmark tracking [Cuendet, 2017]. This fitting method has been developed to minimise non-linear least squares functions. Unlike the parametrised AAM described above, regression-based face trackers do not use a shape or appearance model,

but learn the vectorial regression function from the image directly on the landmark locations $\mathbf{s} = (x_1, y_1, \ldots, x_n, y_n)$. This allows the tracker to more readily adapt to asymmetric shapes or untrained gestures. The landmark locations are updated iteratively through a sequence of descent directions and rescaling factors ($\mathbf{s}_{k+1} = \mathbf{s}_k + \Delta\mathbf{s}$).

Finally, another difference from the AAM is the use of scale-invariant feature transform (SIFT) features around each landmark location in the SDM. The feature vector ϕ_* contains the collected and ordered features of a particular shape $\mathbf{s}_*$ in an image, and is directly used to compute the necessary updates. Due to its better performance, an implementation of the SDM-based face tracker from our lab, with improvements from [Qu et al., 2015], is used where necessary in this work to crop the ROI.

1.3 Traditional approaches

This section describes the traditional approach to visual speech recognition, which comprises the extraction of texture-based or geometrical features and the consequent modelling of specific speech classes through hidden Markov models (HMMs) with Gaussian mixture models (GMMs). It is structured in the following way: first visual feature extraction techniques and then sequence modelling with a GMM-HMM system are discussed.

1.3.1 Visual feature extraction

After deciding on the type of class and obtaining the correct landmark locations, it is necessary to look into the different features used in visual speech recognition. They can be grouped into two approaches: appearance-based and shape-based features [Potamianos et al., 2004].

Appearance-based features exploit the pixel values in the ROI and apply a sort of transformation to these. Popular image transformations include the PCA, discrete cosine transform (DCT), discrete wavelet transform (DWT), linear discriminant analysis (LDA) and maximum-likelihood linear transform (MLLT) [Potamianos et al., 2004]. These can be applied at different stages of a feature extraction framework.

PCA, DCT and DWT are image transformation methods that compress the information in the ROI, while LDA improves classification by remapping the data to a different feature space where the discriminability is maximised; similarly, MLLT is a maximum likelihood data modelling technique. The latter two can be used not only on appearance-based features but also for post-processing of any kind of feature [Potamianos et al., 2004].

Shape-based features, on the other hand, extract information about the shape of the mouth. As summarised by Potamianos et al. [2004], various shape-related features have

been explored in the literature. This can be done for example with the help of active contour models, or snakes, taking into account the outer contours of the mouth, or by exploiting the geometrical appearance of the mouth: computing distances such as height, width, perimeter, protrusion or area with the help of certain points of interest [Potamianos et al., 2004, Chitu and Rothkrantz, 2008, Koller et al., 2014]. Other methods make use of the lip image moments or Fourier descriptors of the lip contours [Potamianos et al., 2004].

Some other works have also developed specific models of the lips based on parametric or active shape models [Luettin et al., 1996, Gurbuz et al., 2001]. These are used to model the mouth shapes for the different visemes and then to recognise them.

All these methods usually need a first tracking of the lips by a model. As mentioned in section 1.2, a common choice for this is a face tracker, such as the AAM [Cootes et al., 2001]. Many works employ it, either to extract further features from the shape or a ROI defined by the shape, or to directly use its parameters as features [Chitu and Rothkrantz, 2008, Bowden et al., 2013, 2012, Lan et al., 2010, Koller et al., 2014, Potamianos et al., 2004, Bowden et al., 2013, Biswas et al., 2015, Sterpu and Harte, 2017].

Some other researchers have worked with more detailed 3D models of the face and lips [Watanabe et al., 2017]; however, these have more often been used with the aim of recreating the motion for speech synthesis in avatars [Basu and Pentland, 1997, Wei et al., 2004].

In some systems a combination of appearance and shape serves as the feature set used for classification in the end. This can be achieved through simple concatenation of the separate feature sets. Using the AAM for this purpose has showed a better performance than the individual shape or appearance features [Lan et al., 2010].

For instance, the parameters of the shape and appearance decompositions of the AAM can be combined:

$$\mathbf{b} = \begin{bmatrix} \mathbf{Wp} \\ \mathbf{q} \end{bmatrix}, \tag{1.5}$$

where $\mathbf{W}$ is a matrix containing weights for unit adaptation of the shape parameters with respect to the appearance parameters.

Finally, another PCA can be applied to this combined parameter vector to reduce the feature dimensionality and decorrelate the features:

$$\mathbf{b} = \mathbf{Vc}. \tag{1.6}$$

The columns of $\mathbf{V}$ are the first v eigenvectors of the covariance matrix of the vector of combined shape and appearance parameters. $\mathbf{c}$ is then the combined parameter vector of

shape and appearance on which the model is built.

In the end, some temporal information is usually included in the feature vector by either incorporating features for several frames or adding delta and acceleration components.

Before using the features for speech recognition, a few techniques like the aforementioned PCA or LDA and MLLT can also be applied as feature post-processing techniques. This is especially useful for purposes of compression and decorrelation of features as well as to achieve speaker independence [Potamianos et al., 2004, Neti et al., 2000]. In particular, LDA has been intensively studied as a feature post-processing technique with different types of classes: phonemes, visemes, or HMM state sequences [Potamianos and Graf, 1998, Potamianos et al., 2000, Lan et al., 2010]. LDA, in combination with MLLT, is often applied once directly to the appearance-based features and then to a concatenation of several frames. It is then referred to as hierarchical LDA (HiLDA) [Potamianos et al., 2004].

In the traditional approaches, but also in some deep learning frameworks, commonly the delta and acceleration components of the feature sequence are computed and concatenated to the features, to be jointly passed to the sequence model. These coefficients provide additional information about the sequence's dynamics. The delta coefficient d_t at time t is computed in the following way:

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta(c_{t+\theta} - c_{t-\theta})}{2\sum_{\theta=1}^{\Theta} \theta^2}, \tag{1.7}$$

where $c_{t+\Theta}$ and $c_{t+\Theta}$ are the corresponding static coefficients, defined on a window of size $2\Theta + 1$ [Young et al., 2009]. The acceleration coefficients are calculated by applying equation (1.7) to the delta components.

1.3.2 Sequence modelling

In the traditional approaches, GMM-HMM systems are most commonly used for sequence modelling. They have been employed in audio-based speech recognition for decades and allow to model the phonemes, or visemes in the case of video-only speech recognition, with an HMM with integrated GMMs.

The aim of using the HMM, which is a finite state machine that models a sequence in time by states [Rabiner, 1989, Gales and Young, 2007, Young et al., 2009], is to find the most likely label sequence $\hat{\mathbf{Y}}$ given an input, or observation sequence $\mathbf{X}$:

$$\hat{\mathbf{Y}} = \arg\max_{\mathbf{Y}} P(\mathbf{Y}|\mathbf{X}). \tag{1.8}$$

Since the HMM is a generative model, we cannot find $P(\mathbf{Y}|\mathbf{X})$ directly. By applying

Bayes' Theorem, it can be approximated by

$$\hat{\mathbf{Y}} = \arg\max_{\mathbf{Y}} \frac{P(\mathbf{X}|\mathbf{Y})P(\mathbf{Y})}{P(\mathbf{X})} \propto \arg\max_{\mathbf{Y}} P(\mathbf{X}|\mathbf{Y})P(\mathbf{Y}), \tag{1.9}$$

where $P(\mathbf{X}|\mathbf{Y})$ is determined by the acoustic model and $P(\mathbf{Y})$ by the language model [Gales and Young, 2007]. The actual state sequence S producing the series of observations $\mathbf{X}$ is unknown in practice, which is why it is called a *hidden* Markov model. The likelihood $P(\mathbf{X}|\mathbf{Y})$ can further be obtained by marginalising over all possible state, or alignment, sequences

$$P(\mathbf{X}|\mathbf{Y}) = \sum_{\mathbf{S}\in\mathcal{S}} P(\mathbf{X}, \mathbf{S}|\mathbf{Y}). \tag{1.10}$$

This can be expanded to give

$$P(\mathbf{X}) = \sum_{\mathbf{S}\in\mathcal{S}} \prod_{t=1}^{T} P(\mathbf{x}_t|s_t)P(s_t|s_{t-1}), \tag{1.11}$$

omitting the conditioning on $\mathbf{Y}$ for simplicity.

More precisely, as shown in Figure 1.3, each observation or input $\mathbf{x}_t$ at time step t is modelled to be emitted, or drawn, from a probability density – the emission probability $b_j(\mathbf{x}_t) = P(\mathbf{x}_t|s_t = j)$, in this case from the GMM – and the transition from state i to state j is modelled by the so-called transition probability $a_{ij} = P(s_t = j|s_{t-1} = i)$.

The above equation can thus also be rewritten as

$$P(\mathbf{X}) = \sum_{\mathbf{S}\in\mathcal{S}} a_{s_0 s_1} \prod_{t=1}^{T} b_{s_t}(\mathbf{x}_t) a_{s_{t-1} s_t}, \tag{1.12}$$

with the constraint that s_0 be the entry state and s_{T+1} the exit state.

The GMM that models the emissions takes as input the chosen features from the previous section. Each class is then modelled by a mixture of Gaussians in a multi-dimensional space depending on the feature dimensionality. The Gaussians are defined by a particular parameter set, depending on the number of free parameters: mean, covariances and mixture weights [Rabiner, 1989].

To train this model the Baum-Welch algorithm is widely used, which is an iterative method based on the Expectation Maximisation algorithm that updates the different model parameters through reestimation [Rabiner, 1989, Young et al., 2009].

The final decoding of the sequence is then performed by the Viterbi algorithm [Viterbi, 1967, Young et al., 2009]. This algorithm computes the most likely path through a phoneme sequence, namely the maximum likelihood state sequence. Within the Viterbi

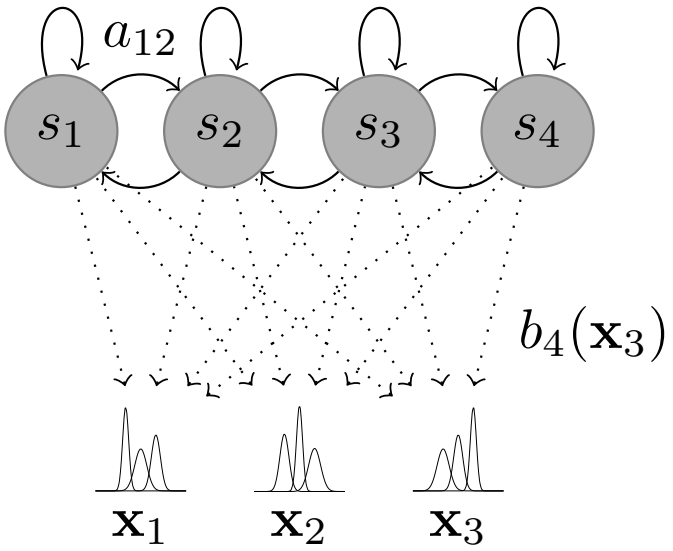


Figure 1.3 – Sequence modelling with a GMM-HMM system.

decoder, certain constraints of a lexicon (pronunciation model) and/or grammar (language model) can be taken into account.

HMMs have been widely used in speech recognition since they allow easy and efficient modelling of sequences of events by states. For a long time they produced the state-of-the-art results in the field.

In this work the Hidden Markov Model Toolkit HTK [Young et al., 2009] is used for all models involving GMMs and HMMs and the decoding.

1.4 Combined approaches

The traditional GMM-HMM approach was widely used until the emergence of artificial neural networks (ANNs) for automatic speech recognition (ASR). With these, the GMM emission models integrated into the HMM temporal modelling scheme were replaced by such new, improved 'acoustic-phonetic' models which led to higher performance [Bourlard and Morgan, 1994]. Whereas in ASR the main use for ANNs was initially these emission models (which was then also repeated with success for visual speech recognition), the bigger impact in this domain was achieved through the developments involving convolutional neural networks (CNNs) and other deep learning methods for feature extraction and directly in end-to-end systems.

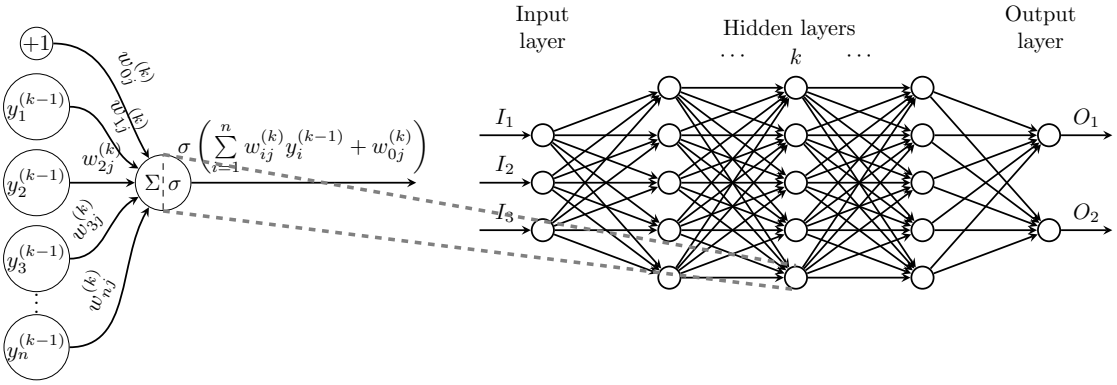


Figure 1.4 – Schematic of an MLP with a sigmoid activation function.

1.4.1 Visual feature extraction

In more recent approaches to VSR, the entire visual feature extraction pipeline has been replaced by specific types of ANNs, usually CNNs or auto-encoders. CNNs are a very common image processing tool nowadays, and they allow to extract features from images.

Artificial neural networks have been inspired by biology in the way brains consist of networks of neurons. These artificial neurons receive and pass raw and processed information (signals) from one to another in an often layered network [Bishop, 2006]. Each neuron has the following characteristics: an activation function $f(\cdot)$ that maps the weighted (by w_{ij}) inputs x_i to a certain output value y_j. These functions are often nonlinear, following their biological models. In this case and when they contain several layers, these networks can also be referred to as multilayer perceptrons (MLPs). The layers of the network that lie between the input and the output layer are often called *hidden* layers since their activations are not observed directly.

The activation $a_j^{(k)}$ of neuron j in layer k with inputs $y_i^{(k-1)}$ in the preceding layer for $i \in \{1, \cdots, n\}$ is given by

$$a_j^{(k)} = \sum_{i=1}^{n} w_{ij}^{(k)} y_i^{(k-1)} + w_{j0}, \tag{1.13}$$

where w_{0j} is also referred to as the bias.

The output of a neuron j in layer k is then

$$y_j^{(k)} = f(a_j^{(k)}), \tag{1.14}$$

with activation function $f(\cdot)$ acting on $a_j^{(k)}$. Figure 1.4 shows a schematic of an MLP with a sigmoid activation function.

ANNs are trained using the backpropagation algorithm which involves the iterative minimisation of a cost, or loss, function. The first stage evaluates the gradient of this cost function with respect to the weights by propagating the error through the network. In the second step the weights are updated by taking into account these derivatives. Generally, this update is limited by a percentage, the *learning rate*, which controls by how much each training sample influences the update and is used to regulate the speed of convergence and the risk of overfitting [Bishop, 2006].

Mathematically, the backpropagation of errors is based on the following formula:

$$\delta_i^{(k)} = f'(a_i) \sum_{j=1}^{n} w_{ij}^{(k)} \delta_j^{(k+1)}, \tag{1.15}$$

where $\delta_i^{(k)}$ is the 'error' corresponding to hidden neuron i in layer k and $\delta_j^{(k)}$ are the 'errors' associated to the preceding (i.e. closer to the output, since the error is propagated backwards through the network) hidden, or output, neurons j in layer $k+1$. $w_{ij}^{(k)}$ are the weights associated to these neurons, and $f'(\cdot)$ is the derivative of the activation function, here evaluated at $a_i^{(k)}$. For the output layer, the 'error' is calculated directly as the difference between the ground truth t_j and the network outputs o_j, i.e. $\delta_j = o_j - t_j$. The 'errors' are then iteratively passed through the network from one layer to the other.

The updates $\Delta w_{ij}^{(k)}$ to each weight $w_{ij}^{(k)}$ can then be obtained by using the gradient descent algorithm – a first-order iterative optimisation algorithm to find the minimum of a function –, making small updates in the direction of the negative gradient:

$$\Delta w_{ij}^{(k)} = -\eta \frac{\partial E}{\partial w_{ij}^{(k)}} = -\eta \delta_j^{(k)} y_i^{(k-1)}, \tag{1.16}$$

where η is the learning rate and E is the loss function. The update of one sample at a time is also referred to as *stochastic* learning, whereas updating after observing a number (the batch size) of training samples is called *batch* learning. For the latter, the individual contributions of each sample in a mini batch are summed up to perform one update after each batch.

Convolutional neural networks (CNNs) are a specific type of ANN based on the concept of receptive fields. In humans and other animals, neurons in the visual cortex process information from a small, specific region in the field of view, the so-called local receptive field. These can be represented by a weight matrix, usually referred to as a filter or kernel, applied to a particular region. In CNNs the weights are often shared between several neurons of a certain layer, thus allowing to recognise similar shapes across the image and to reduce the number of parameters at the same time. This operation can also be seen as a convolution of the neurons of the local receptive field with the filter [Le Cun et al., 1990]. A major advantage of CNNs over MLPs is the fact that unlike the latter, the former's layers are often not fully connected, thus significantly reducing both

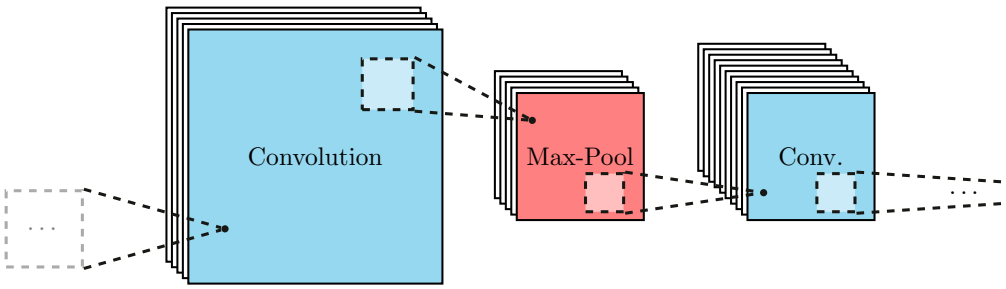


Figure 1.5 – Schematic of the basic building blocks of a CNN.

the training time and the need for training data.

In Figure 1.5 an example of two basic building blocks of a CNN is shown: these are the convolutional layers, in which an input is convolved with each neuron's receptive field, thus creating a new activation matrix for each neuron; and the max pooling layers, which compress the activations by only passing the maximum value for each cluster within a given grid, as shown with a box in Figure 1.5. Another important building block consists in the fully connected layers, which have connections between all neurons, similarly to the neurons in an MLP. The same backpropagation algorithm with gradient descent as mentioned above can be used to train CNNs.

In recent years, deep learning methods with multiple hidden layers between the input and output layers, such as CNNs, have been shown to have superior image and object classification performance [Donahue et al., 2015, Chan et al., 2015] which in turn should also mean that they are better at extracting discriminative features from the image for further processing like in VSR.

In the VSR domain, Ngiam et al. [2011] started by using deep Boltzmann machines for feature extraction in combination with support vector machines to classify the utterances, normalised in length. Other researchers extracted the features with deep belief networks [Huang and Kingsbury, 2013] and CNNs [Noda et al., 2014, Koller et al., 2015].

1.4.2 Sequence modelling

There are multiple ways to use the output of an ANN. One possibility is to simply use the output of these models as an input to the traditional GMM-HMM system, practically like features. This approach is called a tandem system [Hermansky et al., 2000].

However, there is another advantage of using ANNs: Their output represents a probability distribution, which can easily be treated in the subsequent HMM sequence model as the emission probability $P(\mathbf{x}_t|s_t)$ if the posterior probability $P(s_t|\mathbf{x}_t)$ is normalised by the

probability of the state $P(s_t)$. It thus does not require an additional acoustic model. These kind of models are the so-called hybrid approaches [Bourlard and Morgan, 1994].

For VSR tandem systems with a GMM-HMM recogniser have been used in several works [Huang and Kingsbury, 2013, Noda et al., 2014, Sui et al., 2015]. Other research has made use of the hybrid approach [Thangthai et al., 2015, Thangthai and Harvey, 2017]. Mroueh et al. [2015] use another type of combined system by using hand-crafted features, based on scattering coefficients and LDA, and then replace the entire recognition system by a bilinear network.

1.5 Deep learning approaches

With the increasing availability of larger datasets and more powerful computational resources, the latest work in VSR has replaced the whole recognition pipeline by recurrent neural networks (RNNs), such as long short-term memories (LSTMs), on top of features extracted from deep neural networks (DNNs) and more specifically CNNs [Wand et al., 2016, Chung et al., 2017, Petridis et al., 2017a]. Thus, VSR has taken advantage of and joined the recent advances in computer vision and speech recognition.

1.5.1 Feature extraction

In these deep learning frameworks, the feature extraction is typically performed using a specific type of neural network, such as an auto-encoder framework, or the CNN described in the previous section. Often it is trained in sequence, within one large network, together with the RNN, thus resulting in an end-to-end deep learning framework. The features are therefore not analysed on their own, but get evaluated through the overall system.

1.5.2 Sequence modelling

Similarly to other types of artificial neural networks, recurrent neural networks are made up of neurons, or cells. However, contrary to those other networks, a given cell in an RNN receives as input not only the activations from other nodes, but also the outputs from a previous sample's pass through the network, as well as the same cell's so-called *state* from this previous pass. The influence of each of these factors on the cell's new output is determined by a set of weights.

The most common type of RNN is made up of long short-term memory (LSTM) cells. In this type of network the cell is comprised of three gates (see Figure 1.6): an input, a forget and an output gate [Hochreiter and Schmidhuber, 1997, Olah, 2015]. At the different gates, the input, the output and the cell state from the previous timestep are weighted with learned matrices. Thus the cell's new output and state are a function of

these three entry values.

The input gate receives both the current input vector x_t and the previous output vector y_{t-1}:

$$i_t = \sigma(W_i \cdot [y_{t-1}, x_t] + b_i), \tag{1.17}$$

$$\tilde{C}_t = \tanh(W_C \cdot [y_{t-1}, x_t] + b_C), \tag{1.18}$$

where $\cdot$ represents a matrix multiplication, here with weight matrices W_i, and W_C and b_i and b_C are the bias terms, of the input gate and candidate value, respectively.

The candidate values $\tilde{C}_t$ are combined with the output from the forget gate f_t

$$f_t = \sigma(W_f \cdot [y_{t-1}, x_t] + b_f), \tag{1.19}$$

where W_f is the forget gate's weight matrix and b_f its bias, to produce the new cell state:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t, \tag{1.20}$$

where $*$ represents an element-wise multiplication.

Finally, the output gate computes the output weight o_t and, in combination with the cell state, gives the new cell output y_t:

$$o_t = \sigma(W_o \cdot [y_{t-1}, x_t] + b_o), \tag{1.21}$$

$$y_t = o_t * \tanh(C_t), \tag{1.22}$$

with the output gate's weight matrix W_o and bias b_o.

Another important type of RNN is the gated recurrent unit (GRU) [Cho et al., 2014]. Similar to the LSTM, the design of GRUs is based on gates (see Figure 1.7). However, to reduce the number of parameters, the input and forget gates are combined, as are the output and cell states. Furthermore, there is no nonlinearity when computing the output. The lower number of parameters makes it an attractive choice for smaller datasets. In the following, we thus describe the internal workings of a GRU:

The update gate z_t at time t determines what will be retained from the previous memory state y_{t-1}, by also taking into account the current input x_t

$$z_t = \sigma(W_z \cdot [y_{t-1}, x_t]), \tag{1.23}$$

where $\cdot$ represents a matrix multiplication, here with W_z, the update gate's weight matrix.

The other gate, the reset gate r_t, decides how to combine the current input with the

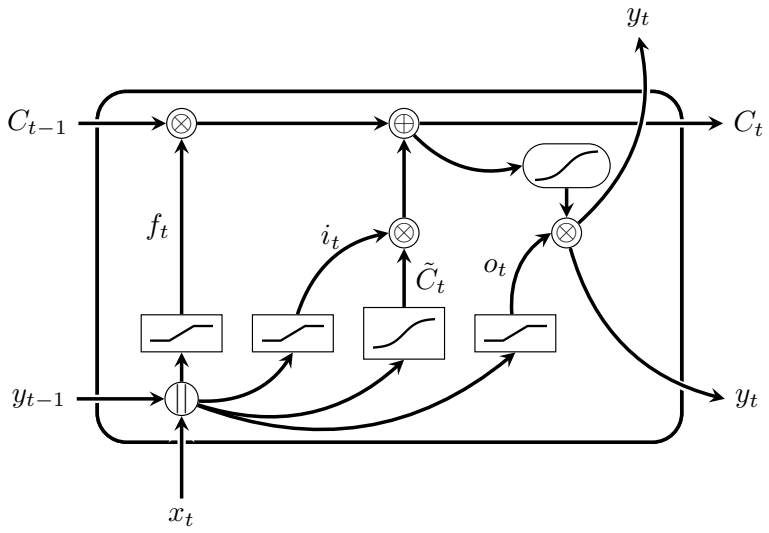


Figure 1.6 – Schematic of an LSTM cell.

previous memory:

$$r_t = \sigma(W_r \cdot [y_{t-1}, x_t]), \tag{1.24}$$

with the reset gate's weight matrix W_r.

The new memory state and the unit's activation are obtained in two steps. First, the hidden state $\tilde{y}_t$ is calculated:

$$\tilde{y}_t = \tanh(W_h \cdot [r_t * y_{t-1}, x_t]), \tag{1.25}$$

where $*$ represents an element-wise multiplication and W_h is the hidden state's weight matrix.

Second, the final current output and cell state y_t is given by

$$y_t = (1 - z_t) * y_{t-1} + z_t * \tilde{y}_t. \tag{1.26}$$

In visual speech recognition, researchers have used both GRUs [Assael et al., 2016, Xu et al., 2018] and LSTM cells [Chung et al., 2017, Stafylakis and Tzimiropoulos, 2017] successfully to model the temporal evolution. They are trained using gradient descent and backpropagation (see Section 1.4.1).

In recent literature, so-called bidirectional RNNs [Schuster and K. Paliwal, 1997] have been employed very often to improve predictability. Each bidirectional layer consists of two recurrent layers which take as input the sequence in its forward and backward direction, as shown in Figure 1.8. This allows learning connections between various

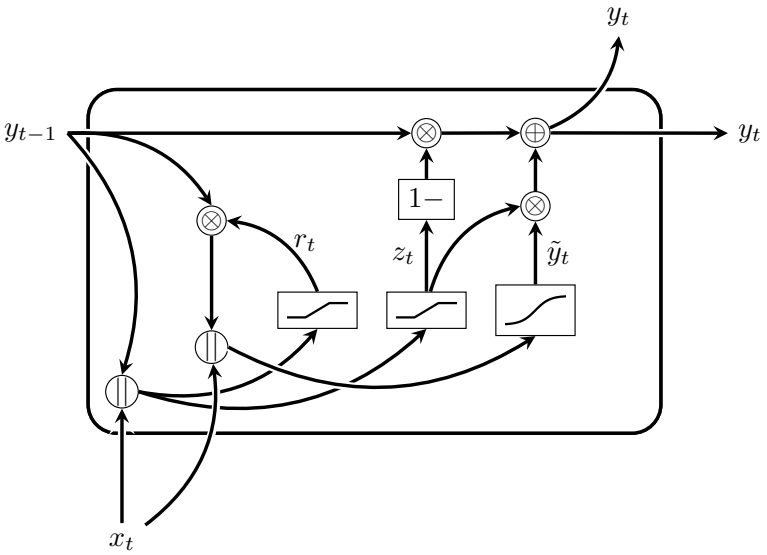


Figure 1.7 – Schematic of a GRU.

elements in both directions, taking into account the previous as well as the future contexts [Graves and Jaitly, 2014]. This advantage has been exploited, among others, by Petridis et al. [2017a], Xu et al. [2018].

The work presented in this thesis compares the results of networks with uni- and bidirectional GRUs and LSTM cells.

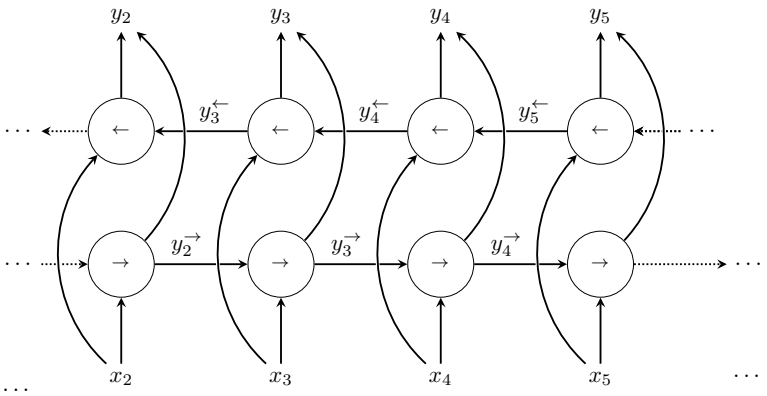


Figure 1.8 – Schematic of a bidirectional RNN.

Decoding

The main issue with the RNN outputs is that classification is still framewise whereas the goal in speech recognition is to label a sequence. Similar to the decoding of HMMs using the Viterbi decoder, connectionist temporal classification (CTC) scoring function determines the joint conditional probability of an overall sequence at given timestamps [Graves, 2008]. In doing so, it functions as an additional layer that can receive as input the framewise output from the bidirectional GRU or LSTM of a certain (variable) length and can output a sequence of symbols (in our case phonemes or visemes, the visual equivalent) of a different (variable) length. For this, CTC does not require time-aligned labels – the overall sequence is sufficient, meaning that it does not depend on the accuracy of the labelling of the training set. The output will also only reflect the overall sequence, and not be evaluated by its timings.

Similarly to HMMs, the CTC algorithm will need to add an extra "blank" class to the labels. This class "collects" all types of silence or other non-speech occurrences in the utterance. It will generally appear between labels which can also help in case of the consecutive occurrence of the same label.

In mathematical terms, the decoding of the network can be described by the maximisation of the following function [Graves and Jaitly, 2014]:

$$\arg\max_{\boldsymbol{Y}} \mathrm{P}(\boldsymbol{Y}|\boldsymbol{X}) \approx \mathcal{B}(\arg\max_{\boldsymbol{S}} \mathrm{P}(\boldsymbol{S}|\boldsymbol{X})), \tag{1.27}$$

where the input sequence $\boldsymbol{X}$ is transcribed by the label sequence $\boldsymbol{Y}$. This can be approximated by using the alignment $\boldsymbol{S}$, related to the transcription $\boldsymbol{Y}$ through the sum of all possible alignments or states $\boldsymbol{S}$,

$$\mathrm{P}(\boldsymbol{Y}|\boldsymbol{X}) = \sum_{\boldsymbol{S}\in\mathcal{B}^{-1}(\boldsymbol{Y})} \mathrm{P}(\boldsymbol{S}|\boldsymbol{X}), \tag{1.28}$$

where the operator $\mathcal{B}$ removes all repeated labels and blanks.

The probability of the alignment itself is the combined probability of the emission probabilities of the alignments s_t at all time steps $t \in \{1, \ldots, T\}$ which, assuming their independence, is given by:

$$\mathrm{P}(\boldsymbol{S}|\boldsymbol{X}) = \prod_{t=1}^{T} \mathrm{P}(s_t, t|\boldsymbol{X}). \tag{1.29}$$

Decoding can be performed either in a greedy manner, using best path decoding, or by taking into account all possible sequences, using prefix search decoding. Best path

decoding finds the particular path with the highest probability:

$$S^* = \arg\max_S \prod_{t=1}^{T} \mathrm{P}(s_t, t|\boldsymbol{X}), \tag{1.30}$$

where S are the possible alignments and S^* is the most probable alignment.

However, this method does not take into account the fact that there might exist several paths which – through repetitions of symbols or intermediate blanks – are ultimately identical. Prefix search decoding also considers these similar paths – which in turn increases the search space exponentially with the length of the input sequence. If certain conditions – such as sufficiently peaked output distributions, or an adapted beam search to only take into account the more promising paths, or additional constraints through a language model – are met, then prefix search decoding still remains feasible [Graves et al., 2006, Hannun, 2017].

Training of the CTC layer functions similarly to HMM training with a forward-backward algorithm based on maximum likelihood. Therefore, the aim is to minimise the objective function based on the negative log probability that the whole training set Z is labelled correctly (with target transcriptions $\boldsymbol{Y}^*$):

$$O(Z, \mathcal{N}_w) = -\sum_{(\boldsymbol{X}, \boldsymbol{Y}^*) \in Z} \ln(\mathrm{P}(\boldsymbol{Y}^*|\boldsymbol{X})), \tag{1.31}$$

where $\mathcal{N}_w$ is the neural network to be trained [Graves et al., 2006]. It is thus indirectly also related to the edit distance or label error rate (see Section 1.6).

When using gradient descent in the training step to update the network with the help of the commonly used backpropagation algorithm (see Section 1.4.1), the derivative of this objective function with respect to the network output has to be taken (see [Graves et al., 2006] for more details).

Other methods for sequence decoding of network outputs have been developed in recent years. One of these decoding methods uses an RNN transducer built on top of a CTC. It combines the CTC-style network with another, separate RNN which acts as a joint acoustic and language model to predict each phoneme given the previous ones [Graves et al., 2013]. This thus results in additional hidden layers for the CTC network, and increases the number of parameters of the model.

While in audio speech recognition these sequence modelling techniques for RNNs have been widely used for several years, they are relatively new in VSR. CTC has been used by Assael et al. [2016], Koumparoulis et al. [2017], Xu et al. [2018].

Another recent sequence modelling technique is the so-called attention scheme [Bahdanau et al., 2016, Chan et al., 2016] for encoder-decoder based systems. It acts between the

encoder and decoder as a sort of weighting which depends on a normalised score of the previous state, the current encoded sequence and the convolutional features. A language model can still be added with final state transducers or in the beam search. Some more recent work combines both CTC and the attention scheme [Kim et al., 2017, Xu et al., 2018].

The work presented in this thesis uses the CTC scheme for sequence decoding since this scheme fits best with the CNN-RNN network and the database used.

1.6 Performance metrics

In this thesis, two metrics are used to present the results: the accuracy and correctness at the viseme or word level, and the percentage of correct sentences. The accuracy and correctness are defined as follows

$$\text{Accuracy} = \frac{H - I}{N} \cdot 100\%, \tag{1.32}$$

$$\text{Correctness} = \frac{H}{N} \cdot 100\%, \tag{1.33}$$

where H, I, and N are the number of correct symbols (visemes or words), number of erroneous symbols (insertion error), and the total number of symbols in the reference, respectively. The number of correct symbols is equal to the number of all symbols minus the total number of ignored symbols (deletion error, D) and the number of wrongly recognised symbols (substitution error, S), i.e. $H = N - D - S$. The accuracy also penalises insertions and is thus related to the edit distance, given by $I + D + S$.

Some of the literature evaluates speech recognition results in terms of another metric related to the edit distance: the word error rate (WER) – or sometimes phoneme/label error rate – rather than the accuracy, which can be obtained from the word accuracy in the following way:

$$\text{WER} = \frac{I + D + S}{N} \cdot 100\% = 100\% - \text{Accuracy}. \tag{1.34}$$

1.7 Multi-view visual speech recognition

While audio-based speech recognition has improved significantly over the past decades and is nowadays applicable in many real-life scenarios, visual speech recognition still mostly focuses on speech produced in controlled lab conditions. However, there is a lot of interest to address, for example, the problem of head pose, which is a large hindrance in the application to real-world scenarios. Early work on multi-pose or non-frontal automatic lip reading focused on using classifiers for each different head pose and defining

a linear transformation or the regression of appearance-based features such as the DCT or a subsequently applied LDA [Lucey et al., 2007, Estellers and Thiran, 2012]. Other researchers worked on cross-view training/testing, i.e., training on one view and testing on another [Lan et al., 2012]. Lan et al. [2012] also established that the 30° view angle provides the best recognition performance, even over the frontal view. Some recent research has applied cross-view analysis to 3D-AAMs [Watanabe et al., 2017] and used channel, image and feature fusion for multiple- and cross-view analysis [Lee et al., 2017].

In [Bowden et al., 2013], three different view angles are explored. Two ways to detect this angle are tested: (i) smallest root-mean-square error of three face trackers for each angle, and (ii) shortest distance of the local gradient orientation (LGO) histogram which performs better since it does not depend on the tracker results.

A method to remove off-plane head rotation has been used by Koller et al. [2014]. Here the shape is registered to a frontal view with a 3D point density model (PDM) trained with a non-rigid structure-from-motion algorithm. However, it is not clear up to which view angles this algorithm is functional. Watanabe et al. [2017] use 3D-AAMs for the same purpose of cross-view VSR.

Nowadays, in many applications, such as driver monitoring, the advantages of using several cameras to cover a larger field of view outweigh the cost of systems with multiple cameras and additional computational resources. Several cameras thus allow to integrate the views to have more confident results, as well as a larger variety of head poses.

Recently, several researchers in this field have also joined different views at various levels of the processing pipeline. In Navarathna et al. [2013], a synchronous HMM was built to include four different views (centre left, centre right, side left, side right). The weights for this multi-stream HMM are determined empirically by comparing the training performance between the centre and the side views and the left and right views for varying weights. The final individual weights are a combination of these coarser weights. Lee et al. [2017], Petridis et al. [2017b] use the OuluVS2 database with five different views [Anina et al., 2015]. Lee et al. [2017] use a combination of CNNs and LSTMs with a final layer that provides the probability that one out of ten different phrases has been uttered. They show results for different types of combinations: 3D-CNN, merging channels, merging images and concatenating features at the output of the CNN. Finally, Petridis et al. [2017b] propose an end-to-end deep learning system made up of restricted Boltzmann machines (RBMs) and bidirectional LSTMs (BLSTMs) where the views are fused between two layers of the BLSTM. They also provide a comparison between the multi-view results from several researchers. Also in the latter system classification is performed at the sentence-level.

1.8 Databases

Recent studies on visual speech recognition have seen VSR moving further and further towards deep learning, a technique widely employed in audio speech recognition as well as computer vision. DNNs have set a new baseline in speech recognition [Graves and Jaitly, 2014] and are now the standard in computer vision tasks [Donahue et al., 2015, Chan et al., 2015]. Since one of the main requirements for deep learning is having a large amount of data, the need for bigger databases is increasing. There exist several databases publicly available to the research community, such as the BBC-Oxford 'Lip Reading in the Wild' (LRW) dataset [Chung and Zisserman, 2017], containing up to 1000 repetitions of 500 different words uttered by hundreds of different speakers, and the GRID audio-visual sentence corpus [Cooke et al., 2006], made up of sentences based on an artificial grammar from 34 speakers. These databases have been widely used in the recent VSR literature that employ deep learning. A few other fairly large audio-visual databases have been developed and published that contain more diverse sentences and various view angles, namely TCD-TIMIT and OuluVS2 [Harte and Gillen, 2015, Anina et al., 2015]. However, the size of these databases still remains rather small compared to audio-only databases or image datasets.

There are numerous other databases in the field, some smaller, some larger, and several in languages other than English. Just to mention some of them: In English there are the CUAVE dataset [Patterson et al., 2002], AVICAR [Lee et al., 2004], an Australian English corpus [Burnham et al., 2009] and the Modality database [Jachimski et al., 2017]. In Spanish there are AV@CAR [Ortega et al., 2004] and the Visual Lip-Reading Feasibility Database [Fernandez-Lopez et al., 2017]. Other languages covered are Czech [Trojanová et al., 2008], Polish [Vorwerk et al., 2010], Turkish [Topkaya and Erdogan, 2012] Russian [Ivanko et al., 2017], Mandarin [Su et al., 2017] and Japanese [Yasui et al., 2017]. Closely related are also a few databases focusing on silent speech [Freitas et al., 2014, Petridis et al., 2018].

In this work we use two databases: TCD-TIMIT and OuluVS2. They were chosen for three reasons: they are publicly available to researchers, they include multiple, simultaneously recorded views and they are relatively large compared to other audio-visual databases.

1.8.1 TCD-TIMIT

The TCD-TIMIT database [Harte and Gillen, 2015] has been collected at Trinity College Dublin and is based on the TIMIT sentences [Garofolo et al., 1993] which are made up of some accent-highlighting sentences, "non-sense" phrases specifically designed to be phonetically rich and sentences from playwrights' books with unusual phoneme contexts. Some example sentences are shown in Table 1.3. The 6913 sentences have been uttered by 62 speakers – among whom three are so-called professional lipspeakers with an

Table 1.2 – Overview over the parts of the TCD-TIMT and OuluVS2 databases used in this work.

	TCD-TIMIT (Irish-speaking volunteers)	OuluVS2 (short phrases)
Speakers	56	52
Accents	Irish	Various non-native
Gender	29 male, 27 female	39 male, 13 female
Utterances	TIMIT sentences	Short English phrases
Utterances per speaker	98	3×10
Amount of speech	7h:0m:36s	0h:19m:26s
Vocabulary	6224	20
Views	0°, 30°	0°, 30°, 45°, 60°, 90°
Video framerate	30 fps	30 fps
Video resolution	1920×1080 pixels	1920×1080 pixels

(a) Frontal view. (b) 30°-side view.

Figure 1.9 – Examples for the different view angles from the TCD-TIMIT database.

Table 1.3 – Example sentences from the TIMIT dataset.

TIMIT example sentences
She had your dark suit in greasy wash water all year.
Never happier in my life.
He will allow a rare lie.
Special task forces rescue hostages from kidnappers.
Are your grades higher or lower than Nancy's?
Did dad do academic bidding?
We experience distress and frustration obtaining our degrees.
Kindergarten children decorate their classrooms for all holidays.
Curiosity and mediocrity seldom coexist.
My mother was beside herself with curiosity.

Table 1.4 – Training and test splits for the TCD-TIMIT dataset.

Training subjects	Test subjects
01M, 02M, 03F, 04M, 05F, 06M, 07F, 10M, 11F, 12M, 13F, 14M, 16M, 17F, 19M, 20M, 21M, 22M, 23M, 24M, 26M, 29M, 30F, 31F, 32F, 37F, 38F, 39M, 40F, 42M, 43F, 46F, 47M, 48M, 50F, 51F, 52M, 57M, 59F	08F, 09F, 15F, 18M, 25M, 28M, 33F, 34M, 36F, 41M, 44F, 45F, 49F, 54M, 55F, 56M, 58F

accentuated articulation. These three speakers, as well as three other speakers with non-Irish accents, have been excluded from our study as suggested by the authors of the database. Each of the remaining subjects pronounced 98 sentences, out of which two are always the same: the ones showing the subject's accent. The rest is made up of different sentences from the other two categories (nonsense phrases and playwrights' book sentences). The training and test data splits as provided by the authors of the database are shown in Table 1.4: the training set is made up of 39 subjects and the test set of 17 subjects. During training we split the training set further into training and validation sets.

Two views, frontal and at a 30° angle, have been recorded simultaneously, and have also been synchronised with the audio. The two cameras each recorded at a framerate of 30 fps (frames per second) with a resolution of 1920×1080 pixels. This database includes already cropped mouth ROIs only for the frontal view. Since we use both views in our work, we use our SDM-based face tracker, see Section 1.2, to extract the mouth region.

(a) Frontal view. (b) 30°-side view. (c) 45°. (d) 60°. (e) 90°.

Figure 1.10 – Examples of the cropped mouths for the different view angles from the OuluVS2 database.

Table 1.5 – List of the short phrases from the OuluVS2 dataset.

Short phrases
Excuse me
Goodbye
Hello
How are you
Nice to meet you
See you
I am sorry
Thank you
Have a good time
You are welcome

1.8.2 OuluVS2

We use the 'phrase recognition' subset of the OuluVS2 database [Anina et al., 2015] in our experiments. This dataset contains video clips of 52 subjects from five different views: frontal and four side views at 30°, 45°, 60°, and 90° (the profile). During each recording session, the subjects were asked to utter 10 daily short English phrases (see Table 1.5) shown on a computer monitor. Each phrase was repeated three times resulting in 30 video recordings (utterances) per subject per view. The recording was performed in an ordinary office environment with varying lighting conditions and background noises producing a more real-world audio-visual dataset. Each of these videos was recorded with a resolution of 1920×1080 pixels, at 30 fps, and with an audio bit rate of 128 kbps (kilobits per second). The authors also provided aligned and cropped mouth videos along with the original videos and fixed the training and test subsets: 40 out of 52 subjects are assigned for training and the rest are used for testing, as shown in Table 1.6. During training we split the training set further into training and validation sets.

Table 1.6 – Training and test splits for the OuluVS2 dataset.

Training subjects	Test subjects
1, 2, 3, 4, 5, 7, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 27, 28, 29, 31, 32, 33, 35, 36, 37, 38, 39, 40, 41, 42, 45, 46, 47, 48, 50, 52	6, 8, 9, 15, 26, 30, 34, 43, 44, 49, 51

2 Traditional approach

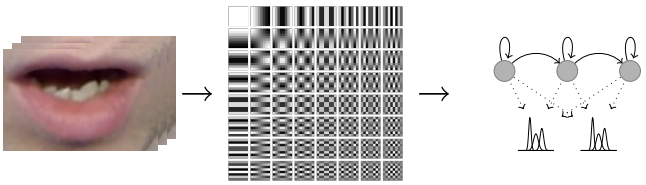
Some traditional approaches to visual speech recognition have been described in chapter 1. This chapter presents some results that were obtained by running experiments using these traditional methods: a DCT is first used to extract texture-based features from the ROI, which are then passed to a GMM-HMM system. Such a setup has been used in the literature very often and thus provides a good baseline system to which other results can be compared. This chapter presents a comparison between the results for the different view angles that are provided for the OuluVS2 and TCD-TIMIT databases and as will be seen in the later sections, there are particularly large differences in recognition performance between the frontal and other view angles for the OuluVS2 dataset. Performance results are very similar for the frontal and 30° side-view for the TCD-TIMIT dataset.

2.1 Motivation

The aim of this chapter is to provide a baseline for the analysis of the two databases, OuluVS2 and TCD-TIMIT, by applying traditional VSR techniques. The techniques applied here were popular methods to VSR for a long time, and involve using a texture-based feature extractor, followed by a GMM-HMM. The systems presented in this chapter are designed based on existing methods, without further improvements and parameter tuning.

A first analysis of the different views from each of the two databases, OuluVS2 (short phrases) and TCD-TIMIT (sentences, viseme recognition) is performed. This gives an indication for later analyses regarding the performance of different views, and also how it might vary between different recognition systems.

The remainder of the chapter is organised as follows: first, the proposed method, based on traditional approaches to feature extraction and sequence modelling, is presented in Section 2.2. Then Section 2.3 describes the two databases and the results obtained on



Mouth frames DCT features GMM-HMM

Figure 2.1 – Flowchart of the GMM-HMM system used.

each. Finally, Section 2.4 summarises the chapter.

2.2 Proposed method

The visual speech recognition pipeline is made up of several steps: first, the face is detected and tracked in each frame and the mouth area is cropped, then features are computed from this mouth area and, finally, these are passed to an acoustic modelling scheme made up of GMMs and HMMs. Figure 2.1 shows this process.

This work is based on texture-based features, namely the DCT. The DCT is related to the Fourier transform and extracts different frequency components from a signal. It is widely used in image processing and especially image compression. This transform is applied to the cropped out mouth images. The DCT coefficients obtained from each of these are then sorted in zig-zag order from the lowest frequency components, and only a certain number, in this case 44, of these coefficients from odd columns – thereby enforcing vertical symmetry – are retained, also excluding the DC component. This feature extraction process is programmed in C++ with the help of the OpenCV library[1].

These coefficients and their delta and acceleration components (see Section 1.3.1 for definition) are passed to a GMM modelling the emissions of the classes. These classes are then modelled in time by an HMM. Finally, the sequences are decoded in a Viterbi scheme. These models are obtained through the HMM toolkit (HTK) [Young et al., 2009].

[1] https://opencv.org/

2.3 Performance analysis

2.3.1 The datasets

Experiments were performed on both the OuluVS2 and the TCD-TIMIT datasets (see Sections 1.8.2 and 1.8.1 for more detail).

2.3.2 Experimental results

OuluVS2 – short phrases

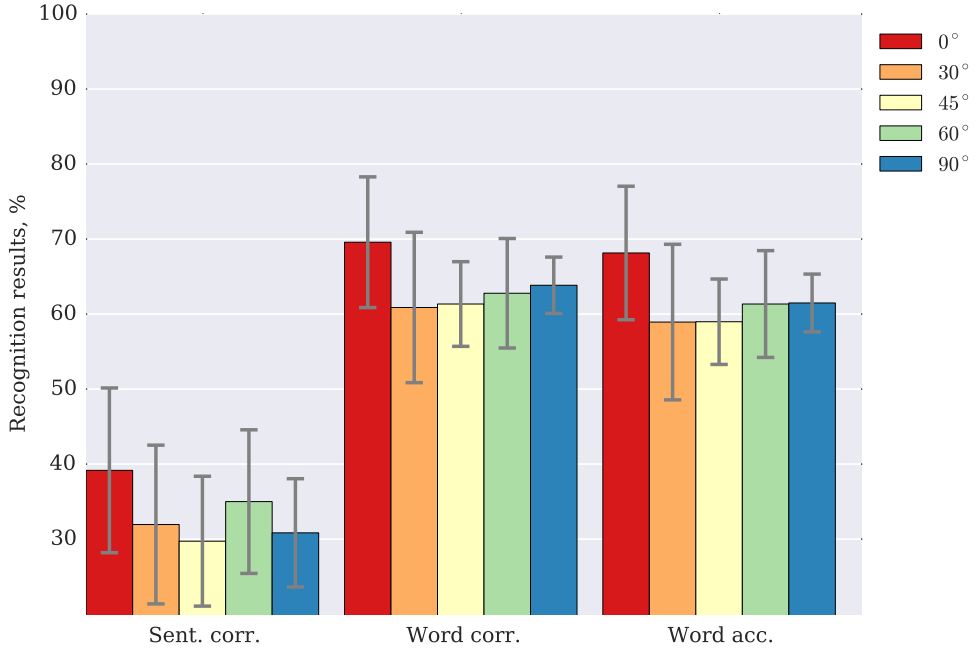
We used the cropped mouth videos provided by the authors of the database. These mouth images were first converted to greyscale before the DCT coefficients were extracted.

Since the phrases are indeed very short and the dataset only contains 3 repetitions of 10 short English phrases (see Table 1.5), in this experiment the GMMs modelled words rather than visemes or phonemes. Each GMM was made up of 15 mixture components and each of the 20 words was modelled by 4 HMM states. A dictionary and grammar corresponding to the phrase set were used as acoustic and language models to constrain the decoding space.

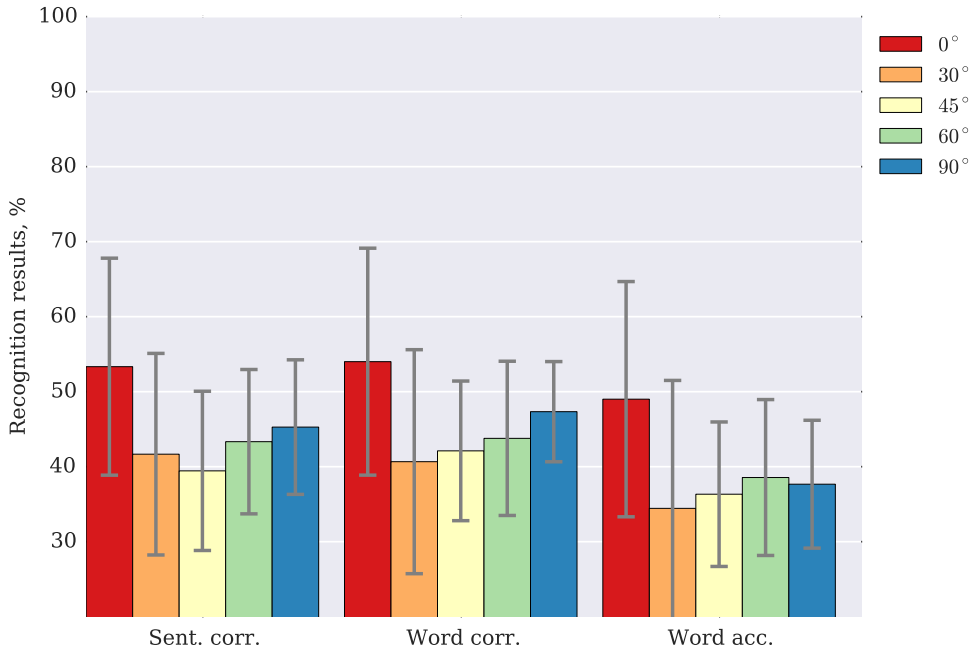
All parameter tuning was performed on a leave-one-out cross validation across speakers of the training set.

The results for the different view angles are shown in Figure 2.2, and in more detail in Tables 2.1 and 2.2. The metrics as explained in Section 1.6 are used, namely the sentence correctness, word correctness and word accuracy. Furthermore, two different sets of results are shown: the first set with the exact labels provided by the models trained with the expectation-maximisation algorithm in HTK, and the second set of results when scoring without considering the "silence" label. The aim of this is to check the influence of the silence label on the results, especially on the sentence-level correctness, since it will indicate how well the model generalises to silence and to the other labels, and whether misclassification of silence could lead to a lower sentence correctness.

In the results we can see that, indeed, removing the silence label has a big impact on the sentence correctness. We note that for certain sentences, the only error is a missing or added silence label between the words. However, we can also observe that the overall word accuracy and correctness are reduced significantly, by around 15-20% when excluding silence labels. We can thus conclude that silence is classified correctly over-proportionally, and that there are more errors in the classification between words. Similarly, a higher between-speaker variability can be observed for the non-silence classes, when looking at the results without silence labels, which shows that there is less generalisability of the models.



(a) Results including the silence label.



(b) Results without the silence label.

Figure 2.2 – Mean phrase recognition results on the multi-view dataset of OuluVS2 using a simple GMM-HMM model on the given test set (a) including the silence label, and (b) without the silence label. Error bars denote the standard deviation across subjects (sent. = sentence, corr. = correctness, acc. = accuracy).

Table 2.1 – Phrase recognition results (in %) on the multi-view dataset of OuluVS2 using our simple GMM-HMM model on the given test set per speaker and the corresponding mean and standard deviation across speakers **including the silence label** (SC = Sentence correctness, WC = Word correctness and WA = Word accuracy).

	0°			30°			45°			60°			90°		
Spkr.	SC	WC	WA	SC	WC	WA	SC	WC	WA	SC	WC	WA	SC	WC	WA
6	40.0	62.8	61.7	43.3	72.2	68.9	40.0	70.0	66.7	30.0	63.3	60.6	36.7	65.0	62.2
8	36.7	65.6	65.6	26.7	55.0	53.9	26.7	58.9	58.9	30.0	58.9	58.9	26.7	63.3	61.1
9	30.0	53.3	52.2	30.0	60.0	58.3	23.3	57.2	53.3	33.3	53.3	51.7	23.3	60.0	56.7
15	26.7	66.1	62.8	26.7	58.9	58.3	16.7	52.8	51.1	26.7	55.0	54.4	26.7	55.6	55.6
26	33.3	64.4	62.8	36.7	64.4	62.2	26.7	63.9	62.2	26.7	64.4	63.3	40.0	66.1	65.0
30	53.3	78.3	74.4	23.3	61.7	57.8	20.0	60.6	55.0	26.7	63.9	60.6	13.3	60.0	56.1
34	43.3	69.4	69.4	43.3	68.3	65.6	33.3	58.3	54.4	43.3	68.9	66.7	36.7	63.3	62.8
43	56.7	81.1	80.0	36.7	61.7	61.1	46.7	68.3	66.7	46.7	71.7	71.1	36.7	68.3	68.3
44	43.3	85.0	85.0	53.3	82.2	82.2	33.3	71.7	69.4	56.7	76.1	74.4	30.0	65.6	60.6
49	50.0	75.0	75.0	26.7	54.4	52.2	40.0	60.6	58.3	30.0	51.1	50.0	36.7	63.3	60.6
51	16.7	61.7	59.4	23.3	48.9	47.2	26.7	56.7	55.6	26.7	58.9	58.3	33.3	65.6	62.2
52	40.0	72.2	69.4	13.3	42.8	39.4	23.3	57.2	56.1	43.3	67.8	66.1	30.0	70.0	66.7
Mean	39.2	69.6	68.1	31.9	60.9	58.9	29.7	61.3	59.0	35.0	62.8	61.3	30.8	63.8	61.5
SD	11.5	9.1	9.3	11.1	10.5	10.8	9.0	5.9	5.9	10.0	7.6	7.4	7.5	3.9	4.0

Comparing the performance between different views, we can easily see that the best performing view is the frontal view. The next highest performing view is the profile view, followed closely by the 60°, 45° and 30° views. Looking at the performance across views and across speakers, we can also notice that the closer the view is to the profile view at 90°, the lower is the standard deviation between the results for different speakers.

TCD-TIMIT

For the TCD-TIMIT database the mouth ROIs extracted with our SDM-based face tracker are used. The image colours in this region are then modified using histogram equalisation before extracting the DCT coefficients.

Since the TIMIT sentences that make up the database have been specifically created for

Table 2.2 – Phrase recognition results (in %) on the multi-view dataset of OuluVS2 using our simple GMM-HMM model on the given test set per speaker and the corresponding mean and standard deviation across speakers **without the silence label** (SC = Sentence correctness, WC = Word correctness and WA = Word accuracy).

	0°			30°			45°			60°			90°		
Spkr.	SC	WC	WA	SC	WC	WA	SC	WC	WA	SC	WC	WA	SC	WC	WA
6	46.7	42.7	40.0	50.0	54.7	48.0	50.0	53.3	49.3	36.7	38.7	34.7	40.0	41.3	32.0
8	50.0	46.7	46.7	40.0	37.3	33.3	40.0	37.3	37.3	36.7	36.0	34.7	36.7	42.7	29.3
9	33.3	30.7	26.7	40.0	44.0	30.7	40.0	41.3	33.3	36.7	32.0	22.7	30.0	37.3	24.0
15	43.3	49.3	36.0	46.7	45.3	41.3	36.7	38.7	33.3	40.0	34.7	33.3	46.7	41.3	38.7
26	36.7	38.7	30.7	40.0	37.3	30.7	43.3	45.3	40.0	40.0	41.3	34.7	46.7	46.7	36.0
30	56.7	61.3	54.7	33.3	36.0	29.3	20.0	34.7	20.0	30.0	45.3	33.3	40.0	46.7	29.3
34	60.0	54.7	52.0	53.3	48.0	44.0	33.3	37.3	29.3	46.7	53.3	49.3	53.3	52.0	48.0
43	66.7	70.7	64.0	40.0	37.3	30.7	53.3	53.3	48.0	56.7	57.3	53.3	60.0	56.0	52.0
44	83.3	86.7	81.3	70.0	74.7	74.7	56.7	61.3	53.3	63.3	61.3	58.7	36.7	45.3	32.0
49	73.3	69.3	69.3	46.7	38.7	33.3	43.3	42.7	37.3	33.3	28.0	25.3	50.0	49.3	45.3
51	43.3	42.7	37.3	26.7	24.0	17.3	33.3	30.7	29.3	50.0	42.7	38.7	43.3	46.7	37.3
52	46.7	54.7	49.3	13.3	10.7	0.0	23.3	29.3	25.3	50.0	54.7	44.0	60.0	62.7	48.0
Mean	53.3	54.0	49.0	41.7	40.7	34.4	39.4	42.1	36.3	43.3	43.8	38.6	45.3	47.3	37.7
SD	15.1	15.8	16.4	14.0	15.6	17.8	11.1	9.7	10.1	10.0	10.7	10.9	9.4	7.0	8.9

phoneme recognition, in this part we will only compare the recognition rates based on visemes (the visual equivalent to phonemes, see Table 1.1 for the phoneme-to-viseme mapping used).

A GMM (with 20 mixtures) models the emissions of the 12 different viseme classes (as defined in Section 1.1) each of which is modelled in time by an HMM with 4 states. No dictionary or grammar is used.

The overall results in Figure 2.3 and the more detailed performance values in Table 2.3 emphasise the difficulty of classifying visemes separately as opposed to the recognition of a limited number of words for OuluVS2. Comparing the discrepancy between viseme correctness and accuracy we can see that insertions in particular are very common and reduce the accuracy by a large margin. We can see this high level of insertions clearly in

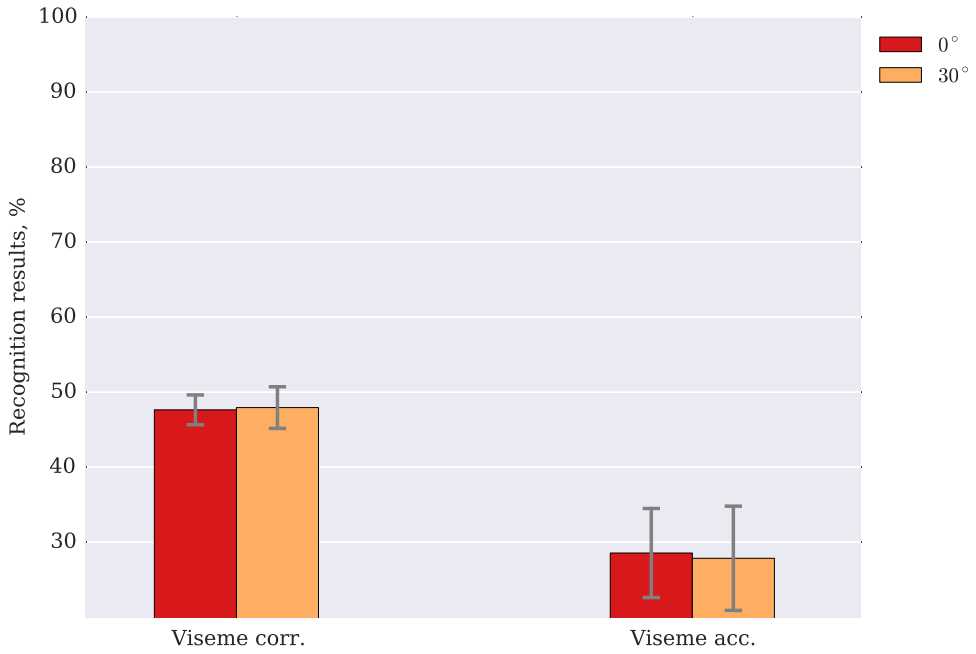


Figure 2.3 – Mean viseme recognition results on the multi-view dataset of TCD-TIMIT using a simple GMM-HMM model on the given test set. Error bars denote the standard deviation across subjects (corr. = correctness, acc. = accuracy).

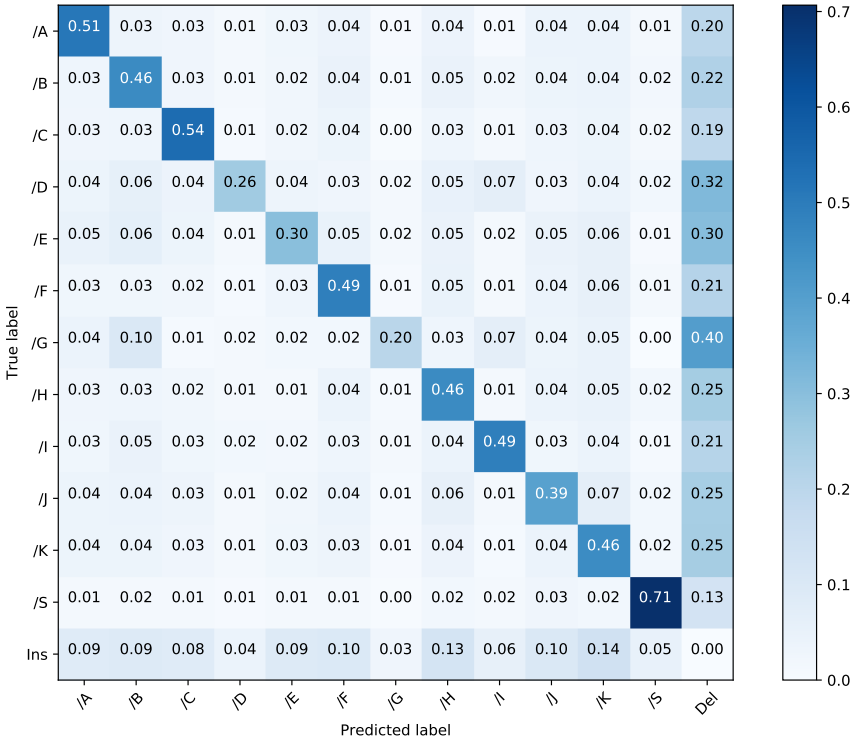
the confusion matrices for the frontal and 30° views in Figure 2.4. This is also confirmed when observing the per speaker results and the mean and standard deviation between the speakers, where the variations between speakers are much higher for the viseme accuracy due to insertions.

The confusion matrices in Figure 2.4 also indicate a high level of deletions. However, the confusions between labels are relatively low.

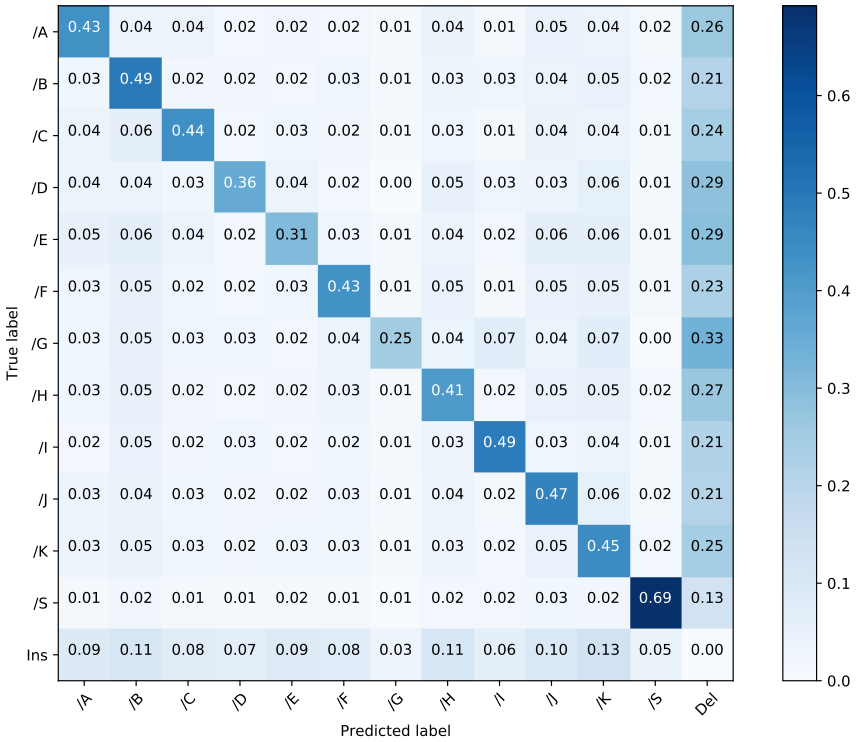
Comparing the two different views, we see that the performance is very similar and that both views seem to contain similar amounts of information for viseme recognition.

2.4 Summary

We can see that first, for the OuluVS2 database the word correctness and accuracy show good results, whereas the correct recognition of entire sentences still requires improvement. Furthermore, we can see that this simple GMM-HMM system seems to distinguish silence and words rather well, but requires further improvements in particular to distinguish between words. This simple system would also require further tuning and further experimentation to be comparable to the baseline results provided for the ACCV



(a) Frontal view.



(b) 30° view.

Figure 2.4 – Confusion matrices of our simple GMM-HMM model for the frontal and 30° views including insertions and deletions (Ins = Insertions and Del = Deletions).

2016 workshop challenge "Multi-view lip-reading/audio-visual challenges"[2] and by Anina et al. [2015].

Two main observations may be made from the baseline results for the TCD-TIMIT database. On the one hand, the viseme correctness of the frontal view results are similar to the baseline results by Harte and Gillen [2015]. On the other hand, the viseme accuracy shows a far lower performance. This indicates that there is a large number of insertions of visemes. These results could be improved by making use of a dictionary and a grammar, or by further fine tuning of the various parameters in the feature extraction and the GMM-HMM system.

[2]The baseline results can be found at http://ouluvs2.cse.oulu.fi/preliminary.html and in Table 3.1.

Table 2.3 – Viseme recognition results (in %) on the multi-view dataset of TCD-TIMIT using our simple GMM-HMM model on the given test set per speaker and the corresponding mean and standard deviation across speakers (VC = Viseme correctness and VA = Viseme accuracy).

	0°		30°	
Spkr.	VC	VA	VC	VA
08F	47.3	30.1	47.9	30.1
09F	48.0	30.6	51.0	29.3
15F	48.8	24.2	48.0	25.0
18M	43.3	31.2	47.3	27.5
25M	49.4	34.8	51.1	40.1
28M	44.3	30.0	44.1	28.1
33F	47.0	32.8	48.9	25.5
34M	49.4	6.9	46.2	5.0
36F	48.9	28.9	48.4	27.3
41M	45.0	31.0	42.8	32.6
44F	50.4	32.6	50.8	29.4
45F	45.9	27.2	44.1	21.5
49F	47.2	31.2	49.2	32.3
54M	47.5	29.0	47.7	27.7
55F	50.5	27.4	51.9	29.9
56M	48.4	26.7	44.2	28.0
58F	48.4	30.7	51.4	34.1
Mean	47.6	28.5	47.9	27.8
SD	2.0	6.1	2.9	7.2

3 Combined approach

The work presented in this chapter combines the advantages of deep learning methods, with the good time modelling techniques of GMM-HMM models. Firstly, mouth image patches are extracted from frames of the video speech data, on which PCA is applied in order to learn the weights of a two-stage CNN. Block histograms are then extracted as the unsupervisedly learned features. These features are employed to learn a recurrent neural network with a set of long short-term memory cells to obtain spatiotemporal features. Finally, the obtained features are used in a tandem GMM-HMM system for speech recognition. Our results show that the proposed method outperforms the baseline techniques applied to the OuluVS2 audiovisual database for phrase recognition.

3.1 Motivation

Efforts to bring visual speech recognition up to date with novel techniques used in both audio speech recognition and computer vision have led researchers to utilise deep learning techniques. DNNs are widely employed in audio-based automatic speech recognition resulting in the current state of the art results [Graves and Jaitly, 2014]. DNNs have also become the standard techniques in computer vision to set baselines in recognition or analysis tasks [Donahue et al., 2015, Chan et al., 2015]. However, one major problem in applying these networks to visual speech data is the fact that visual speech databases are not comparable to audio databases in terms of their sizes and number of speakers, meaning that insufficient amounts of training data are available. This is an important drawback since having a large amount of data is a necessity for training deep learning frameworks for complex acoustic models and complete recognition chains used for continuous speech. Although a few larger audio-visual databases such as TCD-TIMIT and OuluVS2 [Harte

Parts of this chapter have been published by Zimmermann et al. [2017a].

and Gillen, 2015, Anina et al., 2015] have been published recently, the problem still remains highly challenging.

The work presented in this chapter proposes a visual speech recognition approach based on a two-stage PCA-based convolutional network [Chan et al., 2015] followed by a layer of long short-term memories (LSTMs) to extract a set of unsupervised spatiotemporal visual features. These features are then used in a tandem GMM-HMM system for speech recognition. Our contribution is two-fold, with a major focus on feature extraction:

- First, we use principle component analysis in a multi-stage convolutional network to extract the optimal unsupervisedly learned lip representations. The two-stage projection onto the leading principle components allows us to capture the main variations within the image patches, while also extracting higher level features through the concatenation of these in two stages. The subsequent binarisation and extraction of histograms leads to an indexing and pooling of these features, a non-linear step.
- Secondly, we apply recurrent neural networks (RNNs) with LSTM cells to extract spatiotemporal features from lip representations. This approach not only finds the time-series dependencies within the video frame sequences, but also decreases the lips feature set dimension for further processing with the GMM-HMM scheme.

Using this system, we were able to improve the baseline cross-validation results[1] for phrase recognition from a frontal and 30° side view with a large margin of roughly 5%, reaching 79% of all sentences being recognised correctly for each of these views. Combining these two views leads to an even higher recognition rate of 83% of all sentences.

The rest of this chapter is organised as follows. Section 3.2 explains the details of the proposed method for visual speech recognition based on a PCA network, LSTMs, and the GMM-HMM system. Section 3.3 describes the utilised dataset, experiments, and obtained results. Finally, Section 3.4 concludes the chapter with a summary and discussions.

3.2 Proposed method

In this section, novel feature extraction methods are explored for visual speech recognition. More specifically, a two-stage PCA-based convolutional network [Chan et al., 2015] followed by a layer of LSTMs [Hochreiter and Schmidhuber, 1997] extracts features from the cropped mouth images. The obtained spatiotemporal features are then processed in a tandem system with a GMM-HMM basis for speech recognition.

[1]These results were published for the workshop "Multi-view lip-reading/audio-visual challenges" at ACCV 2016 at http://www.ee.oulu.fi/research/imag/OuluVS2/preliminary.html and are shown in Table 3.1.

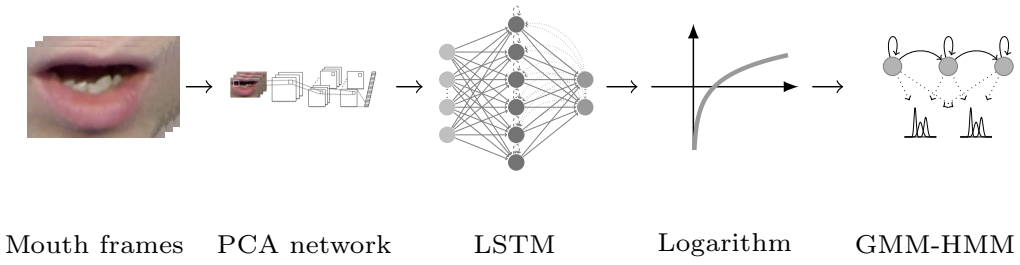


Figure 3.1 – The proposed tandem system with PCA network-LSTMs and GMM-HMMs for visual speech recognition from the mouth video frames.

Feature extraction is performed in a sequential fashion as shown in Figure 3.1. First, a two-stage PCA network is applied to each video frame (see Figure 3.2). The first layer network weights are learned by applying PCA to concatenated square patches – which are extracted from the mouth video frames and then vectorised. We use eight principle components as the networks' first-layer filter bank and convolve these with the input images. In a cascaded scheme, a similar procedure is applied to the filtered patches to obtain the second-layer filter bank. After convolution, the output maps are binarised with a Heaviside step function and every eight binary images are stacked together to compose an 8-bit image – similar to the first layer outputs. Finally, block histograms – with 256 bins – are extracted from the obtained maps and concatenated, resulting in a long feature vector for each video frame. In this work, we extract 16 block histograms which result in feature vectors of length $16 \times 256 \times 8 = 32,768$.

Secondly, an LSTM network is connected to the outputs of the PCA network to extract more abstract representations while taking the time-series dependencies between the video frames into account. This type of RNN is composed of memory cells to store the past values or ignore the dependencies when needed. Therefore, each cell has an input, an output, and a forget gate that can be activated at different levels. This architecture results in three cases: accepting the new input value, forgetting the existing value, or outputting a value at the given level [Graves et al., 2013], see Section 1.5.2 for more details. Since we label each video frame in the phrase recognition subset based on the audio phonemes, there are 28 output nodes in our LSTM network.

Lastly, the posterior probabilities received from the LSTMs are passed as spatiotemporal features concatenated with their delta and acceleration components into a GMM-HMM based speech recognition system, the so-called tandem approach. This system is implemented using the Hidden Markov Model Toolkit (HTK) [Young et al., 2009]. However, since the outputs of the LSTM network show small variations, we first take the logarithm of these features to make them more discriminative. Our tandem system contains GMMs

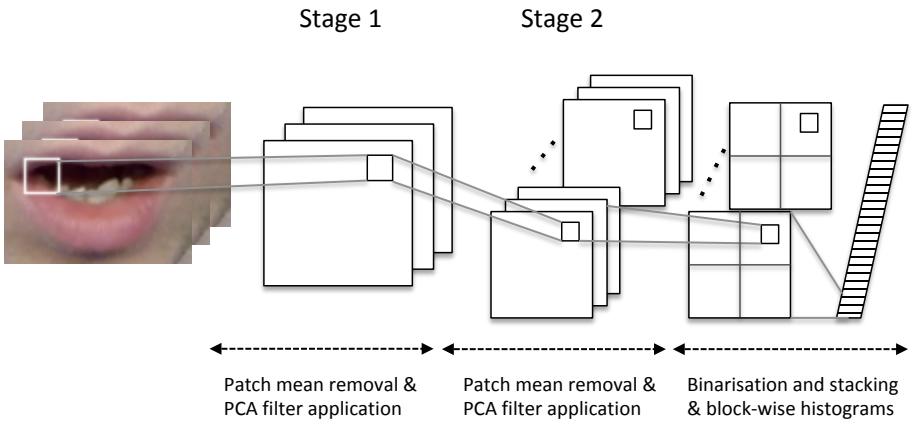


Figure 3.2 – The PCA network used in the first stage of the proposed tandem system with PCA network-LSTMs and GMM-HMMs.

with 15 Gaussian mixtures per observation and 4 states per word.

3.3 Performance analysis

In this section, we review details of the utilised dataset, evaluation metrics, and the conducted experiments. We present the validation and test results and discuss them in detail.

3.3.1 The dataset

We use the phrase recognition subset of the OuluVS2 database [Anina et al., 2015] in our experiments (see Section 1.8.2 for more detail).

3.3.2 Experimental results

In our experiments, we use the provided cropped mouth videos by first extracting and converting all video frames to greyscale images of size 60×90 pixels. PCA is applied to all image patches of size 7×7 pixels to learn eight filter banks in a two-stage cascaded PCA network. We add a max-pooling layer to the output of this network to obtain a more abstract representation before histogram pooling. Finally, 16 block histograms are

Table 3.1 – Baseline sentence recognition accuracy results (in %) on the multi-view dataset of OuluVS2 by the authors of the database using the DCT followed by HiLDA and an HMM (DCT+HiLDA+HMM) and raw pixel values in latent variable models (RAW+PLVM) with the cross-validation technique (results are approximative from http://www.ee.oulu.fi/research/imag/OuluVS2/preliminary.html).

	0°	30°	45°	60°	90°
DCT-HiLDA-HMM	74	71	73	73	68
RAW-PLVM	73	75	76	75	70

extracted and concatenated to obtain a 32,768-dimensional feature vector for each frame.

For spatiotemporal recognition using the LSTM network, we need to obtain frame-based labels using phoneme level transcription. For this purpose, the audio data is first aligned to the sentence transcriptions using a standard GMM-HMM system with mel-frequency cepstral coefficients (MFCCs) – a common feature in audio-based speech recognition[2] – trained on the training subset. These transcriptions are then used as labels for the obtained feature set from the PCA network. We train a one-layer LSTM network with a Sigmoid activation function in the gates and cells. The learning rate, weight decay penalty, and momentum value are set to 0.5, 0.001, and 0.8, respectively. We use a random batch size and train the network until 10,000 iterations.

To adjust our system parameters, we use a leave-one-out cross-validation scheme across speakers on the given training set. Subsequently, we apply the system in a leave-one out cross-validation scheme on the whole data[3]. This system is then trained using the whole training set, and finally applied to the test set to obtain the final recognition at the word or phrase levels. Figures 3.3 and 3.4 show our cross-validation and test results on the OuluVS2 dataset for phrase recognition.

Single-view experiments

We compare our results with the baseline results provided by the authors of the database and organisers of the workshop "Multi-view lip-reading/audio-visual challenges" at ACCV 2016. The results shown in Table 3.1 were obtained using two different methods: DCT followed by HiLDA and an HMM (DCT+HiLDA+HMM) and using raw pixel values in latent variable models (RAW+PLVM). They were obtained using leave-one-speaker-out cross-validation.

[2]The MFCCs are derived from the DCT of the log power spectrum on the nonlinear mel frequency scale.

[3]As was done in the baseline, whose results can be found at http://ouluvs2.cse.oulu.fi/preliminary.html and in Table 3.1

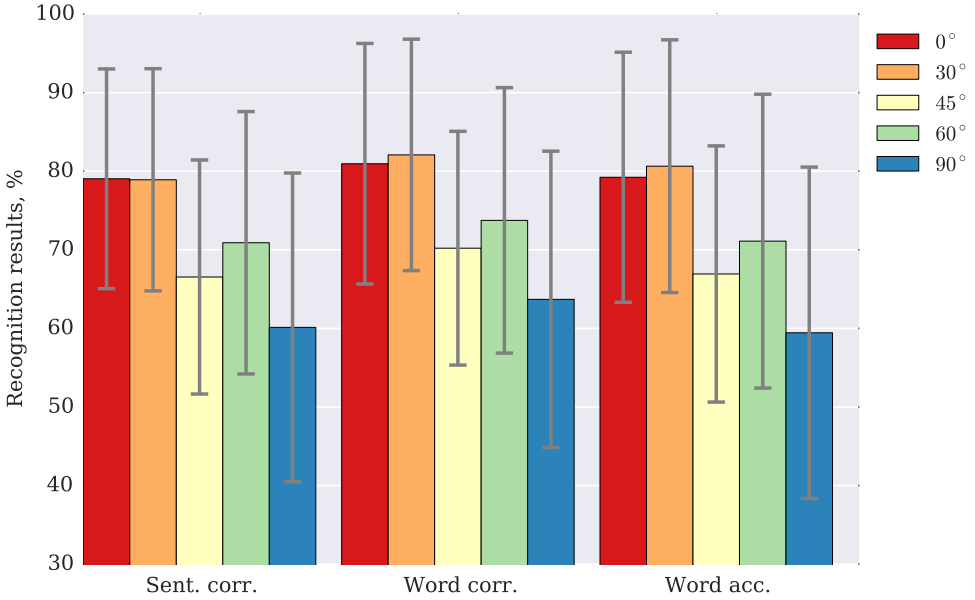


Figure 3.3 – Mean phrase recognition results on the multi-view dataset of OuluVS2 using the proposed tandem system with PCA network-LSTMs and GMM-HMMs with the cross-validation technique on the whole dataset. Error bars denote the standard deviation across subjects (sent. = sentence, corr. = correctness, acc. = accuracy).

As the obtained results show, approximately 81% of the words are correctly recognised on average during the cross-validation approach for the frontal view. In addition, we have achieved a word recognition correctness of around 74% on the test set. Also, we can see that 79% of sentences are correctly recognised during cross-validation while the performance on the test set is 73%. The small differences between the word correctness and accuracies indicate that there are very few insertion errors. Comparing our obtained results with the baseline cross-validation results on the same dataset reveals that we have improved the performance with a large margin of about 5%.

The average phrase recognition results for the 30° view show similar improvements over the baseline provided. Almost 79% of all sentences are classified correctly for the cross-validation data – approximately 4% more than the baseline – and around 76% on the test data. Similarly, for the test set 77% of all words are correct and the accuracy reaches 75%, while on the cross-validation these values reach 82% and 81%, respectively. The other views do not show improvements over the baseline.

Looking into the standard deviation indicated in the figures or the individual test results in Table 3.2, we can see, however, that there is a large margin between the performance of the best speaker and the worst. This hints at a common problem in visual speech recognition where the variability between speakers is very large.

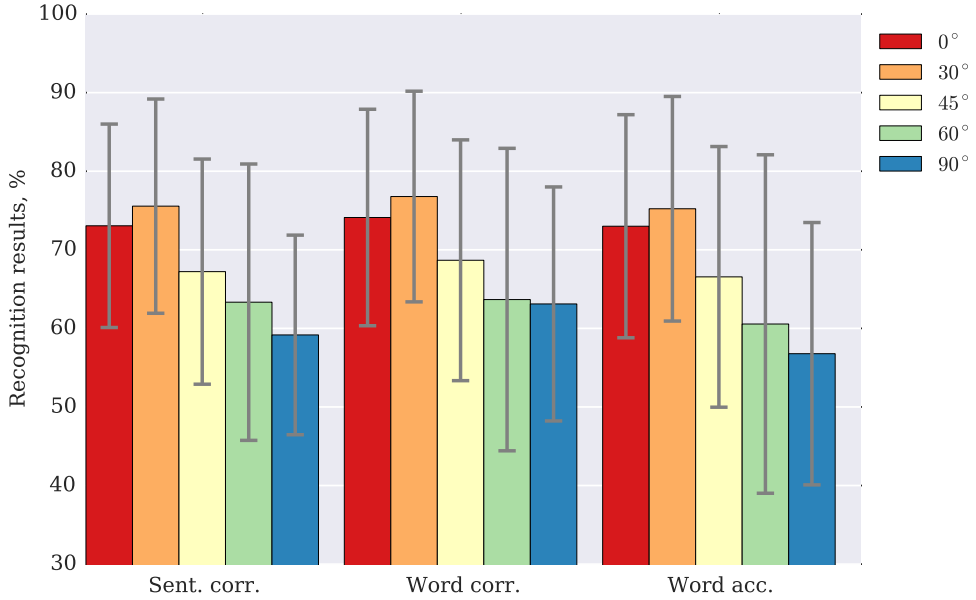


Figure 3.4 – Mean phrase recognition results on the multi-view dataset of OuluVS2 using the proposed tandem system with PCA network-LSTMs and GMM-HMMs on the given test set. Error bars denote the standard deviation across subjects (sent. = sentence, corr. = correctness, acc. = accuracy).

The frame recognition accuracy at the output of the LSTM is shown in Table 3.3. The per frame results on a phoneme and a viseme basis for the training and test sets are displayed here, which are obtained at the output of the LSTM and thus do not include any language modelling. The observations are two-fold: First, it can be seen that on a frame level the differences between the various view angles does not seem very big. However, the combination of successive frames proves more successful for the frontal views as described above. Secondly, the phoneme and viseme-based classification show a big difference between the 28 phoneme classes and 12 viseme classes (defined according to [Harte and Gillen, 2015]) due to the similarity between various phonemes represented only by the shape of the lips.

3.4 Summary

In this chapter, we have proposed a visual speech recognition system that utilises a two-stage cascaded PCA network to extract unsupervised learning based lip representations together with a layer of LSTM networks to obtain a set of spatiotemporal visual features. These features have later been used in a tandem GMM-HMM system for speech recognition. As the results indicate, the proposed method has outperformed the baseline technique with a large margin.

Table 3.2 – Phrase recognition results (in %) on the multi-view dataset of OuluVS2 using the proposed tandem system with PCA network-LSTMs and GMM-HMMs on the given **test set** per speaker and the corresponding mean and standard deviation across speakers (SC = Sentence correctness, WC = Word correctness and WA = Word accuracy).

	0°			30°			45°			60°			90°		
Spkr.	SC	WC	WA	SC	WC	WA	SC	WC	WA	SC	WC	WA	SC	WC	WA
6	73.3	73.3	73.3	70.0	70.0	73.3	53.3	60.0	56.0	40.0	40.0	37.3	50.0	57.3	52.0
8	56.7	54.7	54.7	66.7	66.7	62.7	66.7	74.7	73.3	66.7	66.7	66.7	63.3	66.7	60.0
9	43.3	45.3	41.3	53.3	53.3	56.0	56.7	54.7	50.7	43.3	45.3	40.0	36.7	36.0	24.0
15	73.3	77.3	76.0	63.3	63.3	58.7	53.3	54.7	53.3	40.0	40.0	34.7	40.0	41.3	30.7
26	76.7	74.7	74.7	90.0	90.0	89.3	63.3	62.7	58.7	70.0	65.3	65.3	80.0	85.3	82.7
30	73.3	80.0	78.7	76.7	76.7	77.3	90.0	92.0	92.0	73.3	82.7	80.0	73.3	84.0	74.7
34	96.7	97.3	97.3	86.7	86.7	90.7	80.0	85.3	85.3	80.0	80.0	78.7	63.3	61.3	56.0
43	73.3	78.7	76.0	83.3	83.3	81.3	63.3	62.7	56.0	56.7	50.7	41.3	70.0	70.7	68.0
44	80.0	81.3	81.3	86.7	86.7	88.0	80.0	78.7	78.7	93.3	97.3	97.3	50.0	52.0	48.0
49	86.7	88.0	86.7	80.0	80.0	80.0	93.3	93.3	93.3	83.3	86.7	86.7	63.3	69.3	68.0
51	66.7	60.0	60.0	53.3	53.3	50.7	50.0	42.7	41.3	43.3	41.3	33.3	53.3	56.0	48.0
52	76.7	78.7	76.0	96.7	96.7	94.7	56.7	62.7	60.0	70.0	68.0	65.3	66.7	77.3	69.3
Mean	73.1	74.1	73.0	75.6	76.8	75.2	67.2	68.7	66.6	63.3	63.7	60.6	59.2	63.1	56.8
SD	12.9	13.8	14.2	13.6	13.4	14.3	14.3	15.3	16.6	17.6	19.2	21.5	12.7	14.9	16.7

In this study only a limited dataset with a small vocabulary of 20 words has been explored to point out the benefits of using PCA networks in combination with LSTMs. Future works should thus extend this approach to other available datasets such as TCD-TIMIT [Harte and Gillen, 2015] that allow phoneme classification and provide a larger vocabulary. However, it has been found difficult to extend this approach in practice due to the large amount of computer memory needed to calculate the parameters of the PCA network, which increases with the size of the dataset. It is thus not a scalable approach, but shows good results for a limited amount of training data, resources and time.

In addition, the influence of the different views and their complementary or redundant nature within the framework of these spatiotemporal features could be explored in a more detailed multiple-view visual speech recognition study.

Table 3.3 – Frame recognition accuracy results (in %) on the multi-view dataset of OuluVS2 using the LSTM output of the proposed tandem system with PCA network-LSTMs on the given train and test sets across all speakers for phonemes and visemes (visemes defined according to [Harte and Gillen, 2015]).

		0°	30°	45°	60°	90°
Train	Phoneme	19.5	20.1	18.0	17.7	15.7
	Viseme	34.4	32.9	33.8	33.7	30.8
Test	Phoneme	17.2	17.7	17.1	17.2	16.1
	Viseme	30.8	30.4	32.0	31.4	29.6

4 Deep learning approach

In this chapter the recent advances in deep learning are taken into account to develop a sequence-to-sequence speech recognition system for the TCD-TIMIT database. We propose a system which consists of a CNN for feature extraction, an RNN for sequence modelling and a CTC for sequence decoding. The different building blocks of this system are chosen in a systematic fashion, by defining the blocks stepwise and building one block upon the other. The final results show that sequence-to-sequence deep learning models outperform the traditional method by a large margin.

4.1 Motivation

The aim of the work presented in this chapter is to take advantage of the recent developments in deep learning to design a sequence-to-sequence visual speech recognition system for complete sentences, rather than words. The focus here is on the TCD-TIMIT database, which provides a relatively large corpus of sentences spoken by more than 50 speakers. These sentences are designed for viseme recognition, thus allowing transcriptions of sentences with a large vocabulary of around 6000 words at a viseme-level. We propose to build a system from a CNN, an RNN and a CTC decoding scheme. Two systems are trained separately: one for the frontal view and one for the 30°-side view. The architecture of the two systems is identical, and chosen based on the frontal view.

This approach to VSR can easily be extended to continuous speech recognition, unlike some of the other approaches in the field. Furthermore, the aim is to create a system that can be run on a single computer, rather than a cluster or via a cloud, since speech recognition is often used in direct interaction with a speaker, and thus the model can be stored and applied locally.

The chapter is organised as follows. First, the proposed method is presented in Section 4.2. Then, Section 4.3 details the experiments performed and their results. Finally, Section 4.4 concludes the chapter.

4.2 Proposed method

The aim is to not only propose a new method for VSR, but to also systematically evaluate the parts that make up the sequence-to-sequence model, to develop the highest performing architecture. The pipeline is composed of several parts: a feature-extraction network, made up of CNNs, a time-evolutional model, from RNNs, and a decoding layer, using CTC. In the following, we will describe the different steps in this processing pipeline.

4.2.1 Feature extraction using convolutional neural networks

CNNs are effective networks for feature extraction and image classification. Their most important building block are the convolutional layers, which can be understood as convolving a matrix of weights with the input image or matrix. Through this convolution, certain patterns can be recognised independently of their location inside the input image. Each element in the output of the convolution is one neuron, which then passes the information on to the next layer. These neurons thus share their weights (the convolutional matrix), the width of the convolutional layer determining how many different convolutions will be performed.

In addition to the convolutional layers other layers commonly found in CNNs are max pooling and fully connected layers. Max pooling allows to reduce the size of the input field to the next layer by only retaining the maximum value within a certain grid, thus helping to concentrate the essential information and reduce the parameter size. Fully connected layers are closer to MLPs, since for these, as the name indicates, all the input and output neurons are connected. These are usually used as the last layers, to classify the features extracted from the previous layers.

This work uses a simple architecture involving a sequence of convolutional layers, some of which are followed by max pooling, and compares it to some existing and pre-trained networks (see Figure 4.1a).

The pre-trained networks used in this work are MobileNet [Howard et al., 2017] and VGG16 [Simonyan and Zisserman, 2015], with the weights trained on ImageNet[1] data – a large-scale image database used for a yearly object recognition challenge – provided by the popular deep learning library Keras[2]. The top layers of each of these networks are removed and replaced by a pair of fully connected layers with dropout – a method which randomly sets certain input values to 0 during training to avoid overfitting [Srivastava et al., 2014] – and a final softmax layer – which provides a probability distribution over the different output classes, here the visemes. These fully connected and softmax layers are trained specifically for the viseme recognition task. Before passing the output of the

[1] http://image-net.org/
[2] https://keras.io/applications/

Table 4.1 – Comparison of model sizes (No. of parameters) for the different CNN architectures and input image sizes. For the pre-trained networks, the number in brackets indicates the number of parameters that are trained (top layers).

Network	Small input image	Medium input image	Large input image
Own network small	138,734	280,046	998,894
Own network large	44,526	52,718	89,582
MobileNet small	864,910 (35,374)	864,910 (35,374)	864,910 (35,374)
MobileNet large	3,297,006 (68,142)	3,297,006 (68,142)	3,297,006 (68,142)
VGG16	14,750,062 (35,374)	14,750,062 (35,374)	14,750,062 (35,374)

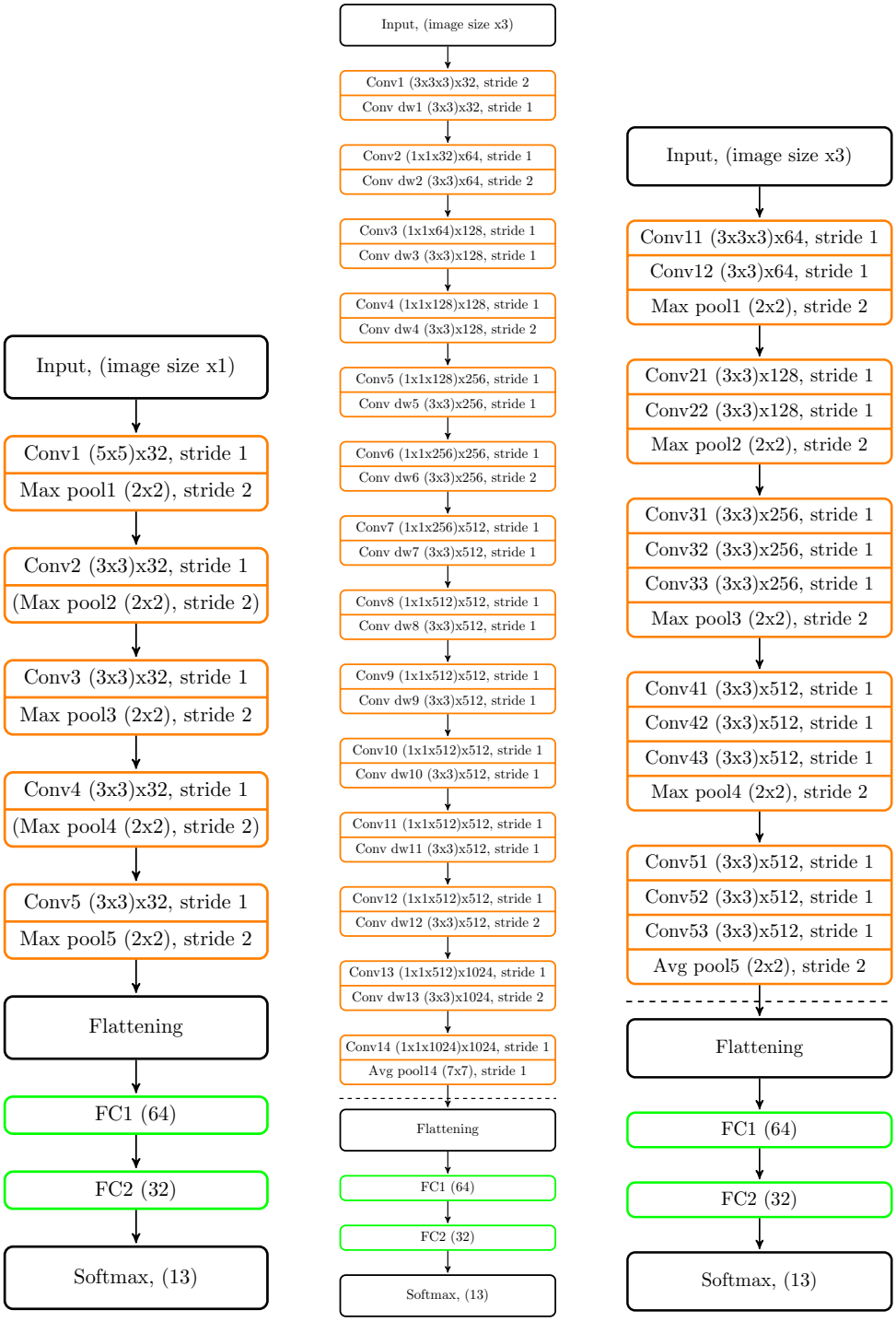
pre-trained networks to the top layers, an average pooling is performed to reduce the data size. No networks pre-trained for face recognition have been used, since usually the mouth is the part of the face that the classifier should be more invariant to, to recognise people irrespectively whether they are smiling, have an open or closed mouth.

MobileNet is a network specifically designed for mobile applications. It has thus been created with the aim to maximise performance while still being computationally light. It is made up of depthwise and pointwise convolutional filters (see Figure 4.1b), called depthwise separable convolutions in combination [Howard et al., 2017]. The former applies the same filter to all channels of an input. The latter then combines these layers by applying a 1×1 convolution. Therefore, the model size and necessary computations are greatly reduced compared to conventional models. While the first layer of the network contains a full convolution, the following layers are made up of depthwise separable convolutions. To reduce the size of this network further, it is possible to reduce the width of the network by a multiplicative factor of $0 < \alpha < 1$. The recognition results on ImageNet are lower than for other common networks, but remain reasonable taking into account the considerably smaller size of the network.

The VGG16 network is made up of 16 layers with convolutional and fully connected layers (see Figure 4.1c). The convolutions are typically grouped in twos or threes which are followed by a max pooling layer [Simonyan and Zisserman, 2015]. The network has shown good performance on the ImageNet dataset[3] and has been widely used as a pre-trained network to build upon.

A comparison of the model sizes is shown in Table 4.1. We can clearly see that our own simple networks are much smaller in size than the networks designed for the ImageNet task. This is very important also in terms of computational power, even if the networks are not being retrained. Keeping all the model parameters in the computer's memory

[3] http://image-net.org/



(a) Own networks small (and large with Max pool2 & 4).

(b) MobileNet.

(c) VGG16.

Figure 4.1 – Flowcharts of the different CNN architectures (conv = convolutional layer, max pool = max pooling layer, conv dw = depthwise separable convolutional layer, avg pool = average pooling layer, FC = fully connected layer). For (b) and (c) the dashed lines indicate the pre-trained layers above and the added layers below.

Table 4.2 – Comparison of model sizes (No. of parameters) for the different RNN architectures in combination with our own large CNN network.

RNN length	GRU	LSTM	BGRU	BLSTM
30	67,910	76,430	100,790	119,630
50	95,320	112,430	165,030	204,430
70	129,750	158,030	248,470	314,830
100	195,030	244,430	409,630	528,430
150	339,830	436,430	774,230	1,012,430
200	529,630	688,430	1,258,830	1,656,430
300	1,044,230	1,372,430	2,588,030	3,424,430

takes up considerable space, in particular for the VGG16 model. We can also observe that the own "small" network actually requires more parameters than the own "large" network, since its output of the convolutional layers to the dense layers is much larger than in the own "large" network, where the additional max pooling layers reduce the feature map size.

4.2.2 Sequence modelling with recurrent neural networks

In this work we compare results between RNNs with LSTM cells and GRUs (see Section 1.5.2 for more details). The aim is to see which kind of network performs better, since we only have a limited amount of data available. The results are also compared to bidirectional RNNs (BRNNs) which process the input sequence in two layers, one receiving the sequence in forward order and one in reverse order. This is common practice in recent speech recognition work, since it is assumed that the network will always be fed sequences of at least a specific length, and not one sample at a time. The integration of the RNN into the previous CNN network is shown in Figure 4.2.

Table 4.2 presents the number of parameters for these different RNN architectures (combined with our own large CNN network). We can clearly see the difference between the GRU and LSTM-based models, where the number of parameters is significantly larger. Since the BRNNs have to model both the forward and backward sequences, as well as the connections between these two, the number of parameters is even larger for the bidirectional GRU (BGRU) and BLSTM.

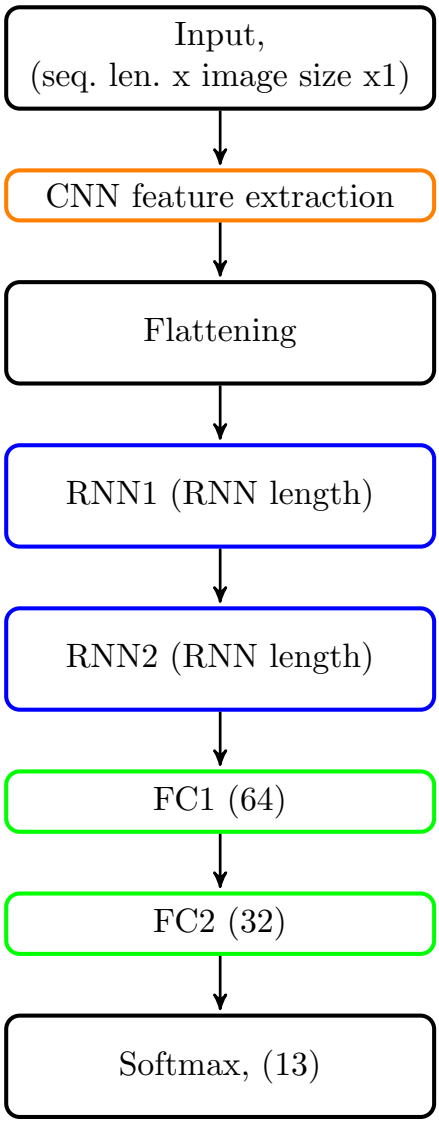


Figure 4.2 – Flowchart of the RNN network integrated with the CNN (seq. len. = training input sequence length).

4.2.3 Decoding with connectionist temporal classification

Finally, the output of the BRNNs is decoded using the CTC. This sequence decoder, described in more detail in Section 1.5.2, calculates the overall combined probabilities for various sequences which allows to choose the most likely one based on the emission probabilities from the network at each time step. In this work we use the greedy method of best path decoding (see equation 1.30 in Section 1.5.2), not performing prefix search decoding with its extensive beam search and grouping of alignments. Neither a dictionary nor a language model are used to restrain the search tree.

4.3 Performance analysis

Since finding the right parameters and the best deep learning architecture is a very complicated procedure, a systematic approach was applied to this part of the work. This involved a multi-step approach: first, the best combination of input image size and image recognition system was found. Next, the best sequence modelling architecture was added. Finally, this was combined with the decoding of the sequence.

For the image recognition/feature extraction part, five different networks were tested:

1. A simple system of five convolutional layers with three max pooling layers after every second convolutional layer, "Own network small" (see Figure 4.1a, excluding the max pooling layers in brackets).
2. Similar to system 1) but with max pooling after each convolutional layer, "Own network large" (see Figure 4.1a).
3. The reduced pre-trained MobileNet [Howard et al., 2017] with a width of $\alpha = 0.5$, "MobileNet small" (see Figure 4.1b, with the widths of the convolutions reduced by factor α; only the last layers after the dashed line are trained).
4. The full pre-trained MobileNet [Howard et al., 2017] with a width of $\alpha = 1$, "MobileNet large" (see Figure 4.1b; only the last layers after the dashed line are trained).
5. The full VGG16 [Simonyan and Zisserman, 2015] network, "VGG16" (see Figure 4.1c; only the last layers after the dashed line are trained).

All the networks had the same architecture for the last few dense layers. For the pre-trained networks only these last few layers were adapted during training, whereas for the other networks the whole network was trained from scratch.

In addition, three different input sizes of images were tested: 150×210 pixels ('large'), 75×105 pixels ('medium') and 50×70 pixels ('small').

From these experiments the best combination of network and input image size was chosen, in order to be combined with the temporal model. Here again, a number of parameters were varied to determine the best possible architecture.

First, the sequence length modelled by the RNN was varied from 30 to 300, and the input sequence length was varied between 30 and 50 samples. A longer input image sequence could not be tested due to memory requirements. Finally, a comparison between GRU, LSTM, BGRU and BLSTM was performed.

Based on the results from these experiments the best four systems were chosen and then integrated into a full end-to-end speech recognition system with CTC as a sequence decoder.

All experiments were implemented using the Keras[4] library in Python 3 with TensorFlow[5] as backend. The Adam optimiser [Kingma and Ba, 2015] was used with the categorical cross entropy functioning as loss function and using the categorical accuracy as metric – except for the experiments with the CTC, where the CTC loss function was used, and no metric was used during training. The default learning rate of 0.001 was used. For each epoch, the training was performed on the whole training set – with batches of 50 for the first CNNs and 32 to reduce the memory use for the RNN and CTC experiments – and then tested on the validation set. Where not mentioned otherwise, the final model was trained for 10 epochs with reshuffled training data. All computations were performed on a single GPU – an NVIDIA GTX 1080 Ti with 11 GB memory.

4.3.1 The dataset

In these experiments the TCD-TIMIT database was used to train the models, since it is considerably larger than the OuluVS2 short phrases (see Section 1.8.1). Two of the sentences were excluded for each subject from both the training and test sets: the ones highlighting the speaker's accent. This is due to two reasons: on the one hand, these sentences are repeated for each subject and are thus more predictable. On the other hand, the fact that they highlight the accent could impact the understandability of these sentences. Thus, 96 sentences were used per speaker.

From the training set, another six subjects were randomly chosen to be in the validation set, to control the evolution of the training across epochs.

Again, we wanted to obtain results for both the frontal and the 30°-side view. Therefore, we use our own SDM-based face tracker to extract the region around the mouth. To this end, first the average mouth size for each image is calculated. In a second round the particular ROI is cropped out. In this experiment three different types of ROIs are

[4]https://keras.io/
[5]https://www.tensorflow.org/

tested:

1. A region cropped with a larger margin of 2.5/3 added to each side horizontally around the mouth in greyscale. All the images were normalised by dividing by 255 (the maximum possible value) and removing their own mean values. Final input image size ('small'): 50×70 pixels.
2. A region cropped with a larger margin of 2.5/3 added to each side horizontally around the mouth in greyscale. All the images were normalised by dividing by 255 (the maximum possible value) and removing their own mean values. Final input image size ('medium'): 75×105 pixels.
3. A region cropped closely around the mouth (margin of 1/3 added to each side horizontally) with the greyscale values adjusted by histogram equalisation. All the images were normalised by removing the mean and dividing by the standard deviation of the training set. Final input image size ('large'): 150×210 pixels.

The transcriptions aligned in time according to the (time-synchronous) audio as provided by the authors of the database were used as ground truth in the training procedure. The phoneme labels were converted into viseme labels according to the mapping provided in Table 1.1 in Section 1.1, except for /hh/ and /hv/, which take the shape of the following phoneme [Harte and Gillen, 2015]. These phonemes are thus converted according to the next phoneme label.

Since the silence label was highly overrepresented in the distribution of visemes, the leading and trailing silence of each utterance were cropped. As a result, the data was distributed as shown in the histogram in Figure 4.3. We can see that certain visemes are much more common than others – a common phenomenon in speech recognition.

4.3.2 Experimental results

In the following, we present the steps and results of the building blocks to the full sequence-to-sequence visual speech recognition system. We start with the feature extraction using CNNs, then add sequence modelling with RNNs and finally combine it with decoding using CTC. In the end, this architecture, designed on the frontal view, is also trained on and applied to the 30°-side view.

Feature extraction using CNNs

The framewise viseme classification results for the analysis of the different CNN architectures after training for 10 epochs with a batch size of 50 can be found in Tables 4.3 (validation set) and 4.4 (test set). Here we can see that the small networks which are

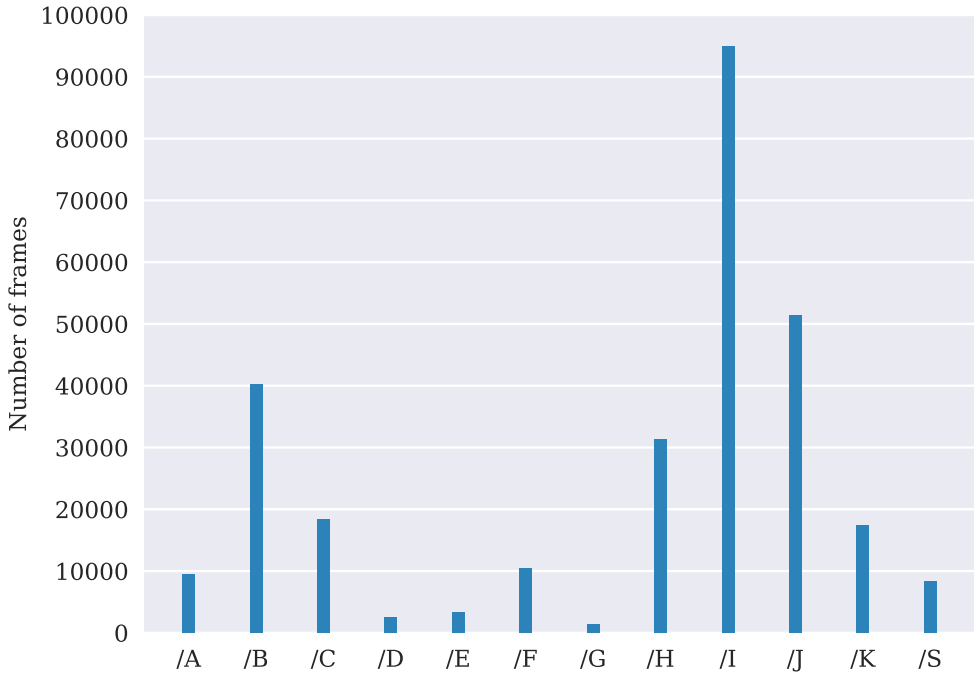
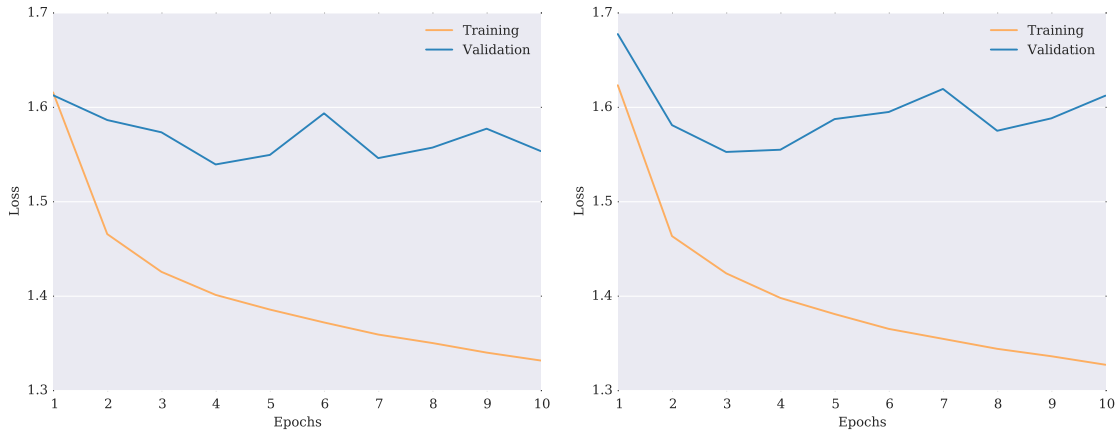


Figure 4.3 – Histogram of the visemes in the TCD-TIMIT database after cropping leading and trailing silence.

trained from scratch perform much better than the pre-trained networks. This might partially be due to the fact that since only the last, classification layers are trained, the same features as for the image classification task ImageNet are extracted. These features are specialised to recognise different animals and objects, despite different appearances. Therefore, the variations in the mouth shape might be partially ignored.

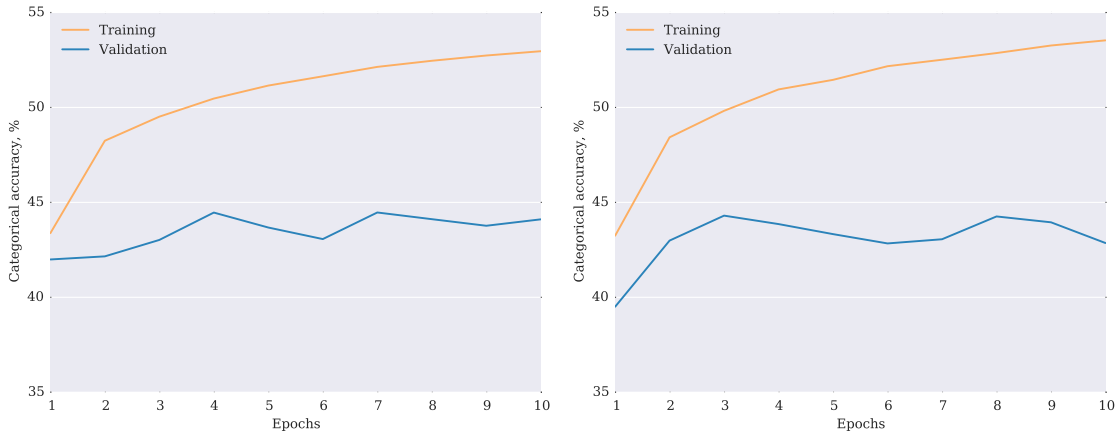
This also indicates that for VSR it might be better to use our own relatively simple network rather than relying on more complicated and larger models which would need more space in computer memory and thus only allow shorter time-sequences and batch sizes for training, which ultimately leads to longer training times.

We can also see that even though the performance for the small and medium image sizes is very similar, the medium size consistently performs better except for VGG16 and MobileNet small, but the results are very close for the latter. For the larger image size only one test was performed with our own small network. However, the training time is largely increased to around 10 hours per epoch, as compared to around 20 minutes per epoch for the medium input image size. Therefore, the training was stopped after 5 epochs, and is compared to the one with medium image size after training for 5 epochs. We can see that the results are very similar. Therefore, there is no added value to using the larger input images.



(a) Own network small, input image size medium. (b) Own network large, input image size medium.

Figure 4.4 – Evolution of the loss for the training and validation sets during training for 10 epochs.



(a) Own network small, input image size medium. (b) Own network large, input image size medium.

Figure 4.5 – Evolution of the categorical accuracy for the training and validation sets during training for 10 epochs.

Table 4.3 – Framewise viseme accuracy results (in %) on the frontal view of TCD-TIMIT using the different CNN architectures on the validation set (* = results after training for 5 epochs).

Network	Small input image	Medium input image	Large input image
Own network small	42.23	**44.11**/43.67*	38.60*
Own network large	41.70	42.86	-
MobileNet small	34.74	34.27	-
MobileNet large	35.40	35.53	-
VGG16	39.79	34.65	-

Table 4.4 – Framewise viseme accuracy results (in %) on the frontal view of TCD-TIMIT using the different CNN architectures on the given test set (* = results after training for 5 epochs).

Network	Small input image	Medium input image	Large input image
Own network small	39.83	39.95/39.47*	39.23*
Own network large	39.56	**40.18**	-
MobileNet small	33.07	34.49	-
MobileNet large	33.70	34.78	-
VGG16	37.70	37.59	-

The evolution of the loss and categorical, framewise accuracy for both training and validation sets are shown in Figures 4.4 and 4.5 across the 10 training epochs for the two best performing networks (on the validation set): our own small and large networks, with a medium input image size. These graphs show that while the training performance increases constantly, even though slower towards the end, the validation performance varies much more. However, still an overall trend towards an improving performance is visible, which is a little stronger in the case of the larger network.

Taking into account these different considerations, we chose to use the medium input images with our own large network for the evaluations with RNNs, also because it reduces the final matrix size passed to the RNN, thus allowing for fewer parameters in the RNN.

Finally, to evaluate how the systems perform at a viseme level, we show the confusion matrices of these two best CNNs in Figure 4.6. These confusion matrices illustrate that certain visemes are always misclassified, whereas others are classified correctly most of the times. Visemes are in particular misclassified to /B, /I and /J ('Lips puckered', 'Lips

relaxed narrow opening' and 'Tongue up or down', see Table 1.1). These are the most common visemes in this database (see Figure 4.3). In addition, their shape is similar to some other visemes. This could also point towards a slight overfitting of the network.

Sequence modelling using RNNs

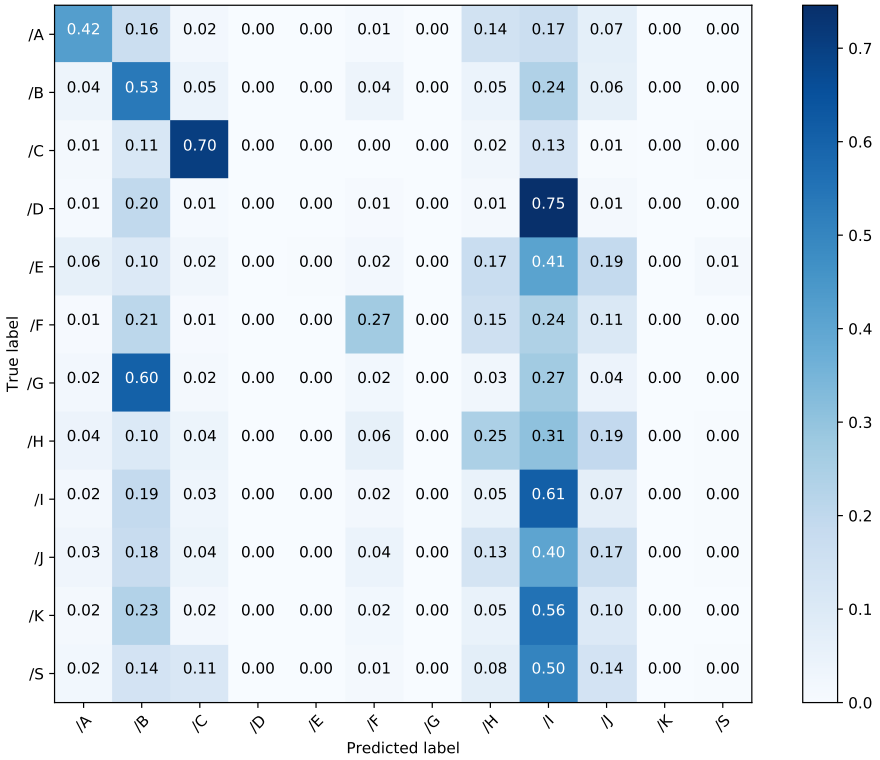
Tables 4.5 (validation set) and 4.6 (test set) present the results of combining our own CNN architecture with different RNNs to model and consider the temporal evolution in the viseme recognition process. The results are framewise classifications and we can see that using longer RNNs proves crucial to improving temporal modelling.

Furthermore, we can see a difference between the input sequence lengths: while it might be counter-intuitive, using training input sequence lengths ($l_{inputseq}$) of 30 leads to better models than using length 50 training input sequences. This could be related to the way these training sequences are created: each utterance of length L can contain $L - l_{inputseq}$ training samples. These sequences are shuffled across the whole training set, to avoid repeating very similar sequences in the same batch and thus overfitting. Therefore, shorter training input sequences result in more training samples: 4721 for $l_{inputseq} = 30$ and 2742 for $l_{inputseq} = 50$, and thus can help to improve the results.

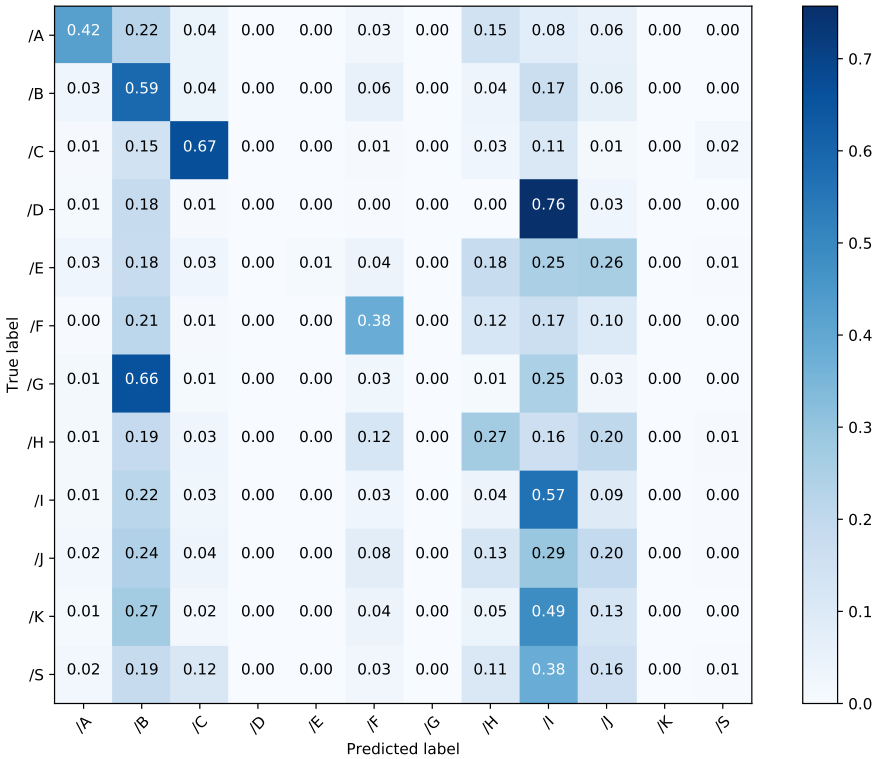
Another interesting observation is the difference between LSTMs and GRUs: the results are rather close, however, most of the time the GRUs perform better. This is most likely due to its lower number of parameters, which is particularly important when dealing with small amounts of data. Furthermore, like expected, the bidirectional versions of the RNNs outperform the unidirectional variants in most cases, and are particularly useful for longer networks. This shows that taking into account some future and past information is very useful to provide a more complete context to each frame. With the shorter networks, the additional information that could be gained is not as significant, and thus the performance increase is smaller.

In the end we can conclude that the best system is the BGRU with length 300 fed with an input sequence of length 30. The framewise confusion matrix on the test set for this setup is shown in Figure 4.7. We can clearly see the improvement over the previous confusion matrices for the CNN-only classification in Figure 4.6. However, similarly to the CNN-only classification, we still have the same classes that function as sink models.

To test the influence of the decoding CTC layer, we will perform tests with a series of best networks: we maintain the CNN architecture with our own large network and the medium image as input. Both, input sequences of 30 and 50 are tested, and the BGRU length is varied between 200 and 300. These architectures are now also tested on the 30° view.



(a) Own network small, input image size medium.



(b) Own network large, input image size medium.

Figure 4.6 – Normalised confusion matrices on the test set for the best two CNNs.

Table 4.5 – Framewise viseme accuracy results (in %) on the frontal view of TCD-TIMIT using the different RNN architectures on the validation set.

RNN type	Input sequence length	RNN length						
		30	50	70	100	150	200	300
GRU	30	41.99	44.89	45.42	44.72	44.91	44.23	44.48
	50	43.50	43.49	45.30	46.58	44.38	43.67	44.56
LSTM	30	40.37	45.97	46.27	44.47	43.22	42.72	42.47
	50	41.46	44.36	45.32	44.00	42.71	42.59	43.09
BGRU	30	45.07	47.09	48.55	48.73	46.96	46.69	**51.81**
	50	45.57	46.98	47.03	48.91	44.73	50.20	50.91
BLSTM	30	44.30	47.29	46.92	46.61	46.15	45.25	45.36
	50	44.28	46.84	46.05	47.05	45.64	46.52	47.04

Table 4.6 – Framewise viseme accuracy results (in %) on the frontal view of TCD-TIMIT using the different RNN architectures on the given test set.

RNN type	Input sequence length	RNN length						
		30	50	70	100	150	200	300
GRU	30	41.75	42.16	42.58	43.25	42.09	42.66	43.07
	50	40.92	41.72	42.40	43.15	42.85	41.39	42.83
LSTM	30	39.84	40.77	41.67	41.74	40.55	42.26	40.73
	50	40.19	41.31	42.30	41.63	41.96	40.95	41.80
BGRU	30	41.93	45.02	44.94	45.15	46.41	45.29	**46.92**
	50	43.40	44.09	44.49	44.87	44.82	46.74	46.09
BLSTM	30	42.50	43.44	43.32	44.41	45.17	43.82	45.23
	50	43.13	43.07	43.59	44.30	44.12	44.55	46.00

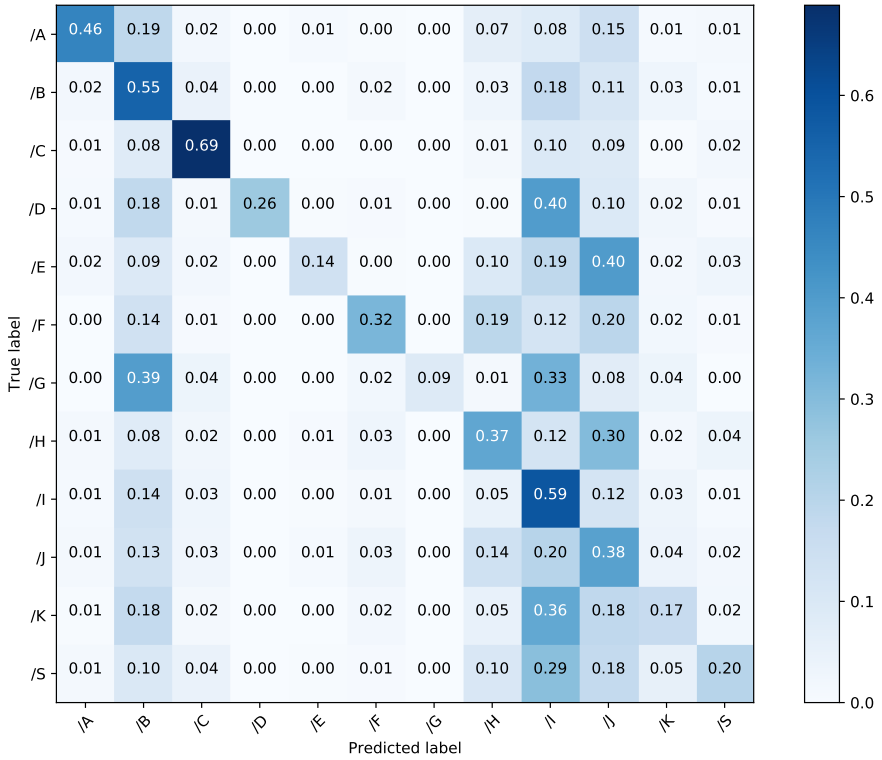


Figure 4.7 – Normalised confusion matrix on the test set for the best RNN: a BGRU with length 300 and trained with an input sequence length of 30.

Table 4.7 – Viseme recognition accuracy (in %) on the frontal and 30° views of TCD-TIMIT using different RNN architectures with CTC on the given test set.

View	Input sequence length	BGRU length	
		200	300
0°	30	53.98	**55.17**
	50	53.78	52.85
30°	30	53.48	54.75
	50	54.54	**55.49**

Decoding with CTC

In this last step we analyse the influence of a CTC decoder on the results. We maintain the previously determined architecture and pass the output of this network to the CTC which will group the framewise output of the BGRU and output sequences of visemes, rather than classifying each frame. In our work we do not constrain the CTC with a dictionary or grammar, since we want to see the model's capability to learn independently of these.

In Table 4.7 the results for the complete network with varying RNN length and input sequence length are presented, now for the two views: frontal and 30°. The parameters to be compared were chosen according to the best performing networks in the previous step. We thus compare RNN lengths of 200 and 300 and vary the input sequence length between 30 and 50 frames. The accuracy is now the viseme accuracy based on the edit distance of a sequence, as defined in Section 1.6, compared to the framewise categorical accuracy in the previous sections. The accuracy is computed based on the edit distance in the 'editdistance' library in Python.

We can clearly see the improvement from using only a CNN-RNN system for framewise classification (see Table 4.6), to the sequence classification results (see Table 4.7). Again, we observe that the longer sequence models perform better. The influence of the input sequence length cannot be defined completely, since it varies between the frontal and 30°-view angle.

Since overall the RNN length of 300 and input sequence length of 30 show the best performance, the results are analysed in more detail with HTK (see Table 4.8). HTK computes both the viseme correctness and accuracy, and allows to analyse the results per subject. The algorithm seems to be slightly different from the algorithm used for the results in Table 4.7, but the results are comparable.

In Figure 4.8, and in more detail in Table 4.8, we can see that the viseme recognition accuracies are much higher than the previous baseline results presented in Chapter 2. Also comparing to previous results presented in Thangthai et al. [2017] and the benchmark results of the database Harte and Gillen [2015] on the same test set, we can see that our method outperforms these by a large margin. For the speaker independent case on the frontal view, Thangthai et al. [2017] report a viseme accuracy of 44.60% (compared to 34.77% for the benchmark), which our method improves to 52.28% (using the HTK results).

The other works only used the frontal view. However, in our comparison we can see that both views perform similarly, with a slightly higher accuracy for the 30° view. This is consistent with our results presented in Chapter 3 for the OuluVS2 database.

Furthermore, looking at the detailed results in Table 4.8, it can be noted that in particular

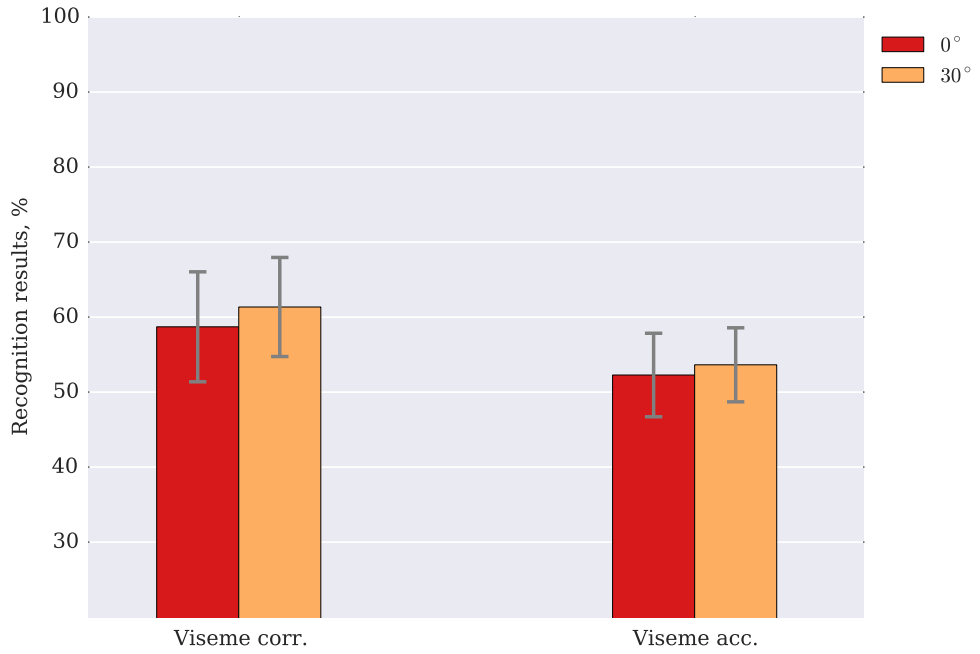
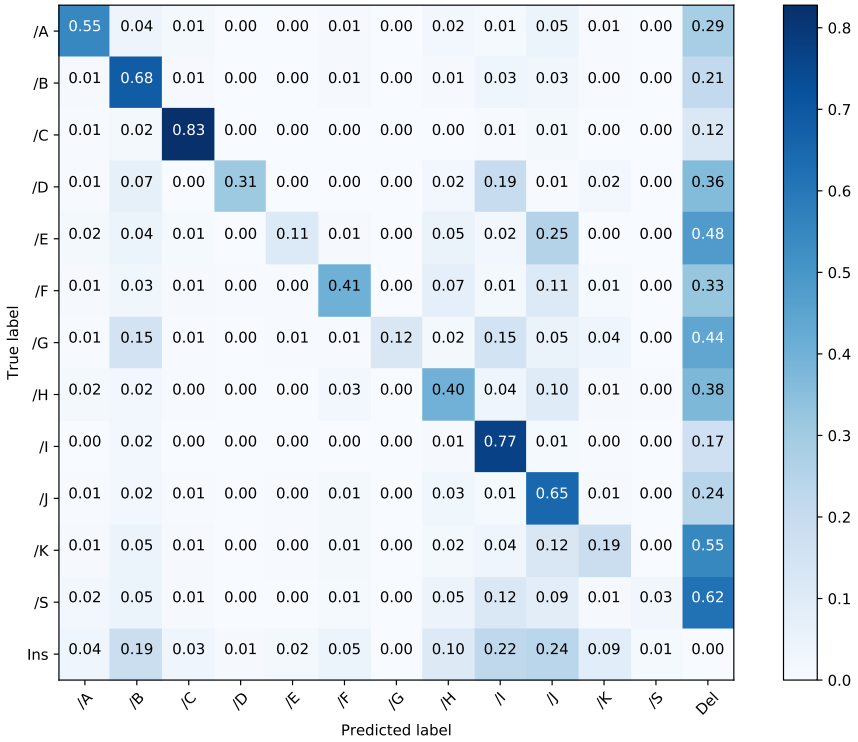


Figure 4.8 – Mean viseme recognition results on the multi-view dataset of TCD-TIMIT using our proposed CNN-RNN method with CTC on the given test set. Error bars denote the standard deviation across subjects (corr. = correctness, acc. = accuracy).

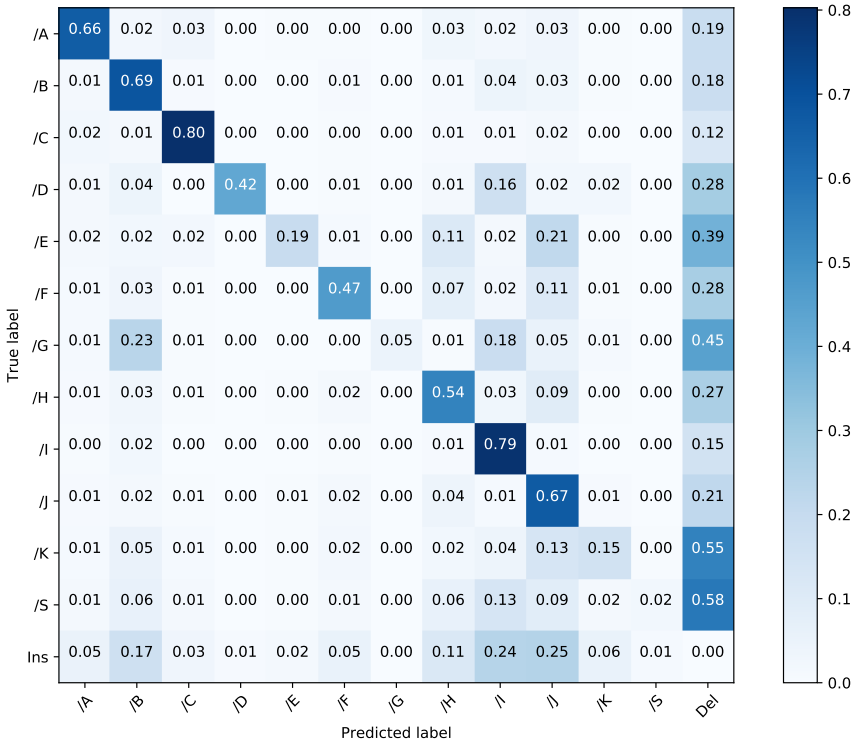
subject 34M performs poorer than the other subjects. This has also been the case in the previous GMM-HMM-based experiments (see Table 2.3). Looking into the videos, we can see that this subject has a beard, which might reduce the performance of the algorithm.

The viseme accuracy is still lower than the viseme correctness, like in the GMM-HMM case. However, the gap is now smaller, which indicates that there are still substitution errors, although not as many as in the GMM-HMM-based system.

Finally, the confusion matrices in Figure 4.9 (this time based on sequence classification, not framewise labels) clearly show that integrating the decoding with CTC improves recognition as compared to sequence modelling through RNNs (see Figure 4.7), which itself enhanced the CNN-based framewise image classification (see Figure 4.6). Comparing these confusion matrices to the ones obtained in the baseline system with GMM-HMMs (see Figure 2.4) we can see that for many visemes the classification has become more reliable. However, unlike for the GMM-HMM systems, we see that certain viseme classes have become sink models. Further we can note that for these particular classes there are higher levels of insertions, while for the other classes insertion levels are lower than for the GMM-HMM systems. Also the number of deletions remains relatively high.



(a) Frontal view.



(b) 30° view.

Figure 4.9 – Confusion matrices of our CNN-RNN with CTC for the frontal and 30° views including insertions and deletions (Ins = Insertions and Del = Deletions).

4.4 Summary

In this chapter we developed an end-to-end, sequence-to-sequence neural network from scratch. This network was built step by step, by first determining the best CNN-based feature extraction framework which was evaluated framewise with the classification accuracy. Here we could show that for such specific use cases, it can be more beneficial to train a small network from scratch, rather than reusing larger pre-trained networks.

Adding RNNs to the per-frame classification proved very useful to improve the results. However, it could also be observed that the results could likely be improved further with access to larger databases. Furthermore, we could see that using BGRUs led to the best performance, due not only to its advantage of bidirectional sequence modelling, but also its reduction in parameters compared to the LSTM-based networks. In addition, using a longer model has shown increased accuracy, since it allows taking more context into account.

Finally, we could see that using the CTC for decoding further increased the accuracy, since it smooths out the results while finding the best path of the viseme sequence. It thus removes outliers or unlikely combinations of frames. It also allows us to do actual sequence-to-sequence speech recognition, since we are ultimately interested in the output sequence rather than a classification for each frame.

We were also able to show that this sequence-to-sequence deep network outperforms previous state-of-the-art results on TCD-TIMIT by a large margin.

The networks were optimised on the frontal view, however, training the same architecture for the 30°-view angle proved that it was just as suitable for this view. The performance on the 30° view network was even slightly higher than for the frontal view, which is consistent with previous results.

Table 4.8 – Viseme recognition results (in %) on the multi-view dataset of TCD-TIMIT using our proposed CNN-RNN method with CTC on the given test set per speaker and the corresponding mean and standard deviation across speakers (VC = Viseme correctness and VA = Viseme accuracy).

	0°		30°	
Spkr.	VC	VA	VC	VA
08F	65.67	56.17	67.77	57.58
09F	65.14	56.33	65.66	59.36
15F	58.38	52.07	63.91	53.72
18M	63.06	57.16	59.93	54.09
25M	60.05	53.77	57.35	50.87
28M	59.97	54.26	58.71	52.37
33F	64.29	55.02	64.64	55.46
34M	34.73	34.42	41.54	37.63
36F	61.44	54.78	61.50	54.12
41M	58.68	51.12	61.72	54.76
44F	65.44	57.57	63.86	56.50
45F	52.80	48.11	54.98	51.07
49F	62.28	55.36	67.65	57.00
54M	54.79	46.48	60.71	52.22
55F	58.67	53.61	72.31	60.79
56M	50.46	46.83	55.98	49.74
58F	62.11	55.64	64.66	54.56
Mean	58.70	52.28	61.35	53.64
SD	7.56	5.74	6.81	5.09

5 Multi-view visual speech recognition

A remaining challenge in the field of visual speech recognition arises from the problem of varying head poses. In this chapter, we present an in-depth study of the possibilities to combine various view angles and the influence of these combinations on the recognition results. To this end, we compare the concatenation of features as well as the decision fusion of simultaneous recordings from different camera angles. The features are obtained through a PCA-based convolutional neural network, followed by an LSTM network. Finally, these features are processed in a tandem system, being fed into a GMM-HMM scheme. The decision fusion acts after this point by combining the Viterbi path log-likelihoods. The results show that the complementary and redundant information contained in recordings from different view angles improves the results.

5.1 Motivation

Pure audio-based speech recognition has seen significant improvements over the last decades and has been tested on and applied to many real-life datasets and scenarios. However, visual speech recognition still focuses mainly on in-lab databases. To overcome some of the shortcomings that need to be addressed to work on real-world data, several studies have focused on VSR for various head poses [Lucey et al., 2007, Estellers and Thiran, 2012]. Moreover, a few databases have included recorded sentences from cameras at various view angles [Lee et al., 2004, Harte and Gillen, 2015]. Following up with this perspective, the work in this chapter continues the use of the recent OuluVS2 dataset [Anina et al., 2015] which includes simultaneous recordings from cameras placed at five different angles.

In this chapter, we continue the use of a PCA-based convolutional network [Chan et al., 2015] in combination with long short-term memory (LSTM) cells and a GMM-HMM scheme to model temporal evolution between words (see Chapter 3). This work focuses

Parts of this chapter have been published by Zimmermann et al. [2017b].

on the combination possibilities between various view angles and the complementary or redundant information that can thus be exploited. We show that different views indeed complement each other or provide redundancy to increase reliability and thus produce better overall sentence recognition results of up to 83% for the combination of the frontal, 30°, 60° and 90° side views. In general, combinations with views such as the 30° and 60° views showed good improvements, highlighting the complementary , or redundant, nature of the information between various view angles. On the contrary, although the 90° pose does not seem to contain as many relevant features to correctly recognise a sentence on its own, combined with other views, especially other lower performing angles such as 45°, it improves recognition rates. This observation implies that there exists a certain amount of complementary information or redundancy between these views as well.

The rest of this chapter is organised as follows. Section 5.2 reviews the proposed method for VSR utilising a PCA network, LSTMs, and the GMM-HMM system and introduces the decision fusion scheme. Section 5.3 presents the dataset, experiments, and obtained results. Finally, Section 5.4 concludes the chapter with a summary and discussions.

5.2 Proposed method

In this work we use a PCA network and LSTM framework developed in Chapter 3 to extract robust features for visual speech recognition inside a tandem GMM-HMM scheme. Unlike previous work, where either multi-stream HMMs or feature fusion are employed, in our method, we also explore a scheme where the decision fusion happens at the end of the recognition pipeline by weighting the log-likelihoods of the paths of the Viterbi algorithm for several views.

In this study we combine the views in the database using two different methods: an early and a late fusion. For the former we fuse the features of multiple views. To this end, the feature vectors obtained from the LSTM are concatenated and then processed similarly to the single views with their delta and acceleration components in a tandem GMM-HMM system (see Figure 5.1).

This simple multi-view scheme is extended by further analyses of results using late fusion with decision fusion techniques. The following fusion scheme of the likelihoods for two views v_a and v_b is used:

$$p(o_{v_a}, o_{v_b}|q = q_i) = p(o_{v_a}|q = q_i)^{\lambda_{v_a}} p(o_{v_b}|q = q_i)^{\lambda_{v_b}} \tag{5.1}$$

where o_{v_a} and o_{v_b} are the observations of the two views, for the speech (viseme) class q_i and with the weights λ_{v_a} and λ_{v_b} constrained by

$$\lambda_{v_a} + \lambda_{v_b} = 1 \tag{5.2}$$

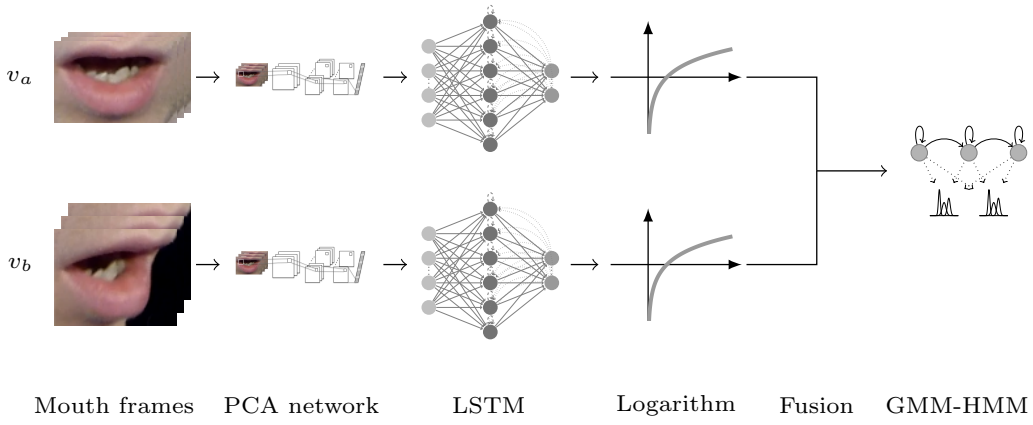


Figure 5.1 – The proposed feature fusion multi-view tandem system with PCA network-LSTMs and GMM-HMMs for visual speech recognition from the mouth video frames.

Taking the logarithm of equation 5.1 to obtain the log-likelihoods like in HTK's implementation of the Viterbi algorithm we have

$$\log(p(o_{v_a}, o_{v_b}|q = q_i)) = \lambda_{v_a} \log(p(o_{v_a}|q = q_i)) + \lambda_{v_b} \log(p(o_{v_b}|q = q_i)) \quad (5.3)$$

To finally fuse the results, the top-5 Viterbi output sequences for each view and utterance are retained together with their log-likelihoods. The weighted log-likelihoods of the two views are summed up and the Viterbi sequence with the highest weighted sum is then selected to obtain the final joint log-likelihood. This approach was similarly extended to multiple views by summing up the weighted log-likelihoods and restricting the sum of their weights to one.

5.3 Performance analysis

This section presents the data evaluated in this study, the experiments conducted, and the results obtained when performing decision fusion across multiple views.

5.3.1 The dataset

The dataset used in this chapter is the short phrase section of the OuluVS2 database [Anina et al., 2015] (see Section 1.8.2 for more detail).

5.3.2 Experimental results

Parameters for the spatiotemporal feature extraction by PCA network and LSTM from the given cropped mouth videos were selected similar to the approach in Chapter 3. First the results of combining the features from different views were analysed. Subsequent experiments were conducted to measure the improvements in performance when combining classifier outputs. These results are compared to the previous recognition rates for single-view.

Three measures of word accuracy, word correctness, and sentence recognition per cent are used for reporting the results, which are defined in Section 1.6.

To adjust our system parameters, we use a leave-one-out cross-validation scheme across speakers on the given training set. Next, we apply the trained system on the test set for the final recognition at the word or phrase levels. We present the best test results obtained for phrase recognition on the OuluVS2 dataset for varying numbers of views combined in Figure 5.4, and the complete test results in Table 5.3.

Feature fusion: Multiple-view experiments

First experiments on the combination of different views involved feature concatenation. These multi-view experiments show interesting results (see Figures 5.2 and 5.3). Various combinations of the frontal view with each of the four side views were tested, as well as the ensemble of all views together.

The multiple-view results show very good improvements especially for the combination of the frontal and the 30°-side view. On the cross-validation set a sentence accuracy of nearly 83% is achieved, while word correctness and word accuracy are around 85% and 84%, respectively. Thus this amounts to improvements of around 3–10% over the separate results for these views. Similar improvements can be seen on the test set, where for the same combination the recognition of sentences reaches 79% and 83% of words are recognised correctly. The word accuracy lies at 81%. These results show that especially between the frontal and the 30°-view there is complementary or redundant information that can be exploited. The improvements for the other views are not as significant – however, there could be further improvements. The concatenation of all the feature vectors from all views shows a particularly bad result. This is probably due to the increase in dimensionality, which could be aided by prior dimensionality reduction techniques.

Furthermore, a large variability in the performance between the different speakers can again be observed from the high standard deviation shown in Figures 5.2 and 5.3 as well as the individual speaker results in Table 5.1.

Table 5.1 – Phrase recognition results (in %) on the combination of different views of the multi-view dataset of OuluVS2 using **feature fusion** in the proposed tandem system with PCA network-LSTMs and GMM-HMMs on the given test set per speaker and the corresponding mean and standard deviation across speakers (SC = Sentence correctness, WC = Word correctness and WA = Word accuracy).

	all views			0° + 30°			0° + 45°			0° + 60°			0° + 90°		
Spkr.	SC	WC	WA	SC	WC	WA	SC	WC	WA	SC	WC	WA	SC	WC	WA
6	30.0	29.3	25.3	76.7	80.0	77.3	63.3	68.0	64.0	40.0	44.0	41.3	50.0	53.3	52.0
8	56.7	61.3	58.7	70.0	74.7	70.7	60.0	64.0	61.3	53.3	57.3	57.3	50.0	54.7	54.7
9	36.7	44.0	38.7	70.0	76.0	73.3	43.3	44.0	41.3	60.0	64.0	62.7	43.3	54.7	45.3
15	70.0	65.3	65.3	70.0	76.0	73.3	66.7	66.7	65.3	70.0	68.0	68.0	70.0	69.3	66.7
26	60.0	56.0	50.7	80.0	84.0	80.0	86.7	88.0	88.0	66.7	62.7	61.3	66.7	66.7	64.0
30	76.7	82.7	81.3	86.7	90.7	90.7	83.3	88.0	88.0	90.0	93.3	93.3	83.3	90.7	88.0
34	96.7	98.7	98.7	86.7	90.7	90.7	96.7	98.7	98.7	96.7	97.3	97.3	90.0	90.7	89.3
43	76.7	80.0	77.3	76.7	81.3	78.7	86.7	89.3	89.3	73.3	80.0	77.3	86.7	85.3	85.3
44	70.0	72.0	72.0	83.3	88.0	88.0	76.7	81.3	78.7	100.0	100.0	100.0	70.0	70.7	70.7
49	80.0	82.7	82.7	86.7	92.0	89.3	86.7	88.0	86.7	90.0	93.3	92.0	83.3	88.0	86.7
51	40.0	37.3	33.3	63.3	61.3	58.7	46.7	33.3	32.0	40.0	36.0	30.7	56.7	56.0	54.7
52	86.7	88.0	86.7	100.0	100.0	100.0	76.7	77.3	76.0	86.7	81.3	81.3	86.7	92.0	88.0
Mean	65.0	66.4	64.2	79.2	82.9	80.9	72.8	73.9	72.4	72.2	73.1	71.9	69.7	72.7	70.4
SD	19.9	20.6	22.2	9.7	9.8	10.8	16.2	18.8	19.5	20.1	20.3	21.4	15.8	15.2	15.8

Decision fusion: Weighting schemes

The aim of these experiments is to investigate whether the complementary information contained in the different views can be exploited by combining the top-5 Viterbi decoder HMM outputs for several angles. To do so, we summed the weighted log-likelihoods for two views to determine which path would have the highest combined log-likelihood.

The optimal weights (see Table 5.2) were obtained through a leave-one-out cross-validation scheme applied only on the training set using the train-test splits of the data as provided by the authors of the dataset. These weights are determined in two ways. In the first method, the weights are based on the sentence correctness of a leave-one-out cross-validation in the training set and are later normalised by the total sum of weights. This measure is referred to as "Training recognition" or "Rec" in Tables 5.2 and 5.3.

Table 5.2 – Optimal weights obtained via grid search and by training performance normalisation.

View combination	Grid search					Training recognition				
$v_a + v_b + v_c + v_d + v_e$	λ_{v_a}	λ_{v_b}	λ_{v_c}	λ_{v_d}	λ_{v_e}	λ_{v_a}	λ_{v_b}	λ_{v_c}	λ_{v_d}	λ_{v_e}
0°	-	-	-	-	-	-	-	-	-	-
30°	-	-	-	-	-	-	-	-	-	-
45°	-	-	-	-	-	-	-	-	-	-
60°	-	-	-	-	-	-	-	-	-	-
90°	-	-	-	-	-	-	-	-	-	-
0° + 30°	0.4	0.6	-	-	-	0.5	0.5	-	-	-
0° + 45°	0.6	0.4	-	-	-	0.6	0.4	-	-	-
0° + 60°	0.9	0.1	-	-	-	0.5	0.5	-	-	-
0° + 90°	0.7	0.3	-	-	-	0.6	0.4	-	-	-
30° + 45°	0.8	0.2	-	-	-	0.6	0.4	-	-	-
30° + 60°	0.6	0.4	-	-	-	0.5	0.5	-	-	-
30° + 90°	0.9	0.1	-	-	-	0.6	0.4	-	-	-
45° + 60°	0.4	0.6	-	-	-	0.5	0.5	-	-	-
45° + 90°	0.8	0.2	-	-	-	0.5	0.5	-	-	-
60° + 90°	0.7	0.3	-	-	-	0.5	0.5	-	-	-
0° + 30° + 45°	0.4	0.5	0.1	-	-	0.4	0.4	0.2	-	-
0° + 30° + 60°	0.4	0.4	0.2	-	-	0.3	0.3	0.4	-	-
0° + 30° + 90°	0.4	0.6	0.0	-	-	0.4	0.4	0.2	-	-
0° + 45° + 60°	0.8	0.1	0.1	-	-	0.4	0.3	0.3	-	-
0° + 45° + 90°	0.8	0.1	0.1	-	-	0.4	0.3	0.3	-	-
0° + 60° + 90°	0.8	0.1	0.1	-	-	0.4	0.3	0.3	-	-
30° + 45° + 60°	0.6	0.0	0.4	-	-	0.4	0.3	0.3	-	-
30° + 45° + 90°	0.8	0.1	0.1	-	-	0.4	0.3	0.3	-	-
30° + 60° + 90°	0.8	0.1	0.1	-	-	0.4	0.3	0.3	-	-
45° + 60° + 90°	0.1	0.8	0.1	-	-	0.3	0.4	0.3	-	-
0° + 30° + 45° + 60°	0.3	0.4	0.1	0.2	-	0.3	0.3	0.2	0.2	-
0° + 30° + 45° + 90°	0.4	0.4	0.1	0.1	-	0.3	0.3	0.2	0.2	-
0° + 30° + 60° + 90°	0.4	0.4	0.1	0.1	-	0.3	0.3	0.2	0.2	-
0° + 45° + 60° + 90°	0.8	0.1	0.0	0.1	-	0.3	0.2	0.3	0.2	-
30° + 45° + 60° + 90°	0.7	0.1	0.1	0.1	-	0.3	0.2	0.3	0.2	-
0° + 30° + 45° + 60° + 90°	0.9	0.1	0.0	0.0	0.0	0.2	0.2	0.2	0.2	0.2

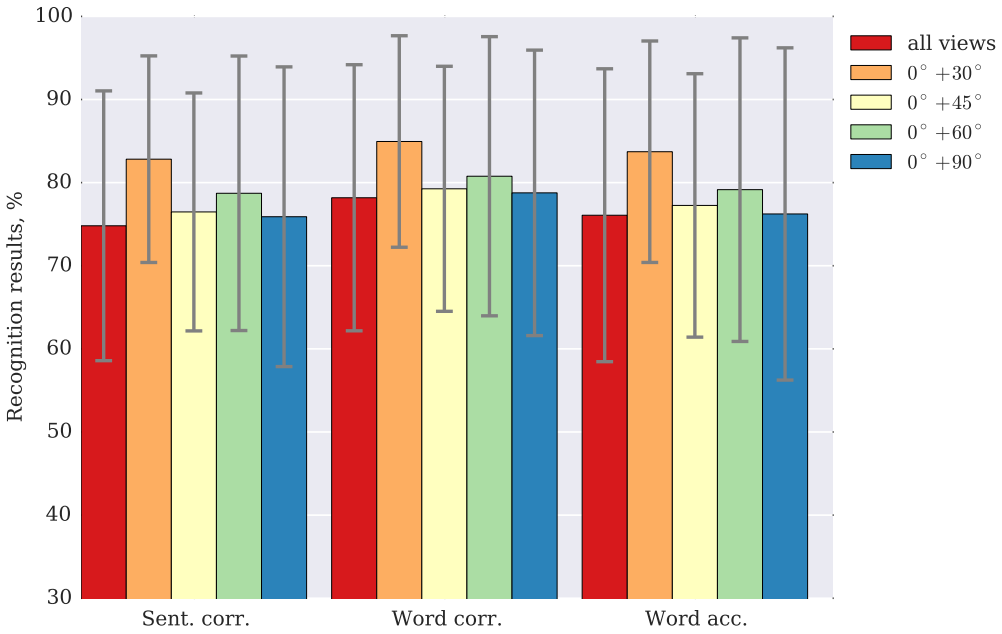


Figure 5.2 – Mean phrase recognition results on the combination of different views of the multi-view dataset of OuluVS2 using **feature fusion** in the proposed tandem system with PCA network-LSTMs and GMM-HMMs with the cross-validation technique on the whole dataset. Error bars denote the standard deviation across subjects (sent. = sentence, corr. = correctness, acc. = accuracy).

In the second approach, optimal weights were obtained by iterating over all possible values between 0 and 1 in 0.1 steps and choosing the highest performing one on the cross-validation of the training set, while observing the constraint of equation 5.2. These weights are referred to as "Grid search" or "Grid" in Tables 5.2 and 5.3. Afterwards, these weights were used on the test set.

Evaluating the weights obtained for various cases in Table 5.2, we can already see that for grid search the weights tend to be higher especially for the frontal and 30° side view. This does not seem very surprising, since these views also have the highest performance when taken separately. Since every time the best results based on the cross-validation of the training set are used, we can still see variations on the performance of the test set. For example, the optimal combination of all views only gives non-zero weights to the 0° and 30° view angles, though still at different rates from the simple two-view combination of those two, and in the end still provides poorer test results.

The weighting scheme based on the training recognition results shows very balanced weights, since the training results are fairly similar so that rounding the weights to the nearest tenth results in very close values. This means, for the same example of the combination of all views, that all views are weighted equally. In the end, the test results

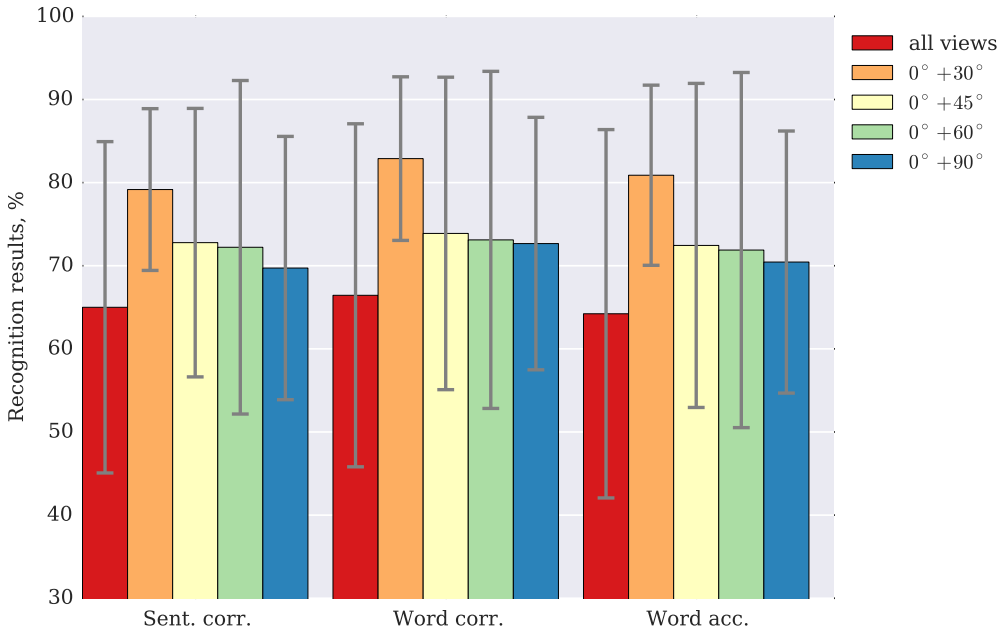


Figure 5.3 – Mean phrase recognition results on the combination of different views of the multi-view dataset of OuluVS2 using **feature fusion** in the proposed tandem system with PCA network-LSTMs and GMM-HMMs on the given test set. Error bars denote the standard deviation across subjects (sent. = sentence, corr. = correctness, acc. = accuracy).

in sentence correctness are similar to the other weighting scheme, however, word accuracy and correctness are much lower.

Decision fusion: Multiple-view experiments

Table 5.3 shows the results of various combinations on the test set. It contains the results obtained with the two weighting schemes, as well as the baseline results from Chapter 3 under "Feat". These are the single-view test results and the combinations through feature concatenation presented in the previous section. When evaluating the results, it should be taken into account that we do not perform a simple 10-class classification, but rather make use of a typical speech evolution process with HMMs, modelling word sequences. This includes silence as a possible utterance, which is only removed for evaluation purposes, in order not to distort the results.

Looking at the results in Table 5.3 we can see that there are various improvements over the baseline results for the 30°-side view (the highest performing single view). While this view on its own has a sentence correctness of around 76% on the test set, we can see that both through feature concatenation, and by combining classifier results with the

Table 5.3 – Mean phrase recognition results (in %) across test subjects on the combination of different views of the OuluVS2 multi-view dataset using the proposed method.

View combination	Sentence Correctness			Word Accuracy			Word Correctness		
	Grid	Rec	Feat	Grid	Rec	Feat	Grid	Rec	Feat
0°	-	-	73.1	-	-	73.0	-	-	74.1
30°	-	-	75.6	-	-	75.2	-	-	76.8
45°	-	-	67.2	-	-	66.6	-	-	68.7
60°	-	-	63.3	-	-	60.6	-	-	63.7
90°	-	-	59.2	-	-	56.8	-	-	63.1
0° + 30°	79.4	80.0	79.2	79.9	80.8	82.9	80.6	81.6	80.9
0° + 45°	76.1	76.1	72.8	76.4	76.4	73.9	77.3	77.3	72.4
0° + 60°	77.2	74.7	72.2	77.9	74.6	73.1	78.9	75.7	71.9
0° + 90°	75.6	76.4	69.7	76.4	76.9	72.7	77.7	77.8	70.4
30° + 45°	77.2	76.9	-	77.4	77.2	-	78.8	78.6	-
30° + 60°	78.1	74.7	-	77.7	73.8	-	79.0	75.7	-
30° + 90°	76.7	76.9	-	77.7	77.9	-	79.2	79.4	-
45° + 60°	69.7	72.2	-	67.6	70.4	-	69.8	72.6	-
45° + 90°	72.5	71.9	-	71.9	71.3	-	73.9	73.8	-
60° + 90°	66.7	67.5	-	65.8	66.2	-	68.4	69.9	-
0° + 30° + 45°	82.3	80.4	-	81.3	79.6	-	81.1	79.7	-
0° + 30° + 60°	82.0	83.1	-	81.4	82.7	-	80.6	81.9	-
0° + 30° + 90°	80.6	82.3	-	79.9	81.3	-	79.4	80.3	-
0° + 45° + 60°	80.9	79.8	-	80.1	79.0	-	79.4	78.9	-
0° + 45° + 90°	78.0	79.4	-	77.2	78.4	-	76.7	78.1	-
0° + 60° + 90°	78.3	78.2	-	77.3	77.0	-	76.1	76.1	-
30° + 45° + 60°	79.0	78.3	-	77.7	77.1	-	78.1	77.2	-
30° + 45° + 90°	80.1	79.9	-	78.8	78.9	-	78.1	78.1	-
30° + 60° + 90°	80.1	79.0	-	78.8	77.4	-	78.1	76.9	-
45° + 60° + 90°	70.6	72.7	-	68.1	70.8	-	69.2	71.4	-
0° + 30° + 45° + 60°	82.7	81.1	-	82.1	80.2	-	81.7	79.4	-
0° + 30° + 45° + 90°	82.7	82.7	-	81.7	81.6	-	80.6	80.3	-
0° + 30° + 60° + 90°	82.1	83.3	-	81.4	82.7	-	80.3	81.4	-
0° + 45° + 60° + 90°	78.0	80.2	-	77.2	79.2	-	76.7	78.1	-
30° + 45° + 60° + 90°	80.0	78.3	-	78.7	76.4	-	77.5	76.1	-
0° + 30° + 45° + 60° + 90°	75.0	75.7	65.0	72.6	67.8	66.4	72.8	66.9	64.2

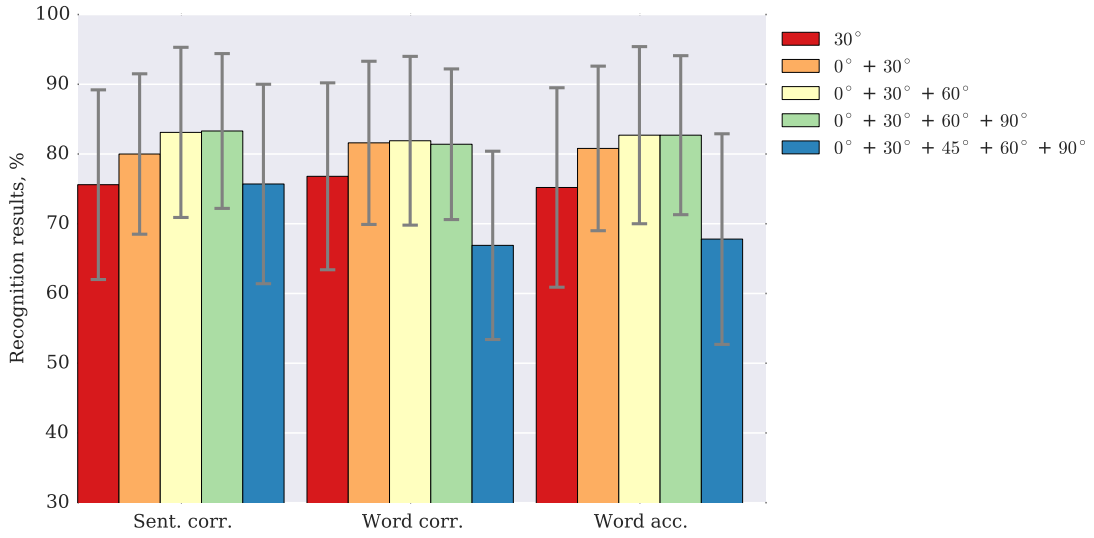


Figure 5.4 – Best mean phrase recognition results (in %) across test subjects for each number of views per combination of the OuluVS2 multi-view dataset using the proposed tandem system. Error bars denote the standard deviation across subjects (sent. = sentence, corr. = correctness, acc. = accuracy).

frontal view we can achieve a sentence correctness of around 80%. For other types of combinations, the impact of combining classifiers over features becomes more apparent: all the combinations involving either the frontal or the 30° side view achieve a sentence correctness of at least 76%, while most of the feature concatenation schemes stay around 70%. Similar trends are also observed for the word accuracy and word correctness.

It is interesting to note that, aside from the frontal and 30° views, the combination of the 30° and 60° side views give particularly significant improvements, showing the degree of complementarity or redundancy between these views. In general, it is evident that the frontal, 30° and 60° side views contain the most information and only some complementary or redundant information can be exploited in the 45° and 90° views.

Comparing the combinations of more than just two views, we can see that they improve the results further. This is true for almost all combinations, but especially in combining the frontal and 30° views with the 60° we can reach sentence recognition results of 83%. A slightly better correctness is also still achieved when adding the 90° side view.

When finally combining all views, we see a drop in performance again. This is probably due to the effect described regarding the weighting, that on the cross-validation of the training set a certain weighting only taking into account limited views achieves the highest performance, which is not the case on the test set.

The above discussions only take into account the sentence correctness. However, similar trends can be observed looking at both word accuracy and word correctness.

5.4 Summary

In this chapter, we have explored the influence of multi-view fusion on visual speech recognition. The results have shown that exploiting multiple-view data can improve the recognition results significantly. This is particularly true for combinations involving the frontal view, the 30° and 60° view angles. The other views do provide some additional information, however, the improvements are not as noteworthy.

In this work we have extended our previous experiments to include a more in-depth study of various combinations of different view angles. However, this study still has several limitations: First, it only takes into account a simple decision fusion scheme of the log-likelihoods of various Viterbi paths. Furthermore, the dataset is limited to simple phrase recognition. Future work should further extend this effort to test other fusion schemes and to evaluate them on larger databases.

Conclusion

In this thesis we compare the results of three different approaches to sequence-to-sequence visual speech recognition. These approaches are the traditional approach with handcrafted features and a GMM-HMM sequential model, the combined approach with deep learning-based features and a subsequent GMM-HMM model, and finally the deep learning approach where deep learning is used for feature extraction and sequence modelling, together with a decoding technique to allow sequence-to-sequence recognition.

The results are compared for two different databases. The databases have different advantages and are used to demonstrate a few points in this work. On the one hand, the OuluVS2 phrases dataset is relatively small in terms of number of sentences per speaker – in particular the vocabulary is very limited since each phrase is repeated three times per speaker. However, this dataset comprises five different view angles, which is very useful for a complete study of multi-view fusion. On the other hand, the TCD-TIMIT dataset, having around the same number of speakers as OuluVS2, has more sentences per speaker which contain a larger variety of vocabulary and thus this dataset is more adapted to viseme-level recognition and, due to its larger size, is more suitable for deep learning methods. However, there are only two different views, which means that the study of multiple-view processing cannot be as complete. Combining the conclusions drawn from both datasets is thus essential.

We could show that using deep learning frameworks, both for feature extraction and for sequence modelling, improves the results. However, finding a deep neural network that works well is not an easy task. There are many different architectures that have to be compared and evaluated, to find out what is the best choice. Furthermore, a number of parameters need to be decided on, which is again a combination of literature research and empirical testing. Finally, the number of parameters in a specific model is important, since, with computational processing and memory constraints of a single GPU, it is important to be able to train the model, i.e. keep the parameters and each training batch in memory.

Contrary to some other recent work, we make sure that the modelling of the utterances is at a sequence-to-sequence level, opposed to the classification at the word or the entire phrase level. This means, that our findings can be extrapolated to continuous speech

recognition and similar approaches can be used for work on large vocabulary databases.

In addition, we have demonstrated that combining several views is advantageous to exploit the complementary information contained in each of the views. There are various techniques for multi-view processing, and we compare feature fusion techniques to late fusion at the decision level. We can clearly observe the advantage of late fusion. A major problem for feature fusion is the increase of the number of model parameters for each additional view, which would require more training data to remain at the same level of accuracy for each additional view. Since we do not have more training data available, it is more feasible to train a feature extraction and sequence model for each view and then weight the suggested sequences, which shows a good performance increase compared to the single view.

Perspectives

To continue the work in this thesis, a good starting point would be to use or collect multi-view data from a real-life scenario. This could be recording a subject while driving or on a mobile device, with several cameras. This data could then be used to train models which, firstly, are independent of the exact view angle – which is not the case in this thesis – and secondly, integrate in a multimodal processing scheme, where the views are weighted depending on the contribution they can bring to the overall system.

Evidently, in this field it is necessary to have sufficiently large amounts of data. Therefore, the data collected should include many subjects, at least comparable to the two databases used in this work, and include many sentences per speaker, with a large range of vocabulary.

Furthermore, the work should be extended to various use cases in which visual speech recognition can be useful, this can be audio-visual speech recognition in noisy scenarios, or as a help to distinguish mouthings in sign language.

Perhaps further in the future, one could dream up a universal speech recognition system that adapts to a particular situation – and not only regarding the noise level for audio-visual speech recognition, but also adapts if the speaker is whispering, using sign language or just articulating without sound. This would require a multimodal approach to detect whether there is sound (and if yes, if it is related to the articulation), or signing, or none of these, and shift the focus of the recogniser accordingly between the modalities.

Other interesting applications of visual speech recognition could be for the study of speech production and speaking disorders. The visible articulation of the mouth could help to understand how certain sounds are changed through a disorder, to determine the type of speaking disorder. Along the same line, visual speech recognition could be an additional tool in computer or mobile phone-based pronunciation and language learning,

where the patient or learner gets additional help on how to move the lips and articulate according to their performance.

Finally, in an increasingly digital world, where not only banking, but also governments move more and more administration online, cybersecurity is an important topic. To combat counterfeiting, ever increasing security layers are added, among which more and more are video-based methods to recognise the user. Here the 'whispering' or 'silent articulation' of passwords can, for example, directly include several security features: the password itself, the way the user articulates and the user's appearance.

Bibliography

I. Anina, Z. Zhou, G. Zhao, and M. Pietikainen. 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)*, 2015.

Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas. Lipnet: Sentence-level lipreading. *CoRR*, abs/1611.01599, 2016. URL http://arxiv.org/abs/1611.01599.

P. Badin, G. Bailly, L. Revéret, M. Baciu, C. Segebarth, and C. Savariaux. Three-dimensional linear articulatory modeling of tongue lips and face, based on MRI and video images. *Journal of Phonetics*, 30(3):533–553, 2002.

D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio. End-to-End Attention-based Large Vocabulary Speech Recognition. In *2016 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2016.

S. Basu and A. Pentland. A three-dimensional model of human lip motions trained from video. In *IEEE Proceedings of the Nonrigid and Articulated Motion Workshop*, number 441, pages 46–53, 1997.

H. L. Bear, G. Owen, R. Harvey, and B.-J. Theobald. Some observations on computer lip-reading: moving from the dream to the reality. In *Proceedings of SPIE*, volume 9253, pages 92530G–92530G–10, 2014.

C. M. Bishop. *Pattern Recognition And Machine Learning*. Springer, 2006.

A. Biswas, P.K. Sahu, and M. Chandra. Multiple camera in car audio–visual speech recognition using phonetic and visemic information. *Computers & Electrical Engineering*, 47:35–50, 2015.

H. A. Bourlard and N. Morgan. *Connectionist Speech Recognition*. Springer Nature, 1994.

R. Bowden, S. Cox, R. Harvey, Y. Lan, E.-J. Ong, G. Owen, and B.-J. Theobald. Is automated conversion of video to text a reality? In *Proceedings of SPIE*, volume 8546, pages 85460U–85460U–9, 2012.

R. Bowden, S. Cox, R. Harvey, Y. Lan, E.-J. Ong, G. Owen, and B.-J. Theobald. Recent developments in automated lip-reading. In *Proceedings of SPIE*, pages 89010J–89010J–13, 2013.

D. Burnham, E. Ambikairajah, J. Arciuli, M. Bennamoun, C. T. Best, S. Bird, A. R. Butcher, S. Cassidy, G. Chetty, F. M. Cox, A. Cutler, R. Dale, J. R. Epps, J. M, R. Goecke, D. B. Grayden, J. T. Hajek, J. C. Ingram, S. Ishihara, N. Kemp, T. Kuratate, T. W. Lewis, D. E. Loakes, M. Onslow, D. M. W. Powers, P. Rose, R. Togneri, D. Tran, and M. Wagner. A Blueprint for a Comprehensive Australian English Auditory-Visual Speech Corpus. *Science And Technology*, pages 96–107, 2009.

L. Cappelletta and N. Harte. Phoneme-to-viseme mapping for visual speech recognition. In *Proceedings of the International Conference on Patter Recognition Applications and Methods*, pages 322–329, 2012.

T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma. PCANet: A Simple Deep Learning Baseline for Image Classification? *IEEE Transactions on Image Processing*, 24(12):5017–5032, 2015.

W. Chan, N. Jaitly, Q. Le, and O. Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 4960–4964, 2016.

A. G. Chitu and L. J. M. Rothkrantz. On dual view lipreading using high speed camera. In *Euromedia '2008*, pages 45–51. Eurosis, 2008.

K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, 2014.

J. S. Chung and A. Zisserman. Out of Time: Automated Lip Sync in the Wild. In *Computer Vision – ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II*, pages 251–263. 2017.

J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman. Lip Reading Sentences in the Wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

M. Cooke, J. Barker, S. Cunningham, and X. Shao. An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 120(5 Pt 1):2421–2424, 2006.

T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

P. Coxhead. *Natural Language Processing & Applications: Phones and Phonemes*. Lecture notes, 2006. URL https://www.cs.bham.ac.uk/~pxc/nlp/.

G. L. Cuendet. *Towards 3D facial morphometry : facial image analysis applications in anesthesiology and 3D spectral nonrigid registration*. PhD thesis, Ecole polytechnique fédérale de Lausanne, 2017.

B. Denby, T. Schultz, K. Honda, T. Hueber, J.M. Gilbert, and J.S. Brumberg. Silent speech interfaces. *Speech Communication*, 52(4):270 – 287, 2010. Silent Speech Interfaces.

J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko. Long-term recurrent convolutional networks for visual recognition and description. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

V. Estellers and J.-P. Thiran. Multi-pose lipreading and audio-visual speech recognition. *EURASIP Journal on Advances in Signal Processing*, 2012(1):51, 2012.

A. Fernandez-Lopez, O. Martinez, and F. M. Sukno. Towards Estimating the Upper Bound of Visual-Speech Recognition: The Visual Lip-Reading Feasibility Database. In *Proceedings - 12th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2017 - 1st International Workshop on Adaptive Shot Learning for Gesture Understanding and Production, ASL4GUP 2017, Biometrics in the Wild, Bwild 2017, Heteroge*, pages 208–215, 2017.

J. Freitas, A. Teixeira, and M. S. Dias. Multimodal Corpora for Silent Speech Interaction. *Proceedings of Language Resources and Evaluation Conference (LREC)*, 2014.

M. Gales and S. Young. The application of hidden markov models in speech recognition. *Foundations Trends Signal Processing*, 1(3):195–304, 2007.

J. S. Garofolo et al. TIMIT Acoustic-Phonetic Continuous Speech Corpus. https://catalog.ldc.upenn.edu/LDC93S1, 1993. Accessed: 2018-04-16.

A. Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. PhD thesis, Technical University Munich, 2008.

A. Graves and N. Jaitly. Towards End-To-End Speech Recognition with Recurrent Neural Networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1764–1772. JMLR Workshop and Conference Proceedings, 2014.

A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber. Connectionist Temporal Classification : Labelling Unsegmented Sequence Data with Recurrent Neural Networks. *Proceedings of the 23rd international conference on Machine Learning*, pages 369–376, 2006.

A. Graves, A. Mohamed, and G. Hinton. Speech Recognition With Deep Recurrent Neural Networks. In *2013 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 6645–6649, 2013.

S. Gurbuz, E. Patterson, Z. Tufekci, and J. N. Gowdy. Lip-reading from parametric lip contours for audio-visual speech recognition. In *EUROSPEECH 2001 Scandinavia, 7th European Conference on Speech Communication and Technology, 2nd INTERSPEECH Event, Aalborg, Denmark, September 3-7, 2001*, pages 1181–1184, 2001.

A. Hannun. Sequence modeling with CTC. *Distill*, 2017. https://distill.pub/2017/ctc.

N. Harte and E. Gillen. TCD-TIMIT: An Audio-Visual Corpus of Continuous Speech. *IEEE Trans. Multimedia*, 17(5):603–615, 2015.

A. Hassanat. Visual Passwords Using Automatic Lip Reading. *International Journal of Sciences: Basic and Applied Research (IJSBAR)*, 13(1), 2014.

H. Hermansky, D.P.W. Ellis, and S. Sharma. Tandem connectionist feature extraction for conventional HMM systems. In *2000 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2000.

S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9 (8):1735–1780, 1997.

A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR*, abs/1704.04861, 2017. URL http://arxiv.org/abs/1704.04861.

J. Huang and B. Kingsbury. Audio-visual deep learning for noise robust speech recognition. In *2013 IEEE International Conference on Acoustics Speech and Signal Processing*, 2013.

D. Ivanko, A. Karpov, D. Ryumin, I. Kipyatkova, A. Saveliev, V. Budkov, D. Ivanko, and M. Železný. Using a high-speed video camera for robust audio-visual speech recognition in acoustically noisy conditions. In *Speech and Computer*, pages 757–766. Springer International Publishing, 2017.

D. Jachimski, A. Czyzewski, and T. Ciszewski. A comparative study of English viseme recognition methods and algorithms. *Multimedia Tools and Applications*, 2017.

J. Jeffers and M. Barley. *Speechreading (lipreading)*. Thomas Springfield, 1971.

S. Kim, T. Hori, and S. Watanabe. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4835–4839, 2017.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

O. Koller, H. Ney, and R. Bowden. Read My Lips: Continuous Signer Independent Weakly Supervised Viseme Recognition. In *Computer Vision – ECCV 2014*, pages 281–296. Springer International Publishing, 2014.

O. Koller, H. Ney, and R. Bowden. Deep Learning of Mouth Shapes for Sign Language. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015.

A. Koumparoulis, G. Potamianos, Y. Mroueh, and S. J. Rennie. Exploring ROI size in deep learning based lipreading. In *International Conference on Auditory-Visual Speech Processing*, 2017.

Y. Lan, B.-J. Theobald, R. Harvey, E.-J. Ong, and R. Bowden. Improving visual features for lip-reading. In *Proceedings of International Conference on Auditory-Visual Speech Processing*, 2010.

Y. Lan, B.-J. Theobald, and R. W. Harvey. View independent computer lip-reading. In *Proceedings - IEEE International Conference on Multimedia and Expo*, pages 432–437, 2012.

Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten Digit Recognition with a Back-Propagation Network. In *Advances In Neural Information Processing Systems*, 1990.

B. Lee, M. Hasegawa-Johnson, C. Goudeseune, S. Kamdar, S. Borys, M. Liu, and T. Huang. AVICAR : Audio-Visual Speech Corpus in a Car Environment. In *8th International Conference on Spoken Language Processing*, 2004.

D. Lee, J. Lee, and K.-e. Kim. Multi-view Automatic Lip-Reading Using Neural Network. In *Computer Vision – ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II*, pages 290–302. 2017.

P. Lucey, G. Potamianos, and S. Sridharan. An Extended Pose-Invariant Lipreading System. In *Proceedings of AVSP'07: International Conference on Auditory-Visual Speech Processing*. International Speech Communication Association, 2007.

J. Luettin, N. A. Thacker, and S. W. Beet. Visual speech recognition using active shape models and hidden Markov models. In *1996 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 817–820, 1996.

H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 264:746–748, 1976.

Y. Mroueh, E. Marcheret, and V. Goel. Deep multimodal learning for Audio-Visual Speech Recognition. In *2015 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2015.

R. Navarathna, D. Dean, S. Sridharan, and P. Lucey. Multiple cameras for audio-visual speech recognition in an automotive environment. *Computer Speech & Language*, 27 (4):911–927, 2013.

C. Neti, G. Potamianos, J. Luettin, I. Matthews, H. Glotin, D. Vergyri, J. Sison, A. Mashari, and J. Zhou. Audio-visual speech recognition. Technical report, 2000.

J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal Deep Learning. *Proceedings of The 28th International Conference on Machine Learning (ICML)*, pages 689–696, 2011.

K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata. Audio-visual speech recognition using deep learning. *Applied Intelligence*, 42(4):722–737, 2014.

C. Olah. Understanding LSTM networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015. Accessed: 2018-04-09.

A. Ortega, F. Sukno, E. Lleida, A. Frangi, A. Miguel, L. Buera, and E. Zacur. AV@CAR: A Spanish Multichannel Multimodal Corpus for In-Vehicle Automatic Audio-Visual Speech Recognition. In *Proceedings of Language Resources and Evaluation Conference (LREC)*, pages 763–766, 2004.

E. K. Patterson, S. Gurbuz, Z. Tufekci, and J. N. Gowdy. Moving-Talker, Speaker-Independent Feature Study, and Baseline Results Using the CUAVE Multimodal Speech Corpus. *EURASIP Journal on Applied Signal Processing*, 11:1189–1201, 2002.

S. Petridis, Z. Li, and M. Pantic. End-To-End Visual Speech Recognition With LSTMs. In *2017 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2017a.

S. Petridis, Y. Wang, Z. Li, and M. Pantic. End-to-End Multi-View Lipreading. In *British Machine Vision Conference*, London, 2017b.

S. Petridis, J. Shen, D. Cetin, and M. Pantic. Visual-only recognition of normal, whispered and silent speech. In *Accepted to 2018 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2018.

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

G. Potamianos, A. Verma, C. Neti, G. Iyengar, and S. Basu. A cascade image transform for speaker independent automatic speechreading. In *IEEE International Conference on Multimedia and Expo (ICME)*, volume 2, pages 1097–1100 vol.2, 2000.

G. Potamianos, C. Neti, J. Luettin, and I. Matthews. Audio-Visual Automatic Speech Recognition : An Overview. In G. Bailly, Eric Vatikiotis-Bateson, and P. Perrier, editors, *Issues in Visual and Audio-Visual Speech Processing*, chapter 10, pages 1–30. MIT Press, 2004.

C. Qu, H. Gao, E. Monari, J. Beyerer, and J.-P. Thiran. Towards robust cascaded regression for face alignment in the wild. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2015-October(April):1–9, 2015.

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

T. Saitoh, Z. Zhou, G. Zhao, and M. Pietikäinen. Concatenated Frame Image Based CNN for Visual Speech Recognition. In *Computer Vision – ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II*, pages 277–289. 2017.

C. Schmidt and O. Koller. Using viseme recognition to improve a sign language translation system. In *International Workshop on Spoken Language Translation*, pages 197–203, Heidelberg, Germany, 2013.

M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45:2673 – 2681, 1997.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

N. Souviraà-Labastie and F. Bimbot. Low latency and tight resources viseme recognition from speech using an artificial neural network. Technical report, 2013. URL http://hal.inria.fr/hal-00848629/.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

T. Stafylakis and G. Tzimiropoulos. Combining residual networks with LSTMs for lipreading. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 3652–3656, 2017.

G. Sterpu and N. Harte. Towards Lipreading Sentences with Active Appearance Models. In *International Conference on Auditory-Visual Speech Processing*, 2017.

R. Su, L. Wang, and X. Liu. Multimodal learning using 3D audio-visual data for audio-visual speech recognition. In *2017 International Conference on Asian Language Processing (IALP)*, pages 40–43, 2017.

C. Sui, M. Bennamoun, and R. Togneri. Listening with Your Eyes: Towards a Practical Visual Speech Recognition System Using Deep Boltzmann Machines. In *2015 IEEE International Conference on Computer Vision (ICCV)*. Institute of Electrical & Electronics Engineers (IEEE), 2015.

K. Thangthai and R. Harvey. Improving computer lipreading via DNN sequence discriminative training techniques. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 3657–3661, 2017.

K. Thangthai, R. Harvey, S. Cox, and B.-J. Theobald. Improving lip-reading performance for robust audiovisual speech recognition using DNNs Acknowledgments. In *FAAVSP - The 1st Joint Conference on Facial Analysis, Animation, and Auditory-Visual Speech Processing*, 2015.

K. Thangthai, H. L. Bear, and R. Harvey. Comparing phonemes and visemes with DNN-based lipreading. In *BMVC Lipreading Workshop 2017*, 2017.

I. S. Topkaya and H. Erdogan. SUTAV: A Turkish Audio-Visual Database. In *Proceedings of Language Resources and Evaluation Conference (LREC)*, pages 2334–2337, 2012.

J. Trojanová, M. Hrúz, P. Campr, and M. Zelezny. Design and Recording of Czech Audio-Visual Database with Impaired Conditions for Continuous Speech Recognition. In *Proceedings of Language Resources and Evaluation Conference (LREC)*, pages 1239–1243, 2008.

A. Turkmani. *Visual Analysis of Viseme Dynamics*. PhD thesis, University of Surrey, 2007.

P. Viola and M. Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.

A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

A. Vorwerk, X. Wang, D. Kolossa, S. Zeiler, and R. Orglmeister. WAPUSK20-A database for robust audiovisual speech recognition. In *Proceedings of Language Resources and Evaluation Conference (LREC)*, pages 3016–3019, 2010.

M. Wand, J. Koutnik, and J. Schmidhuber. Lipreading with long short-term memory. In *2016 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2016.

T. Watanabe, K. Katsurada, and Y. Kanazawa. Lip Reading from Multi View Facial Images Using 3D-AAM. In *Computer Vision – ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II*, pages 303–316. 2017.

X. Wei, L. Yin, Z. Zhu, and Q. Ji. Avatar-mediated face tracking and lip reading for human computer interaction. In *12th annual ACM International Conference on Multimedia*, pages 500–503, 2004.

X. Xiong and F. De la Torre. Supervised Descent Method and its Application to Face Alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

K. Xu, D. Li, N. Cassimatis, and X. Wang. LCANet: End-to-End Lipreading with Cascaded Attention-CTC. *CoRR*, abs/1803.04988, 2018. URL http://arxiv.org/abs/1803.04988.

Y. Yasui, N. Inoue, K. Iwano, and K. Shinoda. Multimodal Speech Recognition Using Mouth Images from Depth Camera. In *APSIPA Annual Summit and Conference 2017*, 2017.

S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland. The HTK Book (for HTK Version 3.4). Technical report, 2009. URL http://htk.eng.cam.ac.uk.

D. Yu, O. Ghita, A. Sutherland, and P. F. Whelan. A new visual speech modelling approach for visual speech recognition. *Journal of Computer and Information Technology*, 2011.

M. Zimmermann, M. Mehdipour Ghazi, H. K. Ekenel, and J.-P. Thiran. Visual Speech Recognition Using PCA Networks and LSTMs in a Tandem GMM-HMM System. In *Computer Vision – ACCV 2016 Workshops: ACCV 2016 International Workshops, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II*, pages 264–276. 2017a.

M. Zimmermann, M. Mehdipour Ghazi, H. K. Ekenel, and J.-P. Thiran. Combining Multiple Views for Visual Speech Recognition. In *International Conference on Auditory-Visual Speech Processing*, 2017b.

Marina Zimmermann

Machine Learning, Computer Vision and Speech Recognition researcher

Avenue du Léman 89
1005 Lausanne, Switzerland
+41 (76) 652 8855
marina.z@gmx.net
marina-zimmermann

Education

03/2014–present **Ecole polytechnique fédérale de Lausanne (EPFL)**, *Lausanne, Switzerland*.
PhD on the topic 'Multi-view visual speech recognition'

09/2011–07/2013 **Ecole polytechnique fédérale de Lausanne (EPFL)**, *Lausanne, Switzerland*.
MSc in Electrical and Electronics Engineering

08/2008–06/2011 **Jacobs University Bremen**, *Bremen, Germany*.
BSc in Electrical and Computer Engineering

08/2005–06/2006 **Ryde High School**, *Isle of Wight, UK*.
High school year abroad

08/1999–06/2008 **Gymnasium Heißen**, *Mülheim, Germany*.
Allgemeine Hochschulreife (Abitur)

Professional experience

09/2013–present **Ecole polytechnique fédérale de Lausanne (EPFL)**, *Lausanne, Switzerland*.
Research assistant, Signal Processing Laboratory (LTS5) under supervision of Prof. J.-Ph. Thiran

- Significantly improved speech recognition performance when combining audio and visual modalities in project for **PSA Peugeot-Citroën**
- Developing a visual speech recognition system based on deep learning (CNNs + RNNs) using Tensorflow with Python
- Training machine learning models on dedicated GPUs and EPFL's HPC clusters
- Collaborated with other colleagues on the development of 3D facial model for various facial expressions, working on a library written in C++
- EU Horizon 2020 **ADAS&ME** project: Leading activity on sensor development by various industrial and research partners; presenting, coordinating and collaborating LTS5 work with other partners
- Supervising various Bachelor and Master student projects on topics within computer vision and visual speech recognition using machine learning
- Teaching assistant for course 'Traitement des signaux' for Bachelor students

02/2013–07/2013 **Ecole polytechnique fédérale de Lausanne (EPFL)**, *Lausanne, Switzerland*.
Master project, Signal Processing Laboratory (LTS5)

- Developed driver sleepiness detection algorithm from video in collaboration with **PSA Peugeot-Citroën** (Press coverage by 24 heures)

07/2012–01/2013 **Koemei SA**, *Martigny, Switzerland*.
Intern and working student

- Introduced the detection of music to an existing speech recognition system
- Developed API clients in Python and PHP

09/2012–01/2013 **Ecole polytechnique fédérale de Lausanne (EPFL)**, *Lausanne, Switzerland*.
Semester project, Applied Signal Processing Group

- Analysed joint dynamics of RR and QT intervals for sympatho-vagal balance assessment

02/2012–06/2012 **Ecole polytechnique fédérale de Lausanne (EPFL)**, *Lausanne, Switzerland*.
Semester project competition, School of Engineering

- Robot competition in teams of three: design and control robots to sort, collect and throw away different types of waste in an arena with diverse types of ground. Focus inside the team: image processing and computer vision

10/2010–05/2011 **Jacobs University Bremen**, *Bremen, Germany*.
Teaching assistant

- Prepared and gave tutorials, corrected homework for the course Digital Signal Processing
- Assembled exercises for the course Digital Signal Processing in Latex

06/2010–08/2010 **Texas Instruments Deutschland GmbH**, *Freising, Germany*.
Internship

- Enhanced GANG430 (industrial tool for programming of MSP430 devices) Test Environment to avoid manual tests
- Developed LabView usage example for the GANG430

Languages

German	mother tongue	English	fluent (C2)
French	advanced (C1)	Spanish	basic knowledge (A2)

Technical skills

Programming	Python (4+ years), C (2+ years), C++ (3+ years), Matlab (3+ years)
Tools, Libraries	Tensorflow, Keras, OpenCV, HTK, Bash, Perl, Latex, Scikit-learn

Selected publications

- **M. Zimmermann** *et al.*, "Multi-view Visual Speech Recognition in a Sequence-to-sequence Deep Learning Framework", *in preparation*, 2018.
- G. L. Cuendet, C. R. J. Ecabert, **M. Zimmermann**, H. K. Ekenel and J.-Ph. Thiran, "3D Spectral Nonrigid Registration of Facial Expression Scans", *in preparation*, 2018.
- **M. Zimmermann** *et al.*, "Combining Multiple Views for Visual Speech Recognition", *14th International Conference on Auditory-Visual Speech Processing (AVSP)*, 2017.
- **M. Zimmermann** *et al.*, "Visual Speech Recognition Using PCA Networks and LSTMs in a Tandem GMM-HMM System", *Multi-view Lip-reading Challenges - ACCV*, 2016.

Extracurricular activities

02/2015–present **SOS Eau Giteranyi**, *Applied French course*.

- Fundraising for the drilling of a well in the region of Giteranyi in Burundi
- Collaborating with the Collège Voltaire in Geneva on a course project on water: class visit to supervise students and help with the 'Exp'eau'

08/2014–present **Member of the EDEE Coaching Team**.
Mentoring PhD students of the EDEE doctoral school at EPFL

Teaching toolkit I and II, *'Learning how to teach'*.
Courses by the 'Centre d'appui à l'enseignement' at EPFL