Imperial College London
Department of Computing

Deep Audio-Visual Speech Recognition

Pingchuan Ma

June 2022

Supervised by Prof. Maja Pantic and Dr. Stavros Petridis

Submitted in part fulfilment of the requirements for the degree of PhD in Computing Research and the Diploma of Imperial College London.

Declaration of Originality

I hereby confirm that, to the best of my knowledge, this thesis is my own work, and, except indicated appropriately by references, describes my own research. This thesis has not been submitted for any other purposes.

Pingchuan Ma

Copyright Declaration

Abstract

Decades of research in acoustic speech recognition have led to systems that we use in our everyday life. However, even the most advanced speech recognition systems fail in the presence of noise. The degraded performance can be compensated by introducing visual speech information. However, Visual Speech Recognition (VSR) in naturalistic conditions is very challenging, in part due to the lack of architectures and annotations.

This thesis contributes towards the problem of Audio-Visual Speech Recognition (AVSR) from different aspects. Firstly, we develop AVSR models for isolated words. In contrast to previous state-of-the-art methods that consists of a two-step approach, feature extraction and recognition, we present an End-to-End (E2E) approach inside a deep neural network, and this has led to a significant improvement in audio-only, visual-only and audio-visual experiments. We further replace Bi-directional Gated Recurrent Unit (BGRU) with Temporal Convolutional Networks (TCN) to greatly simplify the training procedure.

Secondly, we extend our AVSR model for continuous speech by presenting a hybrid Connectionist Temporal Classification (CTC)/Attention model, that can be trained in an end-to-end manner. We then propose the addition of prediction-based auxiliary tasks to a VSR model and highlight the importance of hyper-parameter optimisation and appropriate data augmentations.

Next, we present a self-supervised framework, Learning visual speech Representations from Audio via self-supervision (LiRA). Specifically, we train a ResNet+Conformer model to predict acoustic features from unlabelled visual speech, and find that this pre-trained model can be leveraged towards word-level and sentence-level lip-reading.

We also investigate the Lombard effect influence in an end-to-end AVSR system, which is the first work using end-to-end deep architectures and presents results on unseen speakers. We show that even if a relatively small amount of Lombard speech is added to the training set then the performance in a real scenario, where noisy Lombard speech is present, can be significantly improved.

Lastly, we propose a detection method against adversarial examples in an AVSR system, where the strong correlation between audio and visual streams is leveraged. The synchronisation confidence score is leveraged as a proxy for audio-visual correlation and based on it, we can detect adversarial attacks. We apply recent adversarial attacks on two AVSR models and the experimental results demonstrate that the proposed approach is an effective way for detecting such attacks.

Acknowledgements

It has been graceful as a winding stream, passionate as fire to live and study in London. I would like to express appreciation to everyone who has been with me and offered help to me during this unforgettable journey.

First and foremost, I would like to thank my supervisors, Prof. Maja Pantic and Dr. Stavros Petridis for their unceasing support. Their guidance and encouragement supports me through all my Ph.D. research. It is an honour, a pleasure and a fortune for me to work with Prof. Maja Pantic and Dr. Stavros Petridis.

During my PhD study, I would like to thank my collaborators, Yujiang Wang, Themos Stafylakis, Konstantinos Vougioukas, Rodrigo Mira, Alex Haliassos, Brais Martinez, Shiyang Cheng, Abhinav Shukla, Jie Shen, Georgios Tzimiropoulos, Björn W. Schuller for their invaluable contributions.

I am also thankful to my colleagues, Jiankang Deng, Jing Yang, Jiuxi Meng, Jie Pu, Yiming Lin, Marija Jegorova, Alexander Unsworth-Tomlinson, Christian Fuegen and Roshan Sumbaly, Markos Georgopoulos, Akis Kefalas, Zukang Liao for the constructive discussions. Special thanks to Bowen Li, Jingyi Wan, Ye Zhou, Na Zhou, Zhuofan Zhang, Qishao Wang, Mengyin Liu for their unfailing support and assistance.

I would like to thank my undergraduate supervisors, Fei Tao and Yuanjun Laili, who supervised my first research project and encouraged me to pursue a Ph.D.

I would like to thank my committee members Florian Metze and Yiannis Demiris, who provided very constructive feedback to strength this dissertation.

Finally, I would like to express my deepest gratitude to my dearest parents for their unconditional love and support.

Contents

1 Introduction **19**

1.1 Motivation 19

1.2 Contributions 21

1.2.1 Speech Recognition for Isolated Words 21

1.2.2 Speech Recognition for Continuous Speech 22

1.2.3 Learning Visual Speech Representations from Audio through Self-Supervision . 23

1.2.4 Investigating the Lombard Effect Influence on Audio-Visual Speech Recognition 23

1.2.5 Detecting Adversarial Attacks on Audio-Visual Speech Recognition 24

1.3 Publications 24

1.3.1 Published Works 24

1.3.2 Works under review 27

2 Background **28**

2.1 Datasets 28

2.2 Feature Extraction 31

2.2.1 Mel-Frequency Cepstral Coefficients 31

2.2.2 Active Appearance Models 33

2.3 Modeling Strategies 34

2.3.1 Connectionist Temporal Classification 35

2.3.2 Attention-based Encoder-Decoder Model 35

2.3.3 Recurrent Neural Network Transducer 36

2.4 Neural Networks 37

2.4.1 Recurrent Neural Networks 37

2.4.2 Temporal Convolutional Networks 39

2.4.3 Transformer 40

2.5 Summary 43

3 Audio-Visual Speech Recognition for Isolated Words **44**

3.1 Methodology 46

3.1.1 BGRU based Temporal Model 46

3.1.2 MS-TCN based Temporal Model 47

3.1.3 DC-TCN based Temporal Model 48

3.1.4 DS-TCN based Temporal Model 49

3.1.5 Data Augmentation in AVSR 51

3.1.6 Knowledge Distillation 52

3.2 Experiments 53

3.2.1 Preprocessing 53

3.2.2 Training Details 53

3.2.3 Initialisation 54

3.3 Results 54

3.3.1 Ablation Study 54

3.3.2 Self-Distillation 55

3.3.3 Audio-Visual Experiments 56

3.3.4 Efficient Training 57

3.3.5 Efficient Models 58

3.3.6 Error Analysis 59

3.4 Conclusion 61

4 Audio-Visual Speech Recognition for Continuous Speech 62

4.1 Methodology 64

4.1.1 Our Approach 64

4.1.2 Architecture 65

4.1.3 Prediction-based Auxiliary Tasks 68

4.1.4 Loss Functions 70

4.1.5 Using Additional Training Data 70

4.1.6 Curriculum Learning 71

4.1.7 Time Masking 72

4.1.8 Audio-Visual Fusion 74

4.2 Experimental Setup 74

4.2.1 Performance Metrics 74

4.2.2 Preprocessing . . . 75
4.2.3 Hyper-Parameter Optimisation . . . 75
4.2.4 Improving Language Models . . . 77
4.2.5 Language Models . . . 77
4.2.6 Implementation . . . 78
4.2.7 Baseline VSR Model . . . 78
4.2.8 Baseline ASR Model . . . 78
4.3 Results . . . 78
4.3.1 Results on LRS2 . . . 79
4.3.2 Results on LRS3 . . . 79
4.3.3 Results on CMLR . . . 81
4.3.4 Results on Spanish . . . 82
4.3.5 Results on Italian . . . 82
4.3.6 Results on Portuguese . . . 83
4.3.7 Results on French . . . 84
4.3.8 Comparison between Mean and Best WER/CER . . . 85
4.3.9 Audio-Visual Experiments . . . 85
4.4 Conclusion . . . 87

5 Learning Visual Speech Representations from Audio 88

5.1 Methodology . . . 90
5.1.1 Pretext task . . . 90

5.1.2 Downstream Task 91
5.2 Experimental Setup 91
5.2.1 Training Settings in the Pretext Task 91
5.2.2 Training Settings in Downstream Tasks 93
5.3 Results 94
5.3.1 VSR for Isolated Words 94
5.3.2 VSR for Continuous Speech 96
5.4 Conclusion 97

6 Investigating the Lombard Effect Influence on Audio-Visual Speech Recognition 98

6.1 Methodology 100
6.1.1 Network Architecture 100
6.2 Experimental Setup 101
6.2.1 Preprocessing 101
6.2.2 Data Augmentation 102
6.2.3 Training Settings 102
6.3 Results 103
6.3.1 Multi-Speaker Experiments 103
6.3.2 Subject-Independent Experiments 104
6.4 Conclusion 106

7 Detecting Adversarial Attacks on Audio-Visual Speech Recognition 108

7.1 Methodology 110

7.1.1 Attacks . . . 110
7.1.2 Audio-Visual Speech Recognition Threat Model . . . 111
7.1.3 Synchronisation-based Detection Method . . . 111
7.2 Experimental Setup . . . 112
7.2.1 Attacks . . . 112
7.2.2 Evaluation Metrics . . . 113
7.3 Results . . . 113
7.3.1 AVSR for Isolated Words . . . 113
7.3.2 AVSR for Continuous Speech . . . 114
7.4 Conclusion . . . 116

8 Conclusion **118**

8.1 Summary . . . 118
8.2 Applications . . . 120
8.3 Challenges . . . 121
8.4 Ethical Considerations . . . 122
8.5 Future Work . . . 124

Bibliography **125**

List of Tables

2.1 Details of Audio-Visual Datasets used in this thesis. CMS and MTS denote the Spanish parts of the CMU-MOSEAS and Multilingual TEDx datasets, respectively. 31

3.1 Performance of self-distillation models (Teacher = ResNet-18 + DC-TCN). The best-performing models from Table 3.3 are serving as teachers in first row. For each student model, the model from the line above is used as its teacher, and "Student *i*" stands for the model after the *i*-th self-distillation iteration. 54

3.2 Comparison with state-of-the-art methods on the LRW dataset in terms of classification accuracy. Experiments are divided into two groups, with and without utilising word boundaries indicators, respectively. "S.D.": self-distillation. "Scratch", "LiRA(LRS3)" and "LRS2&3+AVS" correspond to the three pre-training strategies in Table 3.3. 55

3.3 Ablation studies of three temporal models on LRW dataset. Starting from the best-performing DC-TCN model, we remove each data augmentation and the word boundaries indicators to examine their effectiveness. Then we replace the DC-TCN with MS-TCN and BGRU. "Scratch" denotes a model trained from scratch without using external data. "LiRA(LRS3)" indicates a self-supervised pre-trained model using LiRA [1] on the LRS3 dataset, and "LRS2&3+AVS" indicates a fully supervised pre-trained model on LRS2, LRS3 and AVSpeech. 56

3.4 Performance of different efficient models, ordered in descending computational complexity, and their comparison to the state-of-the-art on the LRW dataset. We use a sequence of 29-frames with a size of 88 by 88 pixels to compute the multiply-add operations (FLOPs). The number of channels is scaled for different capacities, marked as 0.5×, 1×, and 2×. Channel widths are the standard ones for ShuffleNet V2, while base channel width for TCN is 256 channels. 58

3.5 The top-1 accuracy of different methods on LRW where N frames are randomly removed from each testing sequence. 60

4.1 The architecture of the front-end encoder of the VSR model. The filter shapes are denoted by {Temporal Size × Spatial Size^2, Channels} and {Spatial Size^2, Channels} for 3D convolutional and 2D convolutional Layers , respectively. The sizes correspond to [Batch Size, Channels, Sequence Length, Height, Width] and [Batch Size × Sequence Length, Channels, Height, Width], for 3D and 2D convolutional layers, respectively. T_v denotes the number of input frames. 65

4.2 The architecture of the front-end encoder of the ASR model. The filter shapes are denoted by {Temporal Size, Channels} for 1D Convolutional Layers, respectively. The sizes correspond to [Batch Size, Channels, Sequence Length]. T_a denotes the length of audio waveforms. 66

4.3 Results of curriculum learning experiments on the LRS2 dataset. 72

4.4 Results of curriculum learning experiments on the LRS3 dataset. 73

4.5 Ablation study on the LRS2 dataset and LRS3 dataset. Models are trained on LRW+LRS2 and LRW+LRS3, respectively. 73

4.6 Investigation of the impact of hyperparameters and Language Model (LM) choices on the LRS2 dataset and LRS3 dataset. 75

4.7 Investigation of the impact of hyperparameters and Language Model (LM) choices on the validation set of LRS2 dataset. 76

4.8 Results on the LRS2 dataset. 80

4.9 Results on the LRS3 dataset. 80

4.10 Results on the CMLR dataset. 81

4.11 Results on the Multilingual TEDx-Spanish (MT_{es}) dataset. 82

4.12 Results on the CMU-MOSEAS-Spanish (CM_{es}) dataset. 82

4.13 Results on the Multilingual TEDx-Italian (MT_{it}) dataset. 83

4.14 Results on the Multilingual TEDx-Portuguese (MT_{pt}) dataset. 83

4.15 Results on the CMU-MOSEAS-Portuguese (CM_{pt}) dataset. 83

4.16 Results on the Multilingual TEDx-French (MT_{fr}) dataset. 84

4.17 Results on the CMU-MOSEAS-French (CM_{fr}) dataset. 84

4.18 Word Error Rate (WER) of the audio-only and audio-visual models on LRS2. 86

1 A comparison of the performance between the baseline methods and ours (pre-trained on LRS3) on the LRW dataset. 93

2 A comparison of the Word Error Rate (WER) between the baseline methods and ours (pre-trained on LRS3) on the LRS2 dataset. CL: Curriculum learning. 96

6.1 Video-only results on a multi-speaker scenario. L: Lombard, NL: non-Lombard. X-Y indicates a model trained on X (L or NL) speech and tested on Y (L or NL) speech. . . . 103

6.2 Video-only results on subject-independent experiments. L: Lombard, NL: non-Lombard. X-Y indicates a model trained on X (L or NL) speech and tested on Y (L or NL) speech. 104

7.1 Results for the proposed adversarial attack detection approach on word recognition models trained on the LRW dataset.L_∞^V is 1, 2 and 4 pixels when ϵ^V is 4, 8 and 16, respectively. . 114

7.2 Average results over 10 utterances of the proposed audio-visual synchronisation detection on partially targeted adversarial attacks on continuous speech recognition models trained on GRID. The success rate is the proportion of adversarial examples with WER less than 50%. ($\epsilon^A \in \{256, 512, 1024\}$, $\epsilon^V \in \{4, 8\}$) 115

7.3 Average results over 10 utterances of the proposed audio-visual synchronisation detection on fully targeted adversarial attacks on continuous speech recognition models trained on GRID. The success rate is the proportion of adversarial examples with WER = 0%. ($\epsilon^A \in \{256, 512, 1024\}$, $\epsilon^V \in \{4, 8\}$) 115

7.4 Results of the proposed audio-visual synchronisation detection on partially targeted adversarial attacks on continuous speech recognition models trained on GRID. The WER between transcribed and target phrases is up to 50%. The success rate is the proportion of adversarial examples with WER less than 50%. ($\epsilon^A = 512$, $\epsilon^V = 4$) 115

List of Figures

2.1 Original images from videos in LRS2. 29

2.2 The diagram from raw audio waveforms to MFCCs 32

2.3 An illustration of three sequence-to-sequence models: (a) CTC architecture. (b) RNN-Transducer architecture. (c) Attention-based encoder-decoder architecture. 34

2.4 (a): Dilated Temporal Convolution (Rate = 1), (b): Dilated Temporal Convolution (Rate = 2). (c): Dilated Temporal Convolution (Rate = 3). 40

2.5 An illustration of a stack of causal temporal convolution layers with the convolution filter size of 2. 40

2.6 Illustration of a N multi-head attention blocks in the encoder of a transformer. 41

3.1 (a): BGRU based audio-only model; (b): BGRU based visual-only model; (c): BGRU based audio-visual model. 46

3.2 (a) Temporal Convolutional Network (TCN). (b) Our Multi-scale TCN, which is used in the lip-reading model. 49

3.3 The architectures of the fully dense block (Up) and the partially dense block (bottom) in DC-TCN. We have selected the block filter sizes set $K = \{3, 5\}$ and the dilation rates set $D = \{1, 4\}$ for simplicity. In both blocks, Squeeze-and-Excitation (SE) attention is attached after each input tensor. A reduce layer is involved for channel reduction. 50

3.4 (a): Base architecture with ResNet18 and multi-scale TCN, (b): Lipreading model with ShuffleNet v2 backbone and multi-scale TCN back-end. (c): Lipreading model with ShuffleNet v2 backbone and depthwise separable TCN back-end. 51

3.5 The pipeline of knowledge distillation in generations 52

3.6 Performance for audio-only (A), video-only (V) and audio-visual (AV) models under different babble noise levels. The baseline corresponds to the model presented in [2]. . . 57

3.7 A comparison of our method and two baseline methods (End-to-End AVR [2] and Multi-Scale TCN [3]) on the five difficulty groups of the LRW test set. 60

4.1 Performance of visual speech recognition on LRS2 test set based on features extracted from different layers. “ce-b0” to “ce-b12” refer to the layers from each conformer block from bottom to top. 71

4.2 End-to-end audio-visual speech recognition architecture. The inputs are pixels and raw audio waveforms. 74

4.3 Performance of visual speech recognition on LRS2 validation set based on features extracted from different layers. “ce-b0” to “ce-b12” refer to the layers from each conformer block from bottom to top. 76

4.4 (a) Baseline ASR model, (b) Baseline VSR model, (c) Proposed model with prediction-based auxilliary tasks. The frame rate of extracted visual features and audio features is 25. (d) Architecture of ASR encoder. (e) Architecture of VSR encoder. 79

4.5 Word Error Rate (WER) as a function of the noise level. A: End-to-End audio model. V: End-to-End visual model, AV: End-to-End audio-visual model. log-Mel filter-bank: A conformer model trained with log-Mel filter-bank features. 85

5.1 The high-level architecture of our model and our methodology for audio-visual self-supervised training. 90

5.2 The variations of the end-to-end lip-reading architecture. The sub-figures in the top row ((a),(b),(c)) refer to the word-level lip-reading training procedures, while the sub-figures in the bottom row ((d),(e),(f)) refer to sentence-level lip-reading. From left to right, (a) and (d) denote training from scratch (the whole model is initialised randomly); (b) and (e) are feature extraction experiments based on visual features extracted from the pre-trained model; and (c) and (f) are fine-tuning experiments. Blue coloured blocks are trained from scratch on the downstream task; yellow coloured blocks are loaded from the pre-trained model and kept frozen during the downstream task; and green coloured blocks are loaded from the pre-trained model and are then fine-tuned for the downstream task. We abbreviate the following model layers: TM: Transformer, FC: Fully-Connected layer, MS-TCN: Multi-Scale Temporal Convolutional Network. 92

5.3 Accuracy of feature classification (LiRA-Frozen) on LRW based on features extracted from different layers after pre-training on LRS3 via self-supervision. “res-b3” and “res-b4” refer to the output of blocks 3 and 4 from the ResNet-18 respectively; and “ce-b2” to “ce-b12” refer to the layers from every two conformer blocks from bottom to top. 93

5.4 Effect of the size of training data on downstream task performance. (a): Accuracy of the end-to-end model as a function of the percentage of the training set (on a logarithmic scale) used for training on LRW. (b) WER achieved by the end-to-end model as a function of the percentage of labelled data used for training on LRS2. All LiRA-Frozen and LiRA-FineTuned models are pre-trained on LRS3 via self-supervision. LiRA-Frozen models are trained using features extracted from the last layer of the ResNet-18 in the pre-trained model, since it achieves the best performance as demonstrated in Figure 5.3. “CL” refers to the model being trained using curriculum learning. LRW and LRS2 contain 165 and 222 hours of labelled training data respectively. 95

6.1 End-to-end AVSR architecture overview. Raw images and audio waveforms are fed to the visual and audio streams, respectively, which produce features at the same frame rate at the bottleneck layer. These features are fused together and fed into another 2-layer BGRU to model the temporal dynamics. CTC [4] is used as the loss function. 101

6.2 WER of the end-to-end models as a function of the noise level in a multi-speaker scenario. A: audio-only model, AV: audio-visual model, L: Lombard, NL: non-Lombard, CL: 'compensated' Lombard. X-Y indicates a model trained on X (L or NL) speech and tested on Y (L or NL or CL) speech. Best seen in colour. 103

6.3 WER of the end-to-end as a function of the noise level in a subject-independent scenario. A: audio-only model, AV: audio-visual model, L: Lombard, NL: non-Lombard, CL: 'compensated' Lombard. X-Y indicates a model trained on X (L or NL) speech and tested on Y (L or NL or CL) speech. Best seen in colour. 105

6.4 WER of the end-to-end audio-visual model as a function of the noise level in a subject-independent scenario. L: Lombard, NL: non-Lombard. (NL,0.25L)-L indicates the performance is reported using a model trained on non-Lombard and 25% Lombard speech and tested on Lombard speech. The other combinations follow the same pattern. Best seen in colour. 106

7.1 An overview of our proposed detection method. (a) A video and an audio clip are fed to the end-to-end AVSR model. They are also fed to the synchronisation network (b) which estimates a synchronisation confidence score used for determining if the audio-visual model has been attacked or not (c). The confidence distribution of 300 adversarial and benign examples from the GRID dataset is shown in (d). 109

7.2 One example using basic iterative attack on the LRW dataset. Benign examples, adversarial noise examples, and adversarial examples are illustrated from top to bottom. (a) Raw images (ϵ^V=4, ϵ^V=8), (b) audio waveforms (ϵ^A=256, ϵ^A=512), and (c) audio log-spectrum (ϵ^A=256, ϵ^A=512) are presented from left to right. It is noted that the adversarial visual noise has been scaled with a ratio of 64 for a better illustration since the maximum distortion (ϵ^V=8) is 2 pixels. 112

Chapter 1

Introduction

Contents

1.1 Motivation **19**

1.2 Contributions **21**

1.3 Publications **24**

1.1 Motivation

AVSR is the task of transcribing text from audio and visual streams, which has recently attracted a lot of research attention due to its robustness against noise. Since the visual stream is not affected by the presence of noise, an audio-visual model can lead to improved performance over an audio-only model as the level of noise increases.

Traditional audiovisual fusion systems consist of two stages, feature extraction from the image and audio signals and combination of the features for joint classification [5, 6, 7]. Recently, several deep learning approaches for audiovisual fusion have been presented, which aim to replace the feature extraction stage with deep bottleneck architectures. Usually a transform, like Principal Component Analysis (PCA), is first applied to the mouth Region Of Interest (ROI) and spectrograms or concatenated Mel-Frequency Cepstral Coefficients (MFCCs) and a deep autoencoder is trained to extract bottleneck features [8, 9, 10, 11, 12, 13]. Then these features are fed to a classifier such as a support vector machine or a Hidden Markov Model.

In the early days, studies on AVSR are mostly based on heavily engineered approaches rather than learning.

In contrast to traditional approaches, deep-learning based approaches are advantageous to learn powerful representations, and their performance usually outperforms that of log-Mel filter-bank features.

However, few works have been presented very recently which follow an end-to-end approach for visual speech recognition. The main approaches followed can be divided into two groups. In the first one, fully connected layers are used to extract features and LSTM layers model the temporal dynamics of the sequence [14, 15]. In the second group, a 3D convolutional layer is used followed either by standard convolutional layers [16] or residual networks (ResNet) [17] combined with LSTMs or GRUs. End-to-end approaches have also been successfully used for speech emotion recognition using 1D CNNs and LSTMs [18].

To the best of our knowledge, work on end-to-end audiovisual speech recognition has been very limited. There are only two works which perform end-to-end training for audiovisual speech recognition [19, 14]. In the former, an attention mechanism is applied to both the mouth ROIs and MFCCs and the model is trained end-to-end. However, the system does not use the raw audio signal or spectrogram but relies on MFCC features. In the latter, fully connected layers together with LSTMs are used in order to extract features directly from raw images and spectrograms and perform classification on the OuluVS database [20].

Furthermore, another limitation of current models barring their use in practical applications is their computational cost. Many speech recognition applications rely on on-device computing, where the computational capacity is limited, and memory footprint and battery consumption play a crucial role in the deployment. As a consequence, a few works have also focused on the computational complexity of visual speech recognition [21, 22], but such models still trail massively behind full-fledged ones in terms of accuracy.

More importantly, the gap between the model augmented by artificial noise injection and background noise is not neglected. Normally, AVSR models have been presented [2, 19, 23, 24] by augmenting the performance of ASR models. The main application of such systems is in noisy acoustic environments since the main assumption is that the visual signal is not affected by noise and can therefore enhance the performance of speech recognition systems. However, this assumption is not true due to the Lombard effect. This mismatch have affected the performance of ASR, VSR and AVSR models in real scenarios.

This thesis studies AVSR in naturalistic conditions. Different challenges of AVSR are investigated and addressed by the proposal of various end-to-end deep models that lead to new state-of-the-arts in this field. Word-level AVSR, a task that aims to recognise isolated words from both audio waveforms and the sequence of images, is the first topic discussed in this thesis. An end-to-end trainable AVSR model, which takes as input raw audio waveforms and mouth ROIs, and predicts the characters of the spoken utterance is presented along with several temporal model variants. Subsequently, we present a novel E2E VSR model with auxiliary tasks for continuous speech. Different from previous works that were only conducted on English-based datasets, the proposed approach is evaluated not only in English but also in Mandarin and Spanish, which are the two most widely-spoken non-English languages. Additionally, we propose a framework which learns visual speech recognition from audio through self-supervision (LiRA), and we also present how the cues of audio-visual correlation can be leveraged to detect adversarial examples. Last but not least, we investigate the impact of Lombard effect in a system for AVSR.

1.2 Contributions

1.2.1 Speech Recognition for Isolated Words

The first contribution of this thesis is the study of AVSR for isolated words. Most previous works [25, 26, 27, 28, 29, 14] focus only on a simplistic setting, namely that of predicting several digits or short phrases in a controlled, laboratory condition, while very few works [17, 19, 30] recognise isolated words in the wild.

We solve the AVSR problem by introducing novel neural networks and data augmentation techniques. In the former case, we propose CNN-based networks including MS-TCN and DC-TCN that effectively improve the performance on the largest publicly available audio-visual speech datasets. In the latter case, we introduce the variable length augmentation technique to enhance the robustness towards frame removal. In addition, we apply simple and effective augmentation techniques such as time masking to achieve a high level of accuracy on the datasets.

Our results demonstrate that: 1) Compared with BGRUs-based models that require multiple stage training strategies [31], our methods reduce the training time from 3 weeks to 1 week. 2) We achieve a new state-of-the-art performance on the LRW dataset by combining all the latest data augmentation methods,

using the recently proposed DC-TCN, word boundary indicators and self-distillation. We achieve an accuracy of 92.8% for a single model and 93.4% for an ensemble. The performance can be slightly improved to 93% and 93.6%, respectively, by pre-training in a self-supervised manner on the LRS3 dataset. 3) Time masking is the most effective augmentation method followed by mixup. The use of DC-TCN significantly outperforms the MS-TCN which in turn outperforms the BGRU model. 4) The error analysis suggest that all these methods improve the performance by significantly increasing the classification accuracy of difficult words.

1.2.2 Speech Recognition for Continuous Speech

We further study the task of AVSR for continuous speech. Recent advances in deep learning and the availability of large audio-visual datasets have led to the development of much more accurate and robust speech recognition models than ever before [19, 32, 33, 34]. However, this constant improvement usually relies on creating larger training sets and less emphasis is put on the model design.

In Chapter 4, we extend our previous AVSR model [24] to an end-to-end model, which extracts features directly from image sequences and audio waveforms, and implement several changes that significantly improve the performance. Specifically, the changes include initialisation methods, language models and network architectures. Furthermore, we perform a comparison between audio-only models trained with log-Mel filter-bank features and raw waveforms. Although in clean conditions they both perform similarly, the raw audio model performs better in noisy conditions. A similar observation is made when comparing between audio-visual and audio-only models. Furthermore, we demonstrate that focusing on designing better models is equally important to using larger training sets. We propose the addition of prediction-based auxiliary tasks to a VSR model and highlight the importance of hyper-parameter optimisation and appropriate data augmentations. We test our approach on several challenging datasets in multiple languages and show that it outperforms all previous methods trained on publicly available datasets by a large margin. It even outperforms existing works trained on non-publicly available datasets which contain up to 20 times more data. We also show that using additional training data, even in other languages or with automatically generated transcriptions, results in further improvement which is in line with the recent trend in the literature.

1.2.3 Learning Visual Speech Representations from Audio through Self-Supervision

In Chapter 5, we leverage the vast amount of available audio-visual speech data to learn generic visual speech features and improve state-of-the-art lip-reading models by predicting audio features from visual speech. Previous work on self-supervised learning has has received substantial attention in recent years within the computer vision community. However, comparatively little attention has been given to leveraging one modality as a training objective to learn from the other.

In this chapter, we propose to learn visual speech representations from audio modalities. We demonstrate that LiRA provides a good initialisation for fine-tuning lip-reading models which consistently outperforms training from scratch, and that this method is particularly beneficial for smaller labelled datasets. We show that LiRA outperforms previous self-supervised methods for word-level lip-reading, achieving an accuracy of 88.1% on LRW by pre-training on unlabelled data. Finally, we leverage our approach towards sentence-level lip-reading, and find that our fine-tuned model achieves state-of-the-art performance on LRS2.

1.2.4 Investigating the Lombard Effect Influence on Audio-Visual Speech Recognition

The third contribution of this thesis is to study the Lombard effect in AVSR models. The Lombard effect [35] is the involuntary tendency of speakers to make speech more intelligible and affects both the acoustic characteristics of speech and lip movements in a noisy environment. It is acoustically characterised by an increase in the sound intensity, fundamental frequency, vowel duration and a shift in the formant frequencies [36, 37, 38, 39]. Visually, it is characterised by hyper-articulation [40, 41] and more pronounced rigid-head motion [39, 42]. Several AVSR models have been recently proposed which aim to improve the robustness over audio-only models in the presence of noise. However, almost all of them ignore the impact of the Lombard effect, i.e., the change in speaking style in noisy environments which aims to make speech more intelligible and affects both the acoustic characteristics of speech and the lip movements.

In Chapter 6, we investigate the Lombard effect influence on E2E audio-visual speech recognition. To the best of our knowledge, this is the first work which does so using end-to-end deep architectures and presents results on unseen speakers. Our results show that properly modelling Lombard speech is always

beneficial. Even if a relatively small amount of Lombard speech is added to the training set then the performance in a real scenario, i.e, noisy Lombard speech, can be significantly improved. We also show that standard approach followed in the literature, where a model is trained and tested on noisy plain speech, provides a correct estimate of the video-only performance but overestimates the performance of audio-only models in a multi-speaker scenario. In a subject-independent scenario the performance is overestimated for SNRs higher than -3dB and underestimated for lower SNRs.

1.2.5 Detecting Adversarial Attacks on Audio-Visual Speech Recognition

The last contribution of this thesis is to detect adversarial examples in an audio-visual model by leveraging the correlation between audio and visual streams. Existing studies have mainly focused on crafting adversarial examples. [43, 44, 45, 46]. However, work on how to detect adversarial attacks is very limited. To the best of our knowledge, the only work in the audio domain was proposed by Yang et al. [47] and exploits the inherent temporal dependency in audio samples to detect adversarial examples.

In this work, we propose an efficient and straightforward detection method based on the temporal correlation between audio and video streams. The main idea is that the correlation between audio and video in adversarial examples will be lower than benign examples due to added adversarial noise. We use the synchronisation confidence score as a proxy for audio-visual correlation and based on it we can detect adversarial attacks. To the best of our knowledge, this is the first work on detection of adversarial attacks on AVSR models. We apply recent adversarial attacks on two AVSR models trained on the GRID and LRW datasets. The experimental results demonstrate that the proposed approach is an effective way for detecting such attacks.

1.3 Publications

This section presents the publications that were completed during the period of this Ph.D. course.

1.3.1 Published Works

- **Pingchuan Ma**[†], Yujiang Wang[†], Jie Shen, Stavros Petridis, and Maja Pantic, “Training Strategies For Improved Lip-reading”. Submitted to *Proceedings of the 47th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8472-8476, 2022.

- Rodrigo Mira, Konstantinos Vougioukas, **Pingchuan Ma**, Stavros Petridis, Björn W. Schuller, Maja Pantic, "End-to-End Video-To-Speech Synthesis using Generative Adversarial Networks", IEEE Transactions on Cybernetics, pp. 1–13, 2022.

- Enrico Varano, Konstantinos Vougioukas, **Pingchuan Ma**, Stavros Petridis, Maja Pantic, Tobias Reichenbach, "Speech-driven Facial Animations Improve Speech-in-noise Comprehension of Humans". *Frontiers in Neuroscience*, 2022.

- **Pingchuan Ma***, Rodrigo Mira*, Stavros Petridis, Björn W. Schuller, Maja Pantic, "LIRA: Learning Visual Speech Representations from Audio through Self-supervision". In *Proceedings of the 22th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 3011-3015, 2021.

- **Pingchuan Ma**, Stavros Petridis, Maja Pantic, "Detecting Adversarial Attacks on Audiovisual Speech Recognition". In *Proceedings of the 46th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6403-6407, 2021.

- **Pingchuan Ma**, Stavros Petridis, Maja Pantic, "End-to-end Audio-visual Speech Recognition with Conformers". In *Proceedings of the 46th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7613-7617, 2021.

- **Pingchuan Ma***, Brais Martinez*, Stavros Petridis, Maja Pantic, "Towards Practical Lipreading With Distilled and Efficient Models". In *Proceedings of the 46th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7608-7612, 2021.

- **Pingchuan Ma***, Yujiang Wang*, Jie Shen, Stavros Petridis, and Maja Pantic, "Lip-reading with Densely Connected Temporal Convolutional Networks". In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2857-2866, 2021.

- Rodrigo Mira, Konstantinos Vougioukas, **Pingchuan Ma**, Stavros Petridis, "End-To-End Video-To-Speech Synthesis using Generative Adversarial Networks with Multiple Critics". In *Proceedings of the 34th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Sight and Sound workshop*, 2021.

- Shiyang Cheng*, **Pingchuan Ma***, Georgios Tzimiropoulos, Stavros Petridis, Adrian Bulat, Jie

Shen, Maja Pantic, "Towards Pose-invariant Lip-Reading". In *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4357-4361, 2020.

- Brais Martinez, **Pingchuan Ma**, Stavros Petridis and Maja Pantic, "Lipreading using Temporal Convolutional Networks". In *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6319-6323, 2020.

- Abhinav Shukla, Konstantinos Vougioukas, **Pingchuan Ma**, Stavros Petridis and Maja Pantic, "Visually Guided Self-supervised Learning of Speech Representations". In *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6299-6303, 2020.

- Stavros Petridis, Yujiang Wang, **Pingchuan Ma**, and Maja Pantic, "End-to-End Visual Speech Recognition for Small-Scale Datasets". *Pattern Recognition Letters*, pp. 421-427, 2020.

- **Pingchuan Ma**, Stavros Petridis, and Maja Pantic, "Investigating the Lombard Effect Influence on End-to-End Audio-Visual Speech Recognition". In *Proceedings of the 20th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 4090-4094, 2019.

- Konstantinos Vougioukas, **Pingchuan Ma**, Stavros Petridis and Maja Pantic, "Video-Driven Speech Reconstruction using Generative Adversarial Networks". In *Proceedings of the 20th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 4125-4129, 2019.

- Stavros Petridis, Themos Stafylakis, **Pingchuan Ma**, Georgios Tzimiropoulos, and Maja Pantic, "Audio-Visual Speech Recognition with a Hybrid CTC/Attention Architecture". In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT)*, pp. 513-520, 2018.

- Stavros Petridis, Themos Stafylakis, **Pingchuan Ma**, Feipeng Cai, Georgios Tzimiropoulos, and Maja Pantic, "End-to-End Audiovisual Speech Recognition". In *Proceedings of the 43rd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6548-6552, 2018.

1.3.2 Works under review

- **Pingchuan Ma**, Stavros Petridis, Maja Pantic, "Visual Speech Recognition for Multiple Languages", Submitted to *Nature Machine Intelligence*, 2022.
- Triantafyllos Kefalas, Eftychia Fotiadou, Markos Georgopoulos, Yannis Panagakis, **Pingchuan Ma**, Stavros Petridis, Themos Stafylakis, and Maja Pantic, "KAN-AV Dataset for Audio-Visual Face and Speech Analysis in the Wild". Submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*. (under review)

Chapter 2

Background

Contents

2.1 Datasets **28**

2.2 Feature Extraction **31**

2.3 Modeling Strategies **34**

2.4 Neural Networks **37**

2.5 Summary **43**

This chapter briefly reviews prior work relevant to my thesis which will be discussed in Chapter 3 to 7. Firstly, we overview the in-the-wild datasets collected in naturalistic conditions for audio-visual speech in Section 2.1. Next, we describe feature extraction approaches relevant to this thesis in Section 2.2. Finally, three modelling strategies and three modelling components are introduced in Section 2.3 and 2.4, respectively, which are served to set the stage for understanding our proposed methods surrounding the topics of this thesis.

2.1 Datasets

AVSR has achieved significant progress in the performance due to the application of deep neural networks and the availability of large datasets. However, most of these datasets [25, 48, 49, 20] do not reflect the real scenarios on a daily basis. Particularly, they are subjected to controlled, laboratory conditions and focused on simplistic tasks, such as digit recognition. Instead, recent research focus on collecting large

Figure 2.1: Original images from videos in LRS2.

corpora of audio-visual speech data in naturalistic conditions. In this section, we review multiple publicly available audio-visual datasets, which cover various languages (English, Mandarin Chinese, Spanish), poses (frontal, profile) and recordings conditions (clean, noisy).

LRW [30] is a large-scale audio-visual dataset that contains 500 different words from over 1 000 speakers and was collected from BBC programs. Each utterance has 29 frames (1.16 seconds) and its boundaries are centered around the target word. The dataset is divided into training, validation and test sets. The training set contains at least 800 utterances for each class while the validation and the test sets contain 50 utterances each.

LRW-1000 [50] is a large-scale audio-visual mandarin dataset collected from Chinese national news programs. It contains a total of 718 018 samples for 1 000 mandarin words, recorded from more than 2 000 subjects. The average duration for each sequence is 0.3 second, and the total length of all sequences is about 57 hours. The videos are divided into a training set with 603 193 utterances, a validation set with 63 237 utterances and a test set with 51 588 utterances, respectively. This dataset is even more challenging than LRW considering its huge variations in speaker properties, background clutters, scale, etc.

LRS2 [51] is a large-scale audio-visual English dataset collected from BBC programs. It consists of 144 482 video clips with a total duration of 224.5 hours. The videos are divided into a pre-training set with 96 318 utterances (195 hours), a training set with 45 839 utterances (28 hours), a validation set with 1,082 utterances (0.6 hours) and a test set with 1 243 utterances (0.5 hours). Examples are presented in

Figure 2.1.

LRS3 [52] is the largest publicly audio-visual English dataset collected from TED and TEDx talks. It contains 438.9 hours with 151 819 utterances. Specifically, there are 118 516 utterances in the pre-train set (408 hours), 31 982 utterances in the train-val set (30 hours) and 1 321 utterances in the test set (0.9 hours).

CMLR [53] is a large-scale audio-visual Mandarin dataset collected from Chinese national news program. It contains 102 072 clips with transcriptions. The training, validation and test sets contain 71 448 (60.6 hours), 10 206 (8.6 hours) and 20 418 (17.3 hours) clips, respectively. To the best of our knowledge, CMLR is the largest publicly available dataset in Mandarin.

CMU-MOSEAS [54] is a large-scale dataset, which contains multiple languages, and was collected from YouTube videos. It consists of 40 000 transcribed sentences and includes Spanish, Portuguese, German and French. We only consider the Spanish videos with a total duration of 15.7 hours. We divided the data into training and test sets which contain 8 287 videos (15 hours) and 329 videos (0.7 hours), respectively.

Multilingual TEDx [55] is a multilingual corpus collected from TEDx Talks. It covers 8 languages with manual transcriptions and has a total duration of 765 hours. For the purposes of this study, we only consider the Spanish videos and use the data split proposed in [55]. We manually cleaned the dataset to exclude videos where the speaker is not visible, resulting in a total of 44 745 videos (73.0 hours) for training, 403 videos (0.7 hours) for validation and 302 videos (0.5 hours) for testing. It should be noted that we only use the training set in this study.

GRID [56] is an audio-visual dataset contains of 33 speakers and 33000 utterances (1000 per speaker). Each utterance is composed of six words taken from the combination of the following components: <command: 4><colour: 4><preposition: 4><letter: 25><digit: 10><adverb: 4>, where the number of choices for each component is indicated in the angle brackets. In this work, we follow the evaluation protocol from [57] where 16, 7 and 10 subjects are used for training, validation and testing, respectively.

Lombard GRID [38] is an audio-visual dataset that consists of 5400 utterances from 54 speakers (30 females and 24 males), with 100 utterances (50 Lombard and 50 plain) per speaker. Each utterance is composed of a six word sequence following the same pattern in GRID [56]. During speaking, both frontal and profile faces were simultaneously recorded at 25 frames per second (fps) and audio was recorded

Table 2.1: Details of Audio-Visual Datasets used in this thesis. CMS and MTS denote the Spanish parts of the CMU-MOSEAS and Multilingual TEDx datasets, respectively.

Dataset	Language	Transcriptions	Utterances	Hours
Publicly Available Datasets				
GRID [56]	English	✓	34 000	28
Lombard GRID [38]	English	✓	54 00	7
LRW [30]	English	✓	538 766	157
LRW-1000 [50]	Mandarin	✓	718 018	57
LRS2 [19]	English	✓	144 482	223
LRS3 [52]	English	✓	151 819	438
CMLR [53]	Mandarin	✓	102 112	61
MTS [55]	Spanish	✓	45 450	71
CMS [54]	Spanish	✓	8 616	16
AVSpeech [58]	English	✗	350 991	641
Non-Publicly Available Datasets				
MVLRS [19]	English	✓	500k	730
LSVSR [32]	English	✓	2 934 899	3 886
YT-31k [33]	English	✓	-	31 000
YT-90k [34]	English	✓	-	90 000
VoxCeleb2^{clean} [59]	English	✗	140k	334

at 48kHz and downsampled to 16kHz. Recordings for each utterance were collected in two conditions, Lombard (L) and Non-Lombard (NL). The non-Lombard condition was performed by reading sentences to a condenser microphone placed 30cm in front of the participants, in which the own-voice attenuation was compensated. The Lombard condition follows the same setting, but speech-shaped noise at 80dB sound pressure level (SPL) was presented to participants via headphones.

2.2 Feature Extraction

2.2.1 Mel-Frequency Cepstral Coefficients

MFCCs, which is commonly used as a low-dimensional set of features in speech recognition systems, was first proposed by Bridle and Brown [60] as the spectrum-shaped coefficients transformed through a

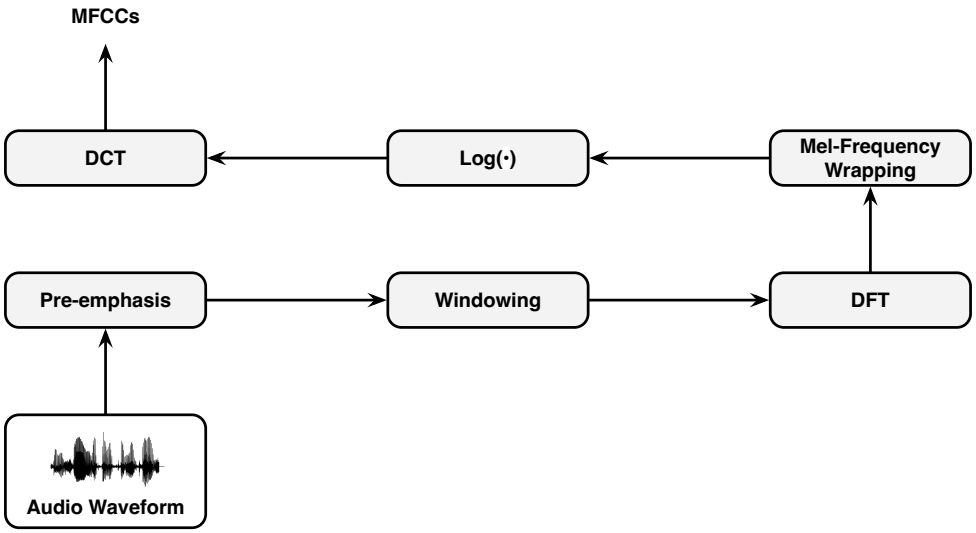


Figure 2.2: The diagram from raw audio waveforms to MFCCs

19-channel bandpass filters, and was further developed by Mermelstein [61, 62] as Mel-based cepstral parameters. MFCCs are derived from modeling the human auditory system. Specifically, the extraction of MFCCs from raw audio waveforms typically includes six steps as illustrated in Figure 2.2. Firstly, we apply pre-emphasis filter to the raw audio waveforms, which aims to compensate the higher frequency parts that was suppressed in raw audio waveforms as well as amplify the importance of high-frequency formants. The signal after pre-emphasis is sliced to a number of audio clips through windows. At this step, in order to keep the continuity of the speech at the ending points, smoothing window functions such as the Hamming window [63] or the Hanning window [64] is applied on a speech frame. Furthermore, DCT is applied to transform each windowed frame $x[n]$, from the time domain to the frequency domain, resulting in a frequency band that is linearly spaced. Since it is not consistent with human perception, in which human ear can detect relatively small changes in lower frequencies, an approximate expression [65] is used to map the powers of spectrum onto the Mel scale:

$$f_{\text{Mel}} = 2\,595 \times \log_{10}(1 + \frac{f}{700}) \tag{2.1}$$

where f and f_{Mel} denotes the physical frequency, and the perceived frequency, respectively. We further compute the energy of the Mel-filter bank vectors by taking the logarithm of the square magnitude; This is because the human reaction to the sound is logarithmic. At this point, the log-Mel spectrum is a feature representation, which is widely used in application of speech such as speaker recognition, phone detection.

It is noted that the energy levels in adjacent bands of the log-Mel spectrum are correlated. To suppress the spectrum, we apply Discrete Cosine Transformation (DCT) to the Mel frequency coefficients and obtain a set of cepstral coefficients, known as MFCCs.

2.2.2 Active Appearance Models

The visual speech information, is often complementary to audio-visual system, especially when the audio signal is degraded. As the lips are one of the most prominent features closely associated with visual speech, the shape of the contours of the lips is often considered. Active Appearance Models (AAMs) [66] are a type of statistical models to describe the shape and texture of objects using a set of detailed descriptive parameters. To model a sequence of head motion, AAMs take as input a sequence of images and a set of landmark points. The pixel values of the images represent visual textures. The landmark points provide an estimated location of face boundaries. In the following paragraphs, we describe a detailed procedure that estimates the parameters of AAMs. Specifically, an AMM is composed of three components, a Shape Model, a Motion Model and an Appearance Model.

A **Shape Model** is defined by the vertex locations of the mesh. The shape of $\mathbf{s}$ with v vertices is defined as $\mathbf{s} = (x_1, y_1, x_2, y_2, ..., x_v, y_v)^T$. The shape can be formalized as a linear combination of a base shape $\mathbf{s}_0$ plus n shape vectors $\mathbf{s}_i$.

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{n} p_i \mathbf{s}_i \tag{2.2}$$

where p_i are the shape parameters. In order to obtain the base mesh $\mathbf{s}_0$, a collection of training face shapes are normalised and PCA is further applied. The shape vectors $\mathbf{s}_i$ are computed by aligning every training face shape to the base mesh $\mathbf{s}_0$ using similarity transform.

An **Appearance Model** is an image $A(\mathbf{x})$ that the pixels $\mathbf{x}$ in the base mesh $\mathbf{s}_0$. In AAMs, the appearance $A(\mathbf{x})$ can be expressed as a linear combination of an appearance $A_0(\mathbf{x})$ plus n appearance $A_i(\mathbf{x})$:

$$A_i(\mathbf{x}) = A_0(\mathbf{x}) + \sum_{i=1}^{n} \lambda_i A_i(\mathbf{x}) \tag{2.3}$$

where λ_i denotes the appearance parameters. It is noted that the appearance A_i are orthonormal in AMMs. The appearance vectors A_i are computed by warping the input images to the base mesh using the piecewise

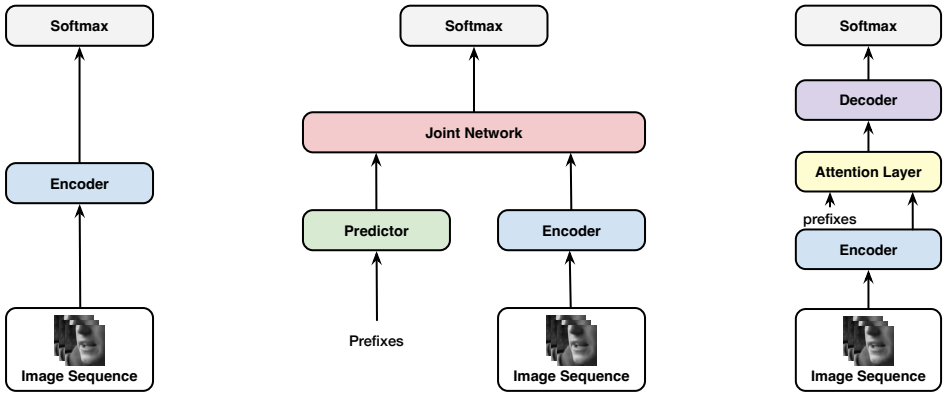


Figure 2.3: An illustration of three sequence-to-sequence models: (a) CTC architecture. (b) RNN-Transducer architecture. (c) Attention-based encoder-decoder architecture.

affine warps that is defined between the shape vector $\mathbf{s}$ and the base vector $\mathbf{s}_0$, and then PCA is applied onto the warpped appearance.

A **Motion Model** is a warping function that warps the texture related to a shape. The choices for the warping function include piece-wise affine and thin plate splines warps. Generally speaking, the motion model defines how the image should be warped into a canonical reference frame given a shape $\mathbf{s}$. It is noted that once the images have been wrapped to the reference frame, we assume all image have same dimensionality and share the same face shape.

2.3 Modeling Strategies

Sequence-to-sequence models aim to transform an input sequence to an output sequence with arbitrary length, which has been widely used in the transformation between texts [67, 68], images [69, 70, 51], and speech [71, 72, 73, 74, 75]. In this section, we briefly review the recent sequence-to-sequence modelling approaches for speech recognition including CTC, RNN-T, and attention-based encoder-decoder models. Before we introduce the modelling strategies in detail, we begin by introducing our notation. Specifically, we assume that the input sequence is parameterised to a feature vector $\mathbf{x} = (x_1, x_2, ..., x_T)$, where $x_t \in \mathbb{R}^{\mathrm{d}}$. The output sequence $\mathbf{y}$ is denoted as $\mathbf{y} = (y_1, y_2, ..., y_L)$, where T denotes the sequence length and L denotes the number of symbols. Their architectures are briefly illustrated in Figure 2.3.

2.3.1 Connectionist Temporal Classification

Connectionist Temporal Classification (CTC) [4], is a type of algorithms that estimates the conditional probability $p(\mathbf{y}|\mathbf{x})$ for an output sequence $\mathbf{y}$ given an input sequence $\mathbf{x}$. The CTC loss is used to transcribe directly between inputs and target outputs without any intermediate annotation. In particular, CTC sums over the probability of all possible alignments to obtain the posterior of the target sequence:

$$p(\mathbf{y}|\mathbf{x}) = \sum_{\tilde{y} \in \tilde{\mathbf{A}}_{\text{CTC}}(\mathbf{y})} \prod_{t=1}^{T} p_t(\tilde{y}_t|x_1, ..., x_t) \tag{2.4}$$

where $\tilde{\mathbf{A}}_{\text{CTC}}$ denotes the set of possible label sequences, which have a length of T including blank tokens and repeated characters. The CTC loss is computed by the negative log likelihood of the posterior probability. The CTC loss can be computed with a dynamic programming algorithm in a feedforward direction. The gradients required to train CTC models can be optimised using the forward-backward algorithm.

2.3.2 Attention-based Encoder-Decoder Model

Over the last several years, attention-based models [76, 72, 77] leads an improvements on sequence-to-sequence applications such as speech recognition. Unlike a traditional ASR system which independently train an acoustic model, a pronunciation model and a language model [78], attention-based encoder-decoder models such as the Listen, Attend and Spell (LAS) model [72], greatly simplify the training pipeline by jointly training these components in an end-to-end fashion and achieved comparable results to state-of-the-art ASR systems.

Encoder Network is to model high-level representations $\mathbf{h} = (h_1, h_2, ..., h_T)$ from input sequences. Neural networks such as LSTM and TCN, as introduced in Section 2.4, are generally serve as the encoder network. It is important to note here that, the design of uni-directional should be considered for some speech applications, which are constrained to be causal with no future context.

Decoder Network is responsible for interpreting the context vector from the encoder network. The decoder network is generally a stack of uni-directional RNNs. At each time step, the recurrent unit receives a hidden state from previous units and produce an output as well as its associated hidden states for further recurrent units. Typically, the encoder-decoder without attention is to compress a sequence of

input into a fixed-size vector. This vector is then fed to a recurrent neural network for decoding. Since there is an instability to extract strong contextual correlation in a fixed-size vector from long variable length sequences, the information in the encoder-decoder network is easily lost.

Attention Mechanism solves the limitation of instability in a basic encoder-decoder network. The attention mechanism determines how much emphasis should be placed on each part and provides a weighted context vector. An attention mechanism can be generalised to compute an alignment score of the value conditioned on the query and associated keys. In a more formal formulation,

$$c_i = \sum_{j=1}^{T} \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{T} \exp\left(e_{ik}\right)} h_j \tag{2.5}$$

where

$$e_{ij} = a(s_{i-1}, h_j)$$

is an alignment model that scores how well the features at position j and the output at position i are correlated. s_{i-1} denotes the hidden score from the previous hidden state at the position $i - 1$.

The alignment model a can be formulated in various ways. For example, Bahdanau *et al.* [79] presents an "additive attention" as the alignment model a such that

$$a(s_{i-1}, h_j) = v_a^\top \tanh\left(W_a s_{i-1} + U_a h_j\right),$$

where $W_a \in \mathbb{R}^{D \times D}$, $U_a \in \mathbb{R}^{D \times 2D}$ and $v_a \in \mathbb{R}^D$ are the weight matrices, and D denotes the dimension of the hidden space. Depending on how the alignments between output and input frames are designed, different types of attention mechanism can also be represented in a form of Dot-Product [80], Location-Base [80], Content-Base[81], Scaled Dot-Product [67], and etc..

2.3.3 Recurrent Neural Network Transducer

Recurrent Neural Network Transducer (RNN-T) [71, 82, 83, 84, 85], was proposed as an extension to the CTC-based modelling approach for sequence labeling tasks has become popular recently. Compared with conventional ASR, RNN-T model has an implicit language model and can directly predict tokens (characters or word-pieces) from audio features in an end-to-end manner. Furthermore, RNN-T has

the advantage of streamability over encoder-decoder based ASR models. Specifically, RNN-T allows the output to be decoded as soon as the first token has been encoded instead of waiting until the entire utterance is available.

RNN-T is built conditioned on the previous non-blank labels and acoustic embedding. In one specific formalism, we define the conditional distributions of RNN-T:

$$p(\mathbf{y}|\mathbf{x}) = \sum_{\tilde{y} \in \tilde{\mathbf{A}}_{\text{RNNT}}(\mathbf{y})} \prod_{i=1}^{T+\tilde{L}} p_t(\tilde{y}_i | x_1, ..., x_{t_{i-1}}, y_0, ..., y_{l_{i-1}}) \tag{2.6}$$

where $\tilde{\mathbf{A}}_{\text{RNNT}}$ denotes the set of possible label sequences, which have a length of T and $\tilde{L}$ labels including blank tokens and repeated characters. The conditional distributions can be parameterised by neural networks. The whole model can be optimised using the forward-backward algorithm by maximising Equation 2.6.

An RNN-T model typically consists of three parts: the encoder network, the prediction network and the joint network, as illustrated in Figure 2.3b. Specifically, *Encoder network* is analogous to the acoustic model, which converts the input to high-level representations. *Prediction network* produces the embedding conditioned on the previous non-blank output label. Note that the initial input of the prediction network is an all-zero tensor. The prediction network can be modelled by temporal modules with casual settings, such as LSTM, casual CNN or casual transformer network. *Joint network* is built using a stack of feed-forward layers. which takes as input both the embedding from the encoder and the output given the previous non-blank label index from the prediction network. At the top of the RNN-T, the output probability distribution is computed by a softmax layer.

2.4 Neural Networks

2.4.1 Recurrent Neural Networks

Recurrent Neural Network (RNN) is a class of neural networks that is capable of modelling temporal dependencies in time-dependent systems. It maintains a set of hidden unit activations, which is fed back into the network along with the inputs through time [86, 87]. The use of a RNN has been widely adopted in the study of sequential data, such as language modelling [88], machine translation [68],

speech recognition [89], and time series forecasting [90]. However, Basic RNN architectures suffer from the problem of vanishing gradients. Specifically, during the back-propagation, the gradient becomes smaller and smaller through layers and also through time [91, 92]. Long Short-Term Memory networks (LSTM) [93], as a specialised type of RNNs, are designed to solve the problem of gradient vanishing in vanilla RNNs. Specifically, in this architecture, an LSTM cell makes decision by considering the current input, previous output and previous hidden state.

Fully-Connected LSTM (FC-LSTM) is the vanilla version of LSTM where the input, cell output and states are all temporal vectors. A FC-LSTM unit is composed of a memory cell c_t, an output gate o_t, a forget gate f_t and an input gate i_t, respectively. The memory cell remembers the dependencies among different elements. The three gates decide which information is allowed in the memory cell. The forward pass of a LSTM unit is given by

$$\begin{bmatrix} \mathbf{f}_t \\ \mathbf{i}_t \\ \mathbf{o}_t \\ \tilde{\mathbf{c}}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} (\mathbf{W} \begin{bmatrix} \mathbf{h}_{t-1} \\ \mathbf{x}_t \end{bmatrix} + \mathbf{b}) \tag{2.7}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \tag{2.8}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \tag{2.9}$$

where $\tilde{\mathbf{c}}_t$ is a memory cell candidate for the current input, $\mathbf{h}_t$ is the hidden state of the LSTM unit, $\mathbf{W}$ and $\mathbf{b}$ denote the weight matrix and the bias, respectively. $\odot$ is element-wise multiplication.

Convolutional LSTM (ConvLSTM) [90] is a variant of LSTM but internal matrix multiplications are substituted by convolutional operations. Stemming from the poverty of the spatial correlation modelling in FC-LSTM, ConvLSTM are capable of modelling spatiotemporal sequence with the leverage of convolutional kernels with the use of convolutional architecture. Besides this, they preserve the advantages of FC-LSTMs which is advantageous to capture long and short-term temporal dependencies. Therefore, ConvLSTM are considered as an ideal candidate for spatialtemporal sequence problems such as action recognition and VSR.

2.4.2 Temporal Convolutional Networks

The concept of "convolution" was firstly proposed by LeCun *et al.* in [94] to recognise handwritten zip code recognition. Time-Delay Neural Network (TDNN), also known as one-dimensional convolutional network, where the convolution is performed in time domain, has been applied to phoneme recognition [95]. Yamaguchi *et al.* [96] combined pooling operation and TDNN to achieve a speaker-independent isolated word recognition system. A modified architecture of TDNN has gained tremendous popularity in [97], who used it as a component of acoustic model and proposed the operation of sub-sampling to reduce the computational cost. Furthermore, the design of dilation operation [98] newly introduced in temporal convolutional networks enlarges the reception field exponentially with linearly increasing number of parameters, which has shown remarkable success on sequence-to-sequence mapping problems such as speech generation and text-to-speech [99]. More recently, Bai *et al.* [100] described a simple yet effective TCN architecture which outperformed baseline RNNs, suggesting that TCN can be a reasonable alternative for RNNs on sequence modelling problems. Following this work, it was further demonstrated in [3] that a multi-scale TCN could achieve better performance than RNNs on lip-reading of isolated words, which is also the state-of-the-art model so far. Such multi-scale TCN stacks the outputs from convolutions with multiple kernel sizes to gain a more robust temporal features, which has already been shown to be effective in other computer vision tasks utilising multi-scale information such as the semantic segmentation [101, 102, 103].

Training TCNs on the raw time signal [104, 105, 106, 107] have been shown to match the recognition performance of classical feature extraction pipelines. Golik *et al.* [106] adopted one-dimensional convolutional layers that perform filtering in time and showed that the first convolutional layer learn a spectrograms that are non-linearly distributed in frequency. Sainath *et al.* [107] firstly show raw waveform and log-mel features match in performance based on the proposed Convolutional, Long Short-Term Memory Deep Neural Network (CLDNN). More recently, Parcollet *et al.* [105] empirically showed that an waveform-based speech recognition model outperform previously E2E systems relying on pre-computed acoustic features by a margin of 1.2% on the Wall Street Journal dataset.

Dilation Convolutional Network (DCN) is a class of convolutional networks that expands kernels by inserting holes between consecutive elements. An illustration of dilated temporal convolution is shown

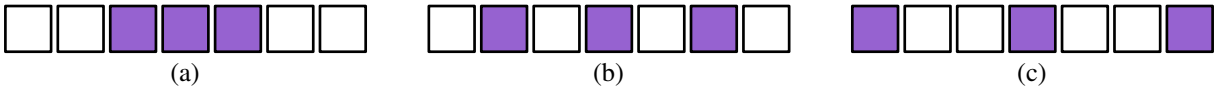


Figure 2.4: (a): Dilated Temporal Convolution (Rate = 1), (b): Dilated Temporal Convolution (Rate = 2). (c): Dilated Temporal Convolution (Rate = 3).

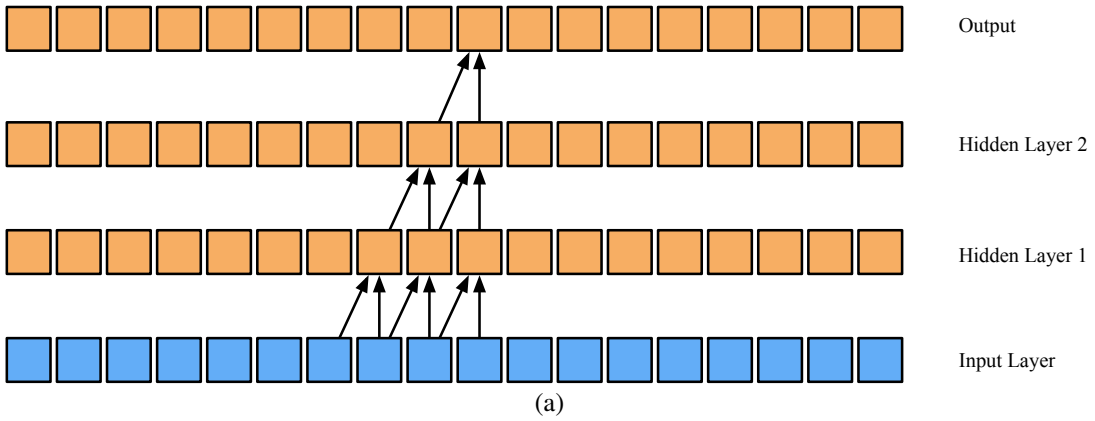


Figure 2.5: An illustration of a stack of causal temporal convolution layers with the convolution filter size of 2.

in Figure 2.4. The design of skip can enlarge the receptive view of the network exponentially. Instead of using vanilla convolution, dilation convolution has an advantage of capturing a global view of the input with fewer parameters. The design of dilation convolution [98] has been applied in the domain of computer vision [3], signal processing [108], and natural language processing [100].

Casual Convolution is a type of convolutional, in which the data ahead of the current position are not involved in calculation. An illustration of a stack of casual convolution is shown in Figure 2.4. By defining the convolution in a design of causality, the future time steps will not be affected when predicting the value of the next one. The casual design in temporal convolution can be achieved by padding operations. In particular, we pad the layer's input with zeros in the front so that we can also predict the values of early time steps in the frame.

2.4.3 Transformer

Transformer is an encoder-decoder architecture that utilises a self-attention mechanism to transform from a sequence of elements into another sequence. It leads the performance in several sequence modelling tasks, such as language modelling [109], machine translation [67]. More recently, transformer has been applied in speech processing [110, 111, 112]. It is typically in conjunction with convolutional neural

networks [111, 112]. In [111], Kong *et al.* proposed a CNN plus Transformer-based network for sound event detection, where a transformer is built on the top of a CNN, while in [112], Gulati *et al.* proposed convolution augmented transformer module which combines CNN and attention module in each block. More recently, Gong *et al.* [113] proposed a convolutional-free backbone and purely leverage attention mechanism to perform audio classification.

Transformer starts with a positional embedding module, followed by a stack of attention blocks. In each attention block, there are a self-attention module, a layer normalisation layer, a feed-forward module, and a second layer normalisation (LN) layer stacked in order. as illustrated in Figure 2.6. The decoder which is composed of an embedding module and a set of residual multi-head attention blocks, receives the embedding from the input and then performs decoding through a stack of attention blocks.

There are many variants in transformers, such as conformers [112], that incorporates both convolutional neural networks and transformer for local and global temporal modelling simultaneously. In Chapter 4, we show that conformers can be seamlessly applied to solve an AVSR problem.

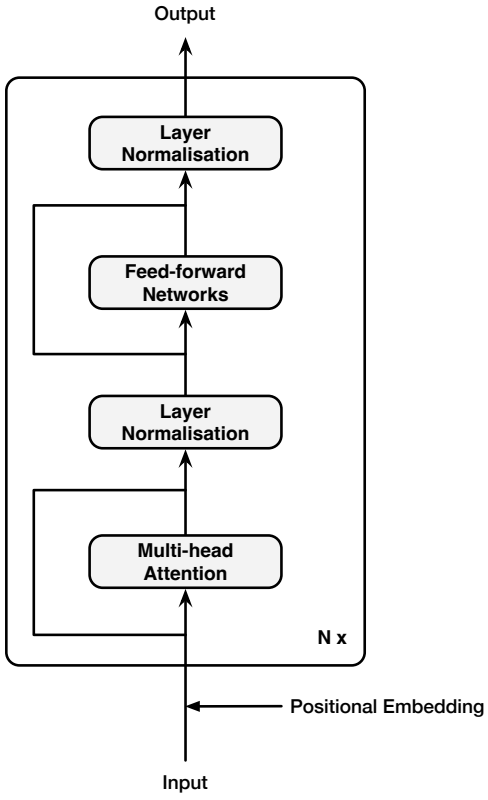


Figure 2.6: Illustration of a N multi-head attention blocks in the encoder of a transformer.

Multi-Headed Attention is a class of attention modules which performs attention mechanisms several

times in parallel. As introduced in Section 2.3.2, an attention mechanism is capable of modeling global dependencies among elements. Specifically, it computes a weighted score of the value dependent on a set of queries and keys, where the queries, keys, and values are all vectors. In a multi-headed attention module, the transformer linearly projects the queries, keys and values K times. The encoder of the transformer is composed of a stack of multi-headed blocks, The attention function is performed in parallel on each head and their embeddings are concatenated and once again projected into a feature for the next block through a dense layer. In each decoding block, there is an encoder-decoder attention which is applied to help the decoder to focus on the relevant part of the input. In particular, this encoder-decoder attention receives the features from the previous self-attention module as queries and the features from the encoder as keys and values.

Feed-Forward Network is built on top of the first LN layer in each transformer block. In a feed-forward network, there is a linear layer that projects the features with a dimension size of D_1 to a space with higher dimensional size of D_2, followed by a Rectified Linear Units (ReLU) activation function. A second linear layer, which transformed the hidden embedding back to the original hidden space, is added at the top of the module in the end. The residual connections are injected in each feed-forward module.

Layer Normalization (LN) is a type of regularisation techniques to normalise the distributions of intermediate layers. It computes the mean and variance from all the summed inputs to the neurons in a layer. Given a sample $h \in \mathbb{R}^D$, we calculate its mean μ and variance σ using:

$$\mu = \frac{1}{D}\sum_{i=1}^{D} h_i \tag{2.10}$$

$$\sigma^2 = \frac{1}{D}\sum_{i=1}^{D}(h_i - \mu)^2 \tag{2.11}$$

$$\mathrm{LN} = \gamma\frac{\mathbf{h} - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{2.12}$$

where h_i is the i-th element in $\mathbf{h}$, γ and β are learnable parameters for scaling and shifting, respectively.

Positional Embedding (PE) is a class of feature vectors that explicitly contains relative position information. A Sinusoidal positional embedding is built using sin and cos functions. The PE is formulated as

follows:

$$\mathbf{PE}_t[i] = \sin(t/10000^{2i/d}) \tag{2.13}$$

$$\mathbf{PE}_t[2i+1] = \cos(t/10000^{2i/d}), \tag{2.14}$$

where $\mathbf{PE}_t$ is defined as the positional embedding at position t of the input sequence. PE is injected to feature embeddings before the first block of multi-head self-attention block. The author hypothesised the equation shown above would allow the model to learn the relative positions.

2.5 Summary

In this chapter, we show that AVSR is still a rather challenging problem and confirmed the importance of datasets, feature extraction and network architectures in performing AVSR in naturalistic conditions. We will present our methods of AVSR for isolated words and continuous speech in Chapter 3 and Chapter 4, respectively. Next, we will explore how to leverage the unlabelled data to perform audio-visual self-supervision learning in Chapter 5. We will further investigate the Lombard effect on an end-to-end AVSR system in Chapter 6. Lastly, we will present a novel audio-visual synchronisation-based adversarial defense strategy against adversarial examples in Chapter 7.

Chapter 3

Audio-Visual Speech Recognition for Isolated Words

Contents

3.1 Methodology **46**

3.2 Experiments **53**

3.3 Results **54**

3.4 Conclusion **61**

In this chapter, we investigate the problem of AVSR for isolated words. In particular, this chapter presents various novel network architectures that can significantly improve the performance of AVSR for isolated words, including an End-to-End VSR model consisting of ResNet and BGRU (Sec. 3.1.1), a Multi-Scale Temporal Convolutional Network (MS-TCN) (Sec. 3.1.2), a Densely-Connected Temporal Convolutional Network (DC-TCN) (Sec. 3.1.3), and a Depthwise Separable Temporal Convolutional Network (DS-TCN) (Sec. 3.1.4). In addition to those architectures, this chapter discusses how different data augmentation techniques can affect the performance of those models (Sec. 3.1.5), while we also apply the knowledge distillation technique (Sec. 3.1.6) to further refine the recognition accuracy and to achieve the new state-of-the-art. The works are published in ICASSP2018 [2], ICASPP2020 [3], WACV2020 [114], ICASSP2021 [115], and ICASSP2022 [116] respectively.

VSR, also known as lip-reading, consists of the task of recognising a speaker's speech content from visual information alone, typically the movement of the lips. This is particularly useful in noisy environments

where the audio signal is corrupted, and can be used in combination with acoustic speech recognisers in order to compensate for the degraded performance due to noise. Despite of many recent advances, VSR is still a challenging task. Multiple factors contribute to the mismatch between theory and practice.

Firstly, VSR follows a two-step approach, where features were first extracted from the mouth region, with the DCT being the most popular feature extractor, and then fed to a HMMs for modeling of the temporal dynamics. The rise of deep learning has led to significant improvement in the performance of lip-reading methods. Similar to traditional approaches, the deep-learning-based methods usually consist of a feature extractor (front-end) and a sequential model (back-end). Autoencoder models were applied as the front-end in the works of [117, 118, 119] to extract deep bottleneck features (DBF) which are more discriminative than DCT features. The state-of-the-art approach for recognition of isolated words is the one proposed in [17]. It achieved the state-of-the-art performance on the LRW [19], which is the largest publicly available dataset for isolated word recognition.

Another major limitation of current deep lip-reading models barring their use in practical applications is their computational cost. Many speech recognition applications rely on on-device computing, where the computational capacity is limited, and memory footprint and battery consumption are also important factors. As a consequence, few works have also focused on the computational complexity of visual speech recognition [21, 22], but such models still trail massively behind full-fledged ones in terms of accuracy.

In this chapter, we aim to address those challenges, and we have successfully improved the overall performance to achieve a new state-of-the-art with simplified training procedures. This is achieved by combining all the latest data augmentation methods, using the recently proposed DC-TCN [114], word boundary indicators [120] and self-distillation [121]. The accuracy achieved is 92.8% for a single model and 93.4% for an ensemble. The performance can be slightly improved to 93% and 93.6%, respectively, by pre-training in a self-supervised manner on the LRS3 dataset. Secondly, we simplify the training procedure, reducing training time from 3 weeks to 1 week GPU-time, and avoid relying on a cumbersome 3-stage sequential training. For this purpose, we adopt a cosine scheduler [122] and show that training from scratch in one stage is not only feasible, but in fact can produce state-of-the-art results. Thirdy, we propose a variable-length augmentation procedure to improve the generalization capabilities of the trained model when applied to sequences of varying length (all LRW videos have a length of 29 frames). Fourthly, time masking is the most effective augmentation method followed by mixup. The use of DC-TCN

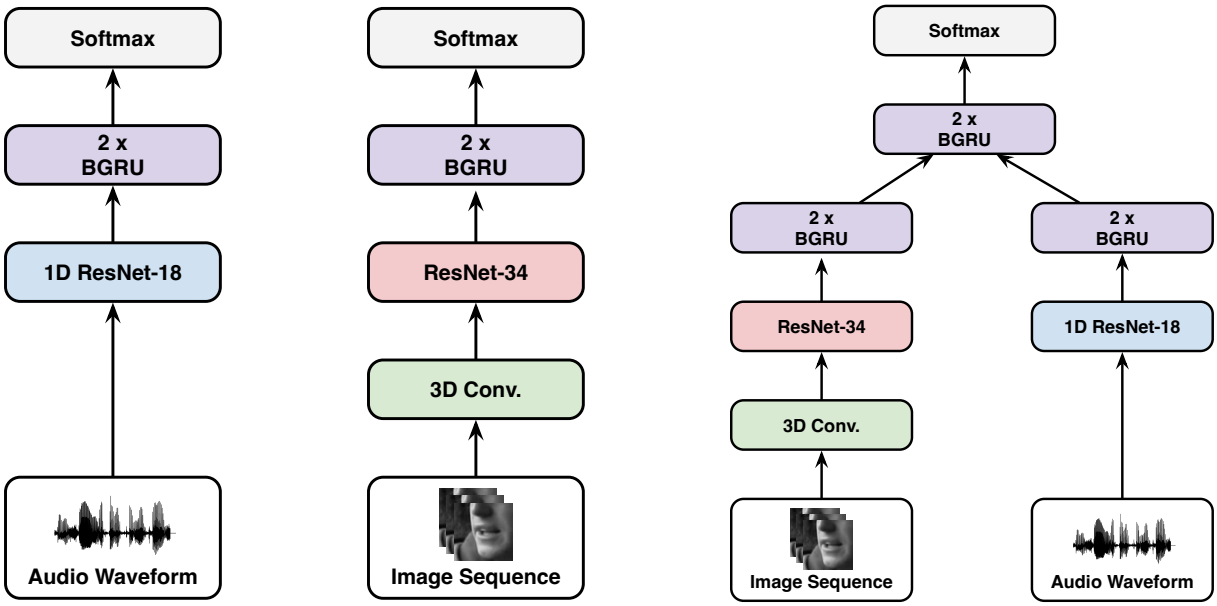


Figure 3.1: (a): BGRU based audio-only model; (b): BGRU based visual-only model; (c): BGRU based audio-visual model.

significantly outperforms the MS-TCN which in turn outperforms the BGRU model. The use of word boundaries and self-distillation is also beneficial with the former resulting in greater improvement. Finally, the error analysis suggests that all proposed methods improve performance by significantly increasing the classification accuracy of difficult words.

3.1 Methodology

3.1.1 BGRU based Temporal Model

The baseline model that we extend in this work is based on [2]. The VSR model is similar to [17] and consists of a spatiotemporal convolution followed by a 34-layer ResNet and a 2-layer BGRU. A spatiotemporal convolutional layer is capable of capturing the short-term dynamics of the mouth region and is proven to be advantageous, even when recurrent networks are deployed for back-end [16]. It consists of a convolutional layer with 64 3D kernels of 5 by 7 by 7 size (time/width/height), followed by batch normalization and rectified linear units. We use the 34-layer identity mapping version, which was proposed for ImageNet [123]. The ResNet drops progressively the spatial dimensionality until its output becomes a single dimensional tensor per time step. We should emphasize that we did not make use of

pre-trained models, as they are optimized for completely different tasks (e.g. static colored images from ImageNet or CIFAR). Finally, the output of ResNet-34 is fed to a 2-layer BGRU which consists of 1024 cells in each layer, as shown in Figure 3.1b.

The ASR model, shown in Figure 3.1a, consists of an 18-layer ResNet followed by two BGRU layers. There is no need to use a spatiotemporal convolution front-end in this case as the audio waveform is a 1D signal. We use the standard architecture for the ResNet-18 with the main difference being that we use 1D instead of 2D kernels, which are used for image data. A temporal kernel of 5ms with a stride of 0.25ms is used in the first convolutional layer in order to extract fine-scale spectral information. The output of the ResNet is divided into 29 frames using average pooling in order to ensure the same frame rate as the video is used. These audio frames are then fed to the following ResNet layers which consist of the default kernels of size 3 by 1 so that the deeper layers extract long-term speech characteristics. The output of the ResNet-18 is fed to a 2-layer BGRU which consists of 1024 cells in each layer (using the same architecture as in [17]).

In the AVSR model, as illustrated in Figure 3.1c, the BGRU outputs of each stream are concatenated and fed to another 2-layer BGRU in order to fuse the information from the audio and visual streams and jointly model their temporal dynamics. The output layer is a softmax layer which provides a label to each frame. The sequence is labelled based on the highest average probability.

3.1.2 MS-TCN based Temporal Model

The state-of-the-art approach for visual speech recognition of isolated words is the one proposed in [2]. It consists of a modified ResNet-18 backbone in which the first convolution has been substituted by a 3D convolution of kernel size of kernel size of $5 \times 7 \times 7$. The rest of the network follows a standard design up to the global average pooling layer. A Bidirectional Gated Recurrent Unit (BGRU) network follows to model temporal information.

Temporal convolutions have emerged as a promising alternative to RNNs [100], in some cases showing remarkable success on a number of tasks [99]. A temporal convolution takes a time-indexed sequence of feature vectors as input, and maps it into another such sequence (i.e., the length of the sequence is not altered) through the use of a 1D temporal convolution. Drawing a parallel to the ResNet's basic block, a temporal convolutional block consists of two sets of temporal conv-batchnorm-activation layers, and

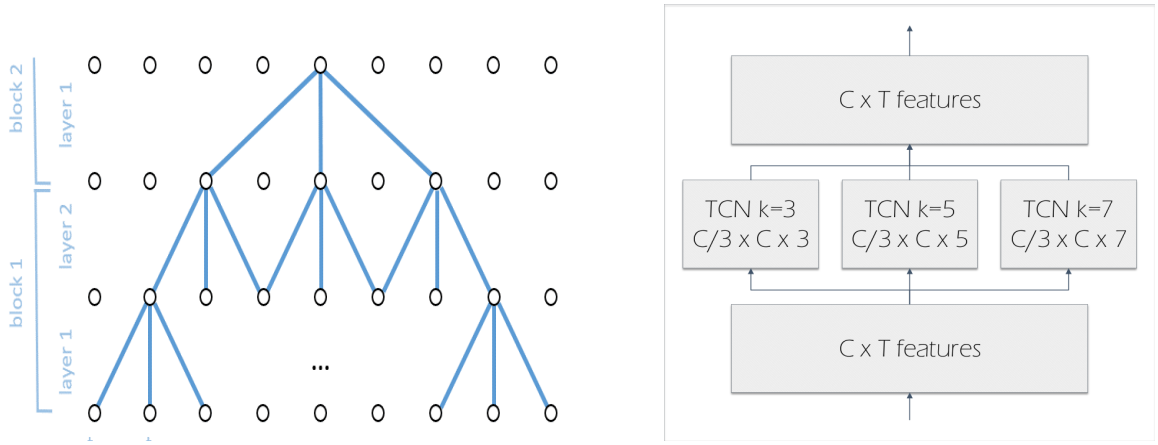
dropout can be used after each activation. A skip connection/downsample layer is also used, going from the block input to its output. Several such temporal convolutional blocks can be stacked sequentially to act as a deep feature sequence encoder. Then, a dense layer is applied to each time-indexed feature vector. Finally, since the aim is sequence classification, a consensus function, in our case a simple averaging, is used.

Dilated convolutions are typically used within TCN to increase the receptive field at a faster rate. In particular, within block i, we use a stride of 2^{i-1}. This architecture is illustrated in Figure 3.2a. It is important to note that TCN can be designed to be causal, so at time t only information prior to it is used, or non-causal. Since we are classifying the whole sequence at once, we use the latter design.

Since the input and output of a temporal convolution have the same length at the temporal domain, the receptive field of a TCN is defined by the kernel sizes and the stride. Thus, on a standard TCN, all activations at a specific layer share the same temporal receptive field. We would like to provide the network with visibility into multiple temporal scales, in a way that short term and long term information can be fused during the feature encoding. To this end, we propose a multi-scale TCN. In this TCN variant, each temporal convolution consists now of several branches, each with different kernel size. Assuming we have a number of C channels, when using n branches, each branch has C/n kernels, and their outputs are simply combined through concatenation. In this way, every convolution layer fuses information at several temporal scales. We depict this architecture in Figure 3.2b. The full lip-reading model is shown in Figure 3.2.

3.1.3 DC-TCN based Temporal Model

Although Temporal Convolutional Networks (TCN) have recently demonstrated great potential in many vision tasks, its receptive fields are not dense enough to model the complex temporal dynamics in lip-reading scenarios. To address this problem, we introduce dense connections into the network to capture more robust temporal features. Densely connected networks have received broad attention since their inception in [124], where a convolutional layer receives inputs from all its preceding layers. Such densely connected structure can effectively solve the vanishing-gradient problem by employing shallower layers and thus benefiting gradient propagation. The authors of [125] have applied dense connections to dilated convolutions to enlarge the receptive field sizes and to extract denser feature pyramid for semantic



(a) TCN (b) Multiscale TCN

Figure 3.2: (a) Temporal Convolutional Network (TCN). (b) Our Multi-scale TCN, which is used in the lip-reading model.

segmentation. Recently, a simple dense TCN for Sign Language Translation has been proposed in [126]. Our work is the first to explore the densely connected TCN for word-level lip-reading, where we present both a fully dense (FD) and a partially dense (PD) block architectures with the addition of the channel-wise attention method described in [127].

We study two approaches of constructing DC-TCN blocks. The first approach applies dense connections for all TC layers, which is denoted as the fully dense (FD) block, as illustrated at the top of Figure 3.3, where the block filter sizes set $K = \{3, 5\}$ and the dilation rates set $D = \{1, 4\}$. As shown in the figure, the output tensor of each TC layer is consistently concatenated to the input tensor, increasing the input channels by C_0 (the growth rate) each time. Note that we have a Squeeze-and-Excitation (SE) block [127] after the input tensor of each TC layer to introduce channel-wise attentions for better performance. Since the output of the top TC layer in the block typically has much more channels than the block input (e.g. C_i+4C_0 channels in Figure 3.3), we employ a 1×1 convolutional layer to reduce its channel dimensionality from $C_i + 4C_0$ to C_r for efficiency ("Reduce Layer" in Figure 3.3). A 1×1 convolutional layer is then applied to convert the block input's channels if $C_i \neq C_r$. In the fully dense architecture, TC layers are stacked in a receptive-field-ascending order.

3.1.4 DS-TCN based Temporal Model

Recent works have placed emphasis on aspects such as improving performance by finding the optimal architecture or improving generalization. However, there is still a significant gap between the current

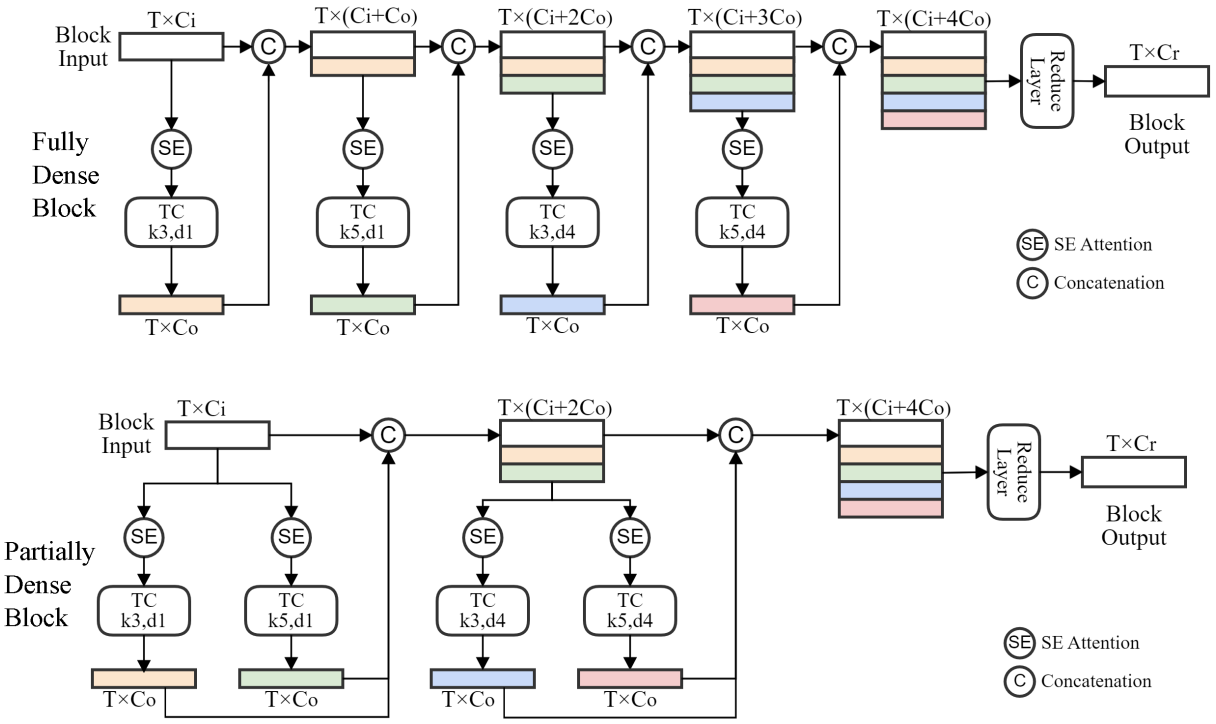


Figure 3.3: The architectures of the fully dense block (Up) and the partially dense block (bottom) in DC-TCN. We have selected the block filter sizes set $K = \{3, 5\}$ and the dilation rates set $D = \{1, 4\}$ for simplicity. In both blocks, Squeeze-and-Excitation (SE) attention is attached after each input tensor. A reduce layer is involved for channel reduction.

methodologies and the requirements for an effective deployment of lip-reading in practical scenarios. Many speech recognition applications rely on on-device computing, where the computational capacity is limited, and memory footprint and battery consumption are also important factors. As a consequence, few works have also focused on the computational complexity of visual speech recognition [21, 22], but such models still trail massively behind full-fledged ones in terms of accuracy. In this work, we propose a series of architectural changes that slashes the computational cost to a fraction of the (already quite efficient) original model. Specifically, the ResNet-18 backbone can be readily exchanged for an efficient one, such as a version of the MobileNet [128] or ShuffleNet [129] families. However, there is no such equivalent for the head classifier. The key to designing the efficient backbones is the use of depthwise separable convolutions (a depthwise convolution followed by a pointwise convolution) [130] to replace standard convolutions. This operation dramatically reduces the amount of parameters and the number of FLOPs. Thus, we devise a novel variant of the Temporal Convolutional Networks that relies on depthwise separable convolutions instead. For the purpose of this study, we use ShuffleNet v2 ($\beta\times$) as the backbone, where β is the width multiplier [129]. This architecture uses depthwise convolutions and channel shuffling

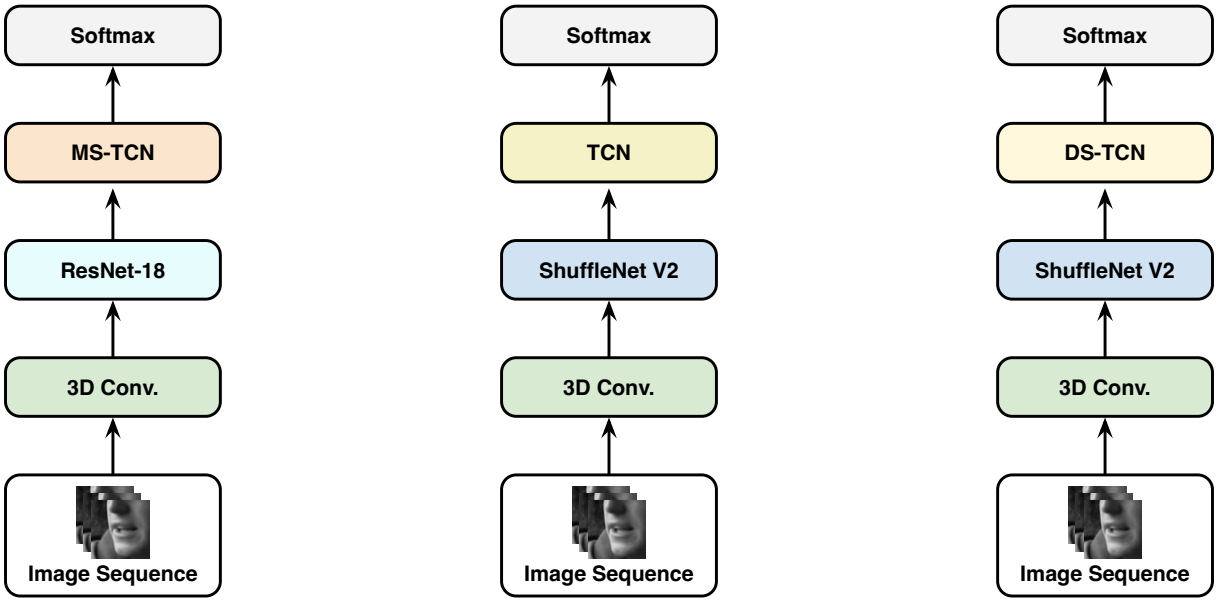


Figure 3.4: (a): Base architecture with ResNet18 and multi-scale TCN, (b): Lipreading model with ShuffleNet v2 backbone and multi-scale TCN back-end. (c): Lipreading model with ShuffleNet v2 backbone and depthwise separable TCN back-end.

which is designed to enable information communication between different groups of channels. ShuffleNet v2 (1.0×) has 5× fewer parameters and 12× fewer FLOPs than ResNet-18. The architecture is shown in Figure 3.4b.

We note that the cost of the convolution operation with kernel size greater than 1 in MS-TCN is non-negligible. To build an efficient architecture (shown in Figure 3.4c), we replace standard convolutions with depthwise separable convolutions in MS-TCN. We first apply in each channel a convolution with kernel size k, where channel interactions are directly ignored. This is followed by a point-wise convolution with kernel size 1 which transforms the C_{in} input channels to C_{out} output channels. Thus, the cost of convolution is reduced from $k \times C_{in} \times C_{out}$ (standard convolution) to $k \times C_{in} + C_{in}C_{out}$. The architecture is denoted as a Depthwise Separable Temporal Convolutional Network (DS-TCN).

3.1.5 Data Augmentation in AVSR

We also investigate how different data augmentation techniques can affect the performance of AVSR models. In this chapter, four commonly-used data augmentation techniques are examined, including Random Cropping, Flipping, Mixup, and Time Masking.

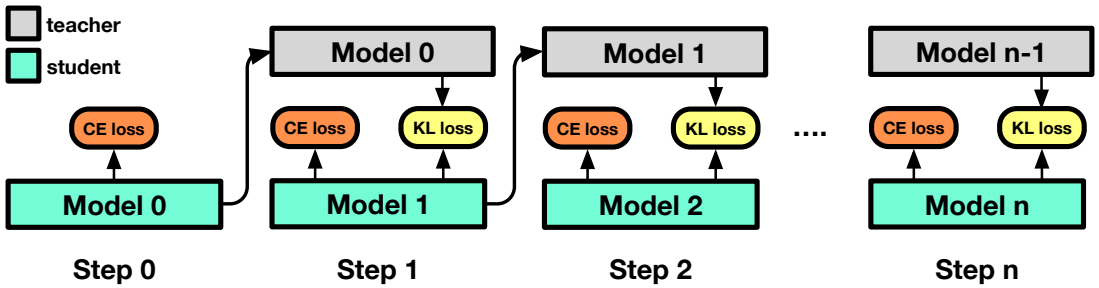


Figure 3.5: The pipeline of knowledge distillation in generations

Random Cropping: We randomly crop an 88×88 patch from the mouth ROI during training. At test time, we simply crop the central patch. This is a commonly used augmentation method that has been used successfully in several lip-reading works [3, 2].

Flipping: We randomly flip all the frames horizontally in a video with a probability of 0.5. This augmentation is commonly used in combination with random cropping [3, 2].

Mixup: We create new augmented training examples by linearly combining two input video sequences and their corresponding targets. We set the linear combination weight λ to be 0.4 similarly to [115].

Time Masking: We mask N consecutive frames for each training sequence where N is sampled between 0 and N_{max} using a uniform distribution. Each masked frame is replaced with the mean frame of the sequence it belongs to. This augmentation is based on SpecAugment [131], which has been proposed for ASR applications, and aims at making the model more robust to small segments with missing frames.

3.1.6 Knowledge Distillation

Knowledge Distillation (KD) [132] was initially proposed to transfer knowledge from a teacher model to a student model for compression purposes, i.e., the student capacity is much smaller than the teacher one. Recent studies [121, 133, 134] have experimentally shown that the student can still benefit when the teacher and student network have identical architectures. This naturally gave rise to the idea of training in generations. In particular, the student of one generation is used as the teacher of the subsequent generation. This self-distillation process, called born-again distillation, is iterated until no further improvement is observed. Finally, an ensemble can be optionally used so as to combine the predictions from multiple generations [121]. The training pipeline is shown in Fig. 3.5.

We use born-again distillation for improving the performance of the state-of-the-art model. We also use the standard knowledge distillation to train a series of efficient models where each student has smaller capacity than the teacher, which helps recover some of the performance drop of these efficient networks. In both cases, we aim to minimise the combination of cross-entropy loss ($\mathcal{L}_{\text{CE}}$) for hard targets and Kullback-Leibler (KL) divergence loss ($\mathcal{L}_{\text{KD}}$) for soft targets. Let us denote the labels as y, the parameters of the student and teacher models as θ_s and θ_t, respectively, and the predictions from the student and teacher models as z_s and z_t, respectively. $\delta(\cdot)$ denotes the softmax function and α is a hyperparameter to balance the loss terms. The overall loss function is calculated as follows:

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(y, \delta(z_s; \theta_s)) + \alpha \mathcal{L}_{\text{KD}}(\delta(z_s; \theta_s), \delta(z_t; \theta_t)) \tag{3.1}$$

Note that we have omitted the temperature term, which is commonly used to soften the logits of the $\mathcal{L}_{\text{KD}}$ term, since we found it to be unnecessary in our case.

3.2 Experiments

3.2.1 Preprocessing

We used RetinaFace [135] tracker to detect the faces and Face Alignment Network (FAN) [136] to align the landmarks. We then remove the size and rotation differences through registering faces to the mean face in the training set. A bounding box of 96×96 is used to crop the mouth ROIs. Each frame is normalised by subtracting the mean and dividing by the standard deviation of the training set.

3.2.2 Training Details

Our proposed model was trained in an end-to-end fashion, where the weights are randomly initialised. We train 80 epochs with a batch size of 32 on LRW, and measure the top-1 accuracy using the validation set to determine the best-performing checkpoint weights. We adopt AdamW [137] as the optimiser, where the initial learning rate is set to 0.0003. A cosine scheduler [122] is used to steadily decrease the learning rate from the initial value to 0. BatchNorm layers [138] are embedded to accelerate training convergence, and we use dropouts with dropping probabilities 0.2 for regularisation. The reduction ratio in the SE block is set to 16, and the channel value C_2 of DC-TCN's input tensor is set to 512. Furthermore, we adopt the

Self-Distillation Models	Top-1 Acc. (%)		
	Scratch	LiRA(LRS3)	LRS2&3+AVS
Teacher	92.1	92.3	92.9
Student 1	92.5	92.8	93.5
Student 2	92.8	92.9	93.5
Student 3	92.5	93.0	93.5
Student 4	-	92.9	93.3
Ensemble	**93.4**	**93.6**	**94.1**

Table 3.1: Performance of self-distillation models (Teacher = ResNet-18 + DC-TCN). The best-performing models from Table 3.3 are serving as teachers in first row. For each student model, the model from the line above is used as its teacher, and "Student i" stands for the model after the i-th self-distillation iteration.

variable length augmentation as proposed in [3] to increase the model's temporal robustness. We use the same training parameters as [3]. The only exception is the use of Adam with decoupled Weight decay (AdamW) [139] with $\beta_1 = 0.9, \beta_2 = 0.98$, $\epsilon = 10^{-9}$ and a L_2 penalty of 0.01.

3.2.3 Initialisation

To investigate the impact of initialisation we consider three cases: 1) we train the model from scratch using only the LRW training set, 2) we pre-train the encoder from Fig. 3.4 on the LRS3 dataset [140] using the LiRA [1] self-supervised approach and fine-tune it on the LRW training set. 3) we pre-train the encoder on LRS2 [19], LRS3 [140] and AVspeech [58] as described in [141].

3.3 Results

3.3.1 Ablation Study

Results for the ablation study are shown in Table 3.3. By removing one augmentation at a time we can estimate its contribution to the final model. We see the time masking is the most important augmentation, resulting in an absolute drop of 2.4% followed by mixup with a drop of 1.1%. By replacing DC-TCN with MS-TCN, we observe that the performance drops by 2.1 %, which demonstrates the importance of dense connections and the SE attention mechanism in DC-TCN. The performance drops by 2.4% by replacing DC-TCN with BGRU. Additionally, the removal of word boundary indicators drops the performance by 1.7 %, which demonstrates the benefits of including auxiliary boundary indicators. Finally, we pre-train the encoder in a self-supervised/supervised manner on the LRS3 / LRS2, LRS3 and AVspeech datasets

Method	Word Boundary	Top-1 Acc. (%)
3D-CNN [30]	✗	61.1
ResNet-34 + BLSTM [17]	✗	83.0
2*3D-CNN + BLSTM [142]	✗	84.1
ResNet-18 + BLSTM [120]	✗	84.3
ResNet-18 + BGRU + Cutout [143]	✗	85.0
ResNet-18 + BGRU [144]	✗	85.0
ResNet-18 + MS-TCN [3]	✗	85.3
ResNet-18 + MS-TCN + S.D. [115]	✗	88.5
ResNet-18 + DC-TCN [114]	✗	88.4
Ours (w/o S.D., Scratch)	✗	**90.4**
Ours (w/o S.D., LRS2&3+AVS)	✗	**91.1**
Ours (Ensemble, Scratch)	✗	**91.6**
Ours (Ensemble, LRS2&3+AVS)	✗	**92.1**
ResNet-18 + BGRU [144]	✓	88.4
ResNet-18 + BLSTM [120]	✓	88.8
Ours (w/o S.D., Scratch)	✓	**92.1**
Ours (w/o S.D., LiRA(LRS3))	✓	**92.3**
Ours (w/o S.D., LRS2&3+AVS)	✓	**92.9**
Ours (Ensemble, Scratch)	✓	**93.4**
Ours (Ensemble, LiRA(LRS3))	✓	**93.6**
Ours (Ensemble, LRS2&3+AVS)	✓	**94.1**

Table 3.2: Comparison with state-of-the-art methods on the LRW dataset in terms of classification accuracy. Experiments are divided into two groups, with and without utilising word boundaries indicators, respectively. "S.D.": self-distillation. "Scratch", "LiRA(LRS3)" and "LRS2&3+AVS" correspond to the three pre-training strategies in Table 3.3.

and then fine-tune the model on the LRW training set, and this slightly increases the performance to 92.3 % / 92.9 %. It is clear from Table 3.2 that the proposed models significantly outperform the current state-of-the-art.

3.3.2 Self-Distillation

Results for self-distillation experiments are presented in Table 3.1. We use the best two models from Table 3.3 as teachers in the first round. It is clear that self-distillation results in a 0.6 % to 0.7 % absolute improvement in all cases. In addition, an ensemble of all models (all students + teacher) leads to a further absolute improvement of 0.6 %. These results suggest that self-distillation is beneficial for lip-reading. However, we should point out that the improvement is smaller compared to [115], probably due to the

Temporal Model	Data Augmentation				Word Boundary	Pre-training Strategies			Top-1 Acc. (%)
	Crop	Flip	Mixup	TM		Scratch	LiRA(LRS3)	LRS2&3+AVS	
DC-TCN [114]	✓	✓	✓	✓	✓	-	-	✓	**92.9**
	✓	✓	✓	✓	✓	-	✓	-	92.3
	✓	✓	✓	✓	✓	✓	-	-	92.1
	-	✓	✓	✓	✓	✓	-	-	91.8
	✓	-	✓	✓	✓	✓	-	-	91.7
	✓	✓	-	✓	✓	✓	-	-	91.0
	✓	✓	✓	-	✓	✓	-	-	89.7
	✓	✓	✓	✓	-	✓	-	-	90.4
MS-TCN [3]	✓	✓	✓	✓	✓	✓	-	-	90.0
BGRU [2]	✓	✓	✓	✓	✓	✓	-	-	89.7

Table 3.3: Ablation studies of three temporal models on LRW dataset. Starting from the best-performing DC-TCN model, we remove each data augmentation and the word boundaries indicators to examine their effectiveness. Then we replace the DC-TCN with MS-TCN and BGRU. "Scratch" denotes a model trained from scratch without using external data. "LiRA(LRS3)" indicates a self-supervised pre-trained model using LiRA [1] on the LRS3 dataset, and "LRS2&3+AVS" indicates a fully supervised pre-trained model on LRS2, LRS3 and AVSpeech.

much better teacher model which makes further improvement harder.

3.3.3 Audio-Visual Experiments

While lip-reading can be used in isolation, the most useful scenario is when combined with audio to improve performance in noisy environments. In this section we show the performance of our model when trained on audio only, visual only and audio-visual data under varying levels of babble noise. The audio-only and audio-visual models are based on [2] but we apply the proposed changes as shown in Figure 3.2. The performance under different Signal to Noise Ratio (SNR) levels is shown in Figure 3.6. We also compute the performance of a TCN network trained with MFCC features. We use 13 coefficients (and their deltas) using a 40ms window and a 10ms step. Performance of MFCCs is similar to the audio-only model at high SNRs but becomes worse at low SNRs.

The audio-visual model is slightly better than the audio-only model at low SNRs but yields a clear advantage at higher levels of noise. In particular, when using a clean audio signal, the audio-only model attains 1.54% error rate, while the audio-visual model attains 1.04%. In the presence of heavy noise, e.g. 0 dB, the audio-visual error rate is 2.92%, while performance for the audio-only model goes down to 8.57%. Similarly at -5dB, the audio-visual model achieves an error rate of 6.53% significantly outperforming the audio-only model which has an error rate of 26.21%. We further compare the performance of the

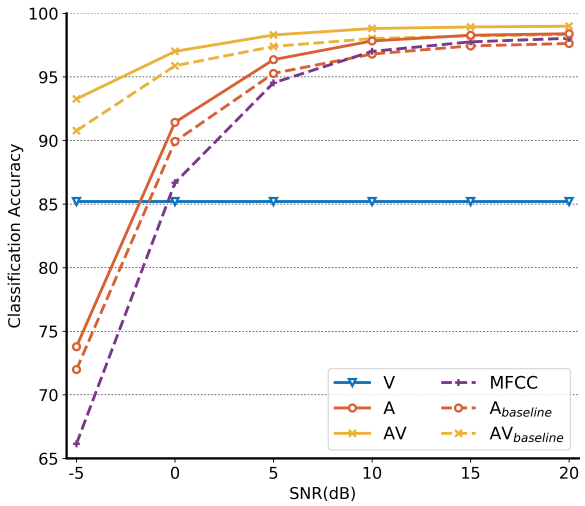


Figure 3.6: Performance for audio-only (A), video-only (V) and audio-visual (AV) models under different babble noise levels. The baseline corresponds to the model presented in [2].

audio-only and audio-visual models with respect to our baseline [2], showing a clear gain throughout the different noise levels.

3.3.4 Efficient Training

Given that we use a purely convolutional architecture, it is reasonable to test whether is possible to train a lip-reading model from scratch. We empirically found that in fact it is possible to successfully train the full-fledged model from scratch and achieve state of the art performance. To this end, we adopt a cosine scheduler, which has been shown to be particularly effective [122]. Such training leads to competitive performance in 1 week GPU-time.

However, we observe that it is also possible to first pre-train on a subset of the 10% hardest words, which amounts to 50 classes for LRW†. Such initialization allows for faster training, and even yields a small performance improvement. Thus, we adopt this pre-training strategy as it adds a minimal training overhead.

Student Backbone (Width mult.)	Student Back-end (Width mult.)	Distillation	Top-1 Acc.	Params $\times 10^6$	FLOPs $\times 10^9$
ResNet-18 [3]	MS-TCN (3×)	-	85.3	36.4	10.31
ResNet-34 [2]	BGRU (512)	-	83.4	29.7	18.71
MobiVSR-1 [22]	TCN	-	72.2	4.5	10.75
ShuffleNet v2 (1×)	MS-TCN (3×)	✗	84.4	28.8	2.23
	MS-TCN (3×)	✓	85.5	28.8	2.23
ShuffleNet v2 (1×)	DS-MS-TCN (3×)	✗	84.4	9.3	1.26
	DS-MS-TCN (3×)	✓	85.3	9.3	1.26
ShuffleNet v2 (1×)	TCN (1×)	✗	81.0	3.8	1.12
	TCN (1×)	✓	82.7	3.8	1.12
ShuffleNet v2 (0.5×)	TCN (1×)	✗	78.1	2.9	0.58
	TCN (1×)	✓	79.9	2.9	0.58

Table 3.4: Performance of different efficient models, ordered in descending computational complexity, and their comparison to the state-of-the-art on the LRW dataset. We use a sequence of 29-frames with a size of 88 by 88 pixels to compute the multiply-add operations (FLOPs). The number of channels is scaled for different capacities, marked as 0.5×, 1×, and 2×. Channel widths are the standard ones for ShuffleNet V2, while base channel width for TCN is 256 channels.

3.3.5 Efficient Models

The frame encoder can be made more efficient by replacing the ResNet-18 with a lightweight ShuffleNet v2 (shown in Figure 3.4b), as explained in section 3.1.3. We should note that we maintain the first convolution of the network as a 3D convolution. Preliminary experiments showed ShuffleNet v2 [129] yields superior performance over other lightweight architectures like MobileNetV2 [145]. It can be seen in Table 3.4 that this change results in a drop of 0.9 % in accuracy while reducing both the number of parameters and FLOPs.

The next step is the replacement of the MS-TCN head with its depthwise-separable variant, noted as DS-MS-TCN. As shown in Table 3.4 this variant leads to a model with almost one third of parameters and a 50 % reduction in FLOPs while achieving the same accuracy as the ShuffleNet v2 with a MS-TCN head.

Models can become even lighter (shown in Figure 3.4c) by reducing the number of heads to 1, denoted by TCN, and by reducing the width multiplied of the ShuffleNet v2 to 0.5. In the former case, performance drops by 3.4 %, and in the latter by a further 1.9 % resulting in accuracy of 78.1 %. However, it should be

[†] The list of "hardest words" is obtained from [17]

noted that the number of parameters and FLOPs is significantly reduced for both models.

In order to partially bridge this gap, we explore Knowledge Distillation once again. Since now there are higher capacity models that can act as teachers, we do not need to resort to self distillation. We first explored the standard distillation approach in which we take the best-performing model as the teacher. However, it is known that a wider gap in terms of architecture might mean a less effective transfer [146]. Thus, we also explore a sequential distillation approach. More specifically, for lower-capacity networks, we use intermediate-capacity networks to more progressively bridging the architectural gap. For example, for the ShuffleNet v2 (1×)+DS-MS-TCN, we can first train a model using the full fledged ResNet-18+MS-TCN model as teacher, and use the ShuffleNet v2 (1×)+MS-TCN as the student. Then, on the second step, we use the latter model as the teacher, and train our target model, ShuffleNet v2 (1×)+DS-MS-TCN, as the student. This procedure resembles the self-distillation strategy described above in the sense that trains a sequence of teacher-student pairs, where the previous student becomes the teacher in the next iteration. However, unlike that strategy, it progressively changes the architecture from the full-fledged model to the target architecture.

The results on the LRW dataset are shown in Table 3.4. Replacing the state-of-the-art ResNet-18+MS-TCN with ShuffleNet v2 (1×)+ DS-MS-TCN leads to the same accuracy, after distillation, than the previous state-of-the-art MS-TCN of [3], while requiring 8.2× fewer FLOPs and 3.9× fewer parameters. This is a significant finding since the MS-TCN is already quite efficient, having slightly lower computational cost than the lightweight architecture of MobiVSR-1 [22]. Another interesting combination is the ShuffleNet v2 (0.5×)+ TCN model, which achieves 79.9 % accuracy on LRW with as little as 0.58G FLOPs and 2.9M parameters, a reduction of 17.8× and 12.5× respectively when compared to the ResNet-18+MS-TCN model of [3].

3.3.6 Error Analysis

Difficulty Categories In order to better understand how the presented models improve the word classification accuracy, we perform some error analysis. We divide the test samples in the LRW dataset into five groups [114]. Each group contains 100 distinct isolated words and it is created based on the word accuracy of the model in [2]. The 100 words with the highest classification accuracy are grouped in the "Very Easy" group, the next 100 words in the "Easy" group and so on. The average classification accuracy

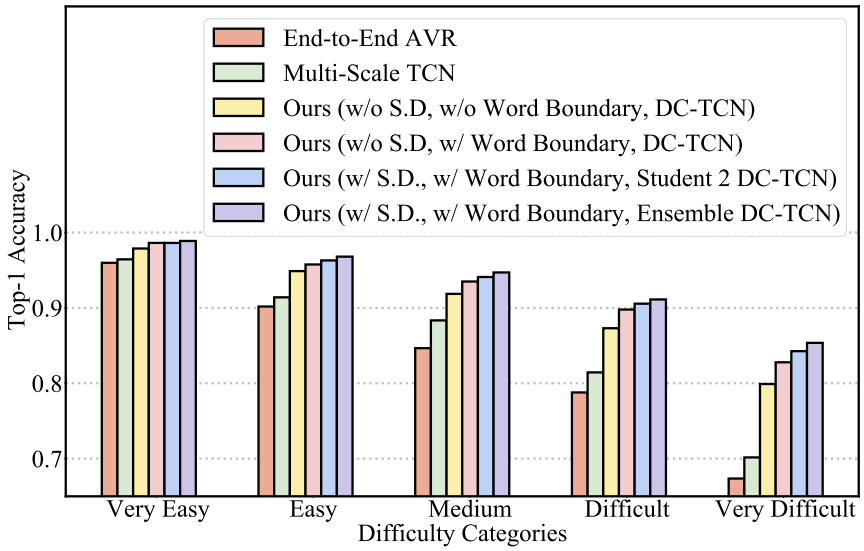


Figure 3.7: A comparison of our method and two baseline methods (End-to-End AVR [2] and Multi-Scale TCN [3]) on the five difficulty groups of the LRW test set.

Drop N Frames →	N=0	N=1	N=2	N=3	N=4	N=5
End-to-End AVR [2]	84.6	80.2	71.3	59.5	45.9	32.9
MS-TCN [3]	85.3	83.5	81.2	78.7	75.7	71.5
Ours (PD)	88.4	86.2	84.0	81.0	77.5	73.3
Ours (FD)	88.0	86.4	83.6	81.3	77.7	73.8

Table 3.5: The top-1 accuracy of different methods on LRW where N frames are randomly removed from each testing sequence.

in each group is shown in Fig. 3.7. For comparison purposes, we also include the performance of [3] and [2]. We can see that our models outperform the two baselines across all groups and the improvement is more pronounced in the "Difficult" and "Very Difficult" groups.

Variable Lengths We further evaluate the temporal robustness of different models against video sequences with variable lengths, i.e. N frames are randomly dropped from each testing sequence in LRW dataset where N ranges from 0 to 5. As shown in Table 3.5, the performance of End-to-End AVR [2] drops significantly as increasing frames are randomly removed from the testing sequences. In contrast, MS-TCN [3] and our DC-TCN (both PD and FD) demonstrate better tolerance to such frame removals, mainly due to the usage of variable length augmentation [3] during training. Besides, the accuracy of our models (both PD and FD) constantly outperforms that of MS-TCN [3] no matter how the number of frames to remove varies, which verifies the superior temporal robustness of our method.

3.4 Conclusion

In this chapter, we study the problem of AVSR for isolated words. We start from an end-to-end visual speech recognition model [31]. In contrast to performing VSR experiments using BLSTMs, we present an AVSR model that performs classification using BGRUs. Next, our work reveals that replacing the BGRU layers with TCN could achieve even better performance than recurrent layers. Given the fact that the receptive fields of TCN module are limited, we further introduce a DC-TCN. Characterised by the dense connections and the SE attention mechanism, the proposed DC-TCN could capture more robust features at denser temporal scales and therefore improves the performance of the original TCN architectures. We show that DC-TCN have surpassed the performance of all baseline methods on the LRW dataset. Additionally, we simplify the training procedure and reduce training time from 3 weeks to 1 week by using of TCN and cosine scheduler. Furthermore, we implement a series of architecture changes to develop efficient lip-reading models. In particular, the frame encoder could be made more lightweight by replacing the ResNet-18 with a ShuffleNet v2. We also replace the MS-TCN head with its depthwise-separable variant, noted as DS-MS-TCN.

We study not only the architecture design but also data augmentation techniques at the temporal domain. Specifically, we improve the generalisation capabilities to sequences of varying length by the use of variable length augmentation. We further push the state-of-the-art performance on LRW by applying time masking.

In future work we will investigate the performance of the proposed approach on other databases with more extreme poses like LRS3 and on continuous visual speech recognition. It will be interesting to investigate in future work how cross-modal distillation affects the performance of AVSR models. In the next chapter, we will focus on the problem of sentence-level AVSR in in-the-wild scenarios.

Chapter 4

Audio-Visual Speech Recognition for Continuous Speech

Contents

4.1 Methodology **64**

4.2 Experimental Setup **74**

4.3 Results **78**

4.4 Conclusion **87**

In the previous chapter, we presented deep AVSR models for isolated words, which achieved very impressive results on in-the-wild lip-reading datasets. In this chapter, we present AVSR models for continuous speech recognition, which is a more challenging and realistic task than isolated-word classification.

VSR, also known as lipreading, is the task of automatically recognising speech from video based only on the lip movements. This is a field which attracted a lot of research attention in the past within the speech recognition community [147, 148] but failed to meet the initial high expectations. As a consequence, research interest declined and no further progress was made. The two main reasons why the first generation of VSR models fell short are the following: 1) The lack of large transcribed audio-visual datasets resulted in models which could only recognise a limited vocabulary and work only in a laboratory environment, 2) The use of hand-crafted visual features, which might not have been optimal for VSR applications, prevented the development of high accuracy models. Recently, large audio-visual transcribed datasets,

like LRS2 [19] and LRS3 [51], have become available which have allowed the development of large vocabulary and robust models. In addition, advances in deep learning have made possible the use of end-to-end models which learn to extract VSR-related features directly from the raw images. These developments have led to a new generation of deep learning based VSR models which achieve much higher accuracy than older models and also work in unseen real-life situations.

The constant improvement of VSR models is mainly fuelled by using increasingly larger transcribed datasets, which are usually not publicly available, and the development of new models which work well when trained with huge amounts of data. Some recent works [32, 34] use tens of thousands of hours of non-publicly available training data in order to achieve state-of-the-art performance on standard benchmarks. In contrast to this recent trend, we demonstrate that carefully designing a model is equally important to using larger training sets. Our approach consists of 22three key ingredients: 1) hyper-parameter optimisation of an existing architecture, 2) appropriate data augmentations, and 3) addition of prediction-based auxiliary tasks to a VSR model. This leads to a great reduction in word error rate (WER) and results in state-of-the-art performance in almost all benchmarks. This is achieved by using only publicly available datasets which are two orders of magnitude smaller than the ones used in previous works. We also show that combining multiple datasets further improves the performance which is in line with the results reported in the literature. Hence, we argue that further progress in the field can be achieved not only by increasing the size of the training data but also by designing suitable architectures.

The vast majority of existing works focus on improving the performance of English-only VSR models. There are also few works which design models tailored to a specific language, like Mandarin [149, 53, 150]. In contrast to previous works, our approach is evaluated not only on English but also on Mandarin and Spanish, which are the two other widely spoken languages, Italian, French and Portuguese. State-of-the-art performance is achieved in all languages.

Specifically, in this chapter, we make the following contributions:

- We propose a novel method for visual speech recognition, which outperforms state-of-the-art methods trained on publicly available data by a large margin.
- We do so by a VSR model with auxiliary tasks that jointly performs visual speech recognition and prediction of audio and visual representations.

- We demonstrate that the proposed VSR model performs well not only in English but also in other languages, like Spanish, Mandarin, Italian, French and Portuguese.

- We show that enlarging the training sets, even by including unlabelled data with automatically generated transcriptions or videos in other languages, results in improved performance. This provides further evidence to the hypothesis that the recent improvements presented in the literature are probably the result of larger training sets and not necessarily of better models.

- We extend the AVSR model presented in [24] to an end-to-end model and perform a comparison between audio-only models trained with log-Mel filter-bank features and raw waveforms. Although in clean conditions they both perform similarly, the raw audio model performs slightly better in noisy conditions.

Our method outperforms state-of-the-art methods by a large margin for visual speech recognition in multiple languages. In what follows we explain the details of our approach and the changes that we have made to the training strategy and architecture which led to this highly improved performance.

4.1 Methodology

4.1.1 Our Approach

In contrast to previous works which improve the VSR performance by using increasingly larger training sets, we focus on improving the performance by carefully designing a model without relying on additional data. This is achieved by revising the training strategy and architecture of the state-of-the-art model proposed in [151]. Firstly, we optimise hyperparmeters and improve the language model with the aim of squeezing extra performance out of the model. Secondly, we introduce time-masking which is a temporal augmentation method and is commonly used in ASR models. It significantly improves the VSR performance by forcing the model to rely more on contextual information and as a consequence, it can better disambiguate similar lip movements which correspond to different phonemes. Finally, we use a VSR model with auxiliary tasks where the model jointly performs visual speech recognition and prediction of audio and visual representations extracted from pre-trained VSR and ASR models (as described in Sections 4.2.7 and 4.2.8, respectively). This prediction task provides additional supervisory signal and

Table 4.1: The architecture of the front-end encoder of the VSR model. The filter shapes are denoted by {Temporal Size × Spatial Size^2, Channels} and {Spatial Size^2, Channels} for 3D convolutional and 2D convolutional Layers , respectively. The sizes correspond to [Batch Size, Channels, Sequence Length, Height, Width] and [Batch Size × Sequence Length, Channels, Height, Width], for 3D and 2D convolutional layers, respectively. T_v denotes the number of input frames.

Component Name	Layer Type	Input Size	Output Size
Stem_1	Conv 3D, 5×7^2, 64	[B, 1, T_v, 88, 88]	[B, 64, T_v, 44, 44]
	3D Max Pooling, 1×3^2	[B, 64, T_v, 44, 44]	[B, 64, T_v, 22, 22]
Reshape	-	[B, 64, T_v, 22, 22]	[B×T_v, 64, 22, 22]
Residual Block_2	$\begin{bmatrix}\text{Conv 2D}, 3^2, 64\\ \text{Conv 2D}, 3^2, 64\end{bmatrix} \times 2$	[B×T_v, 64, 22, 22]	[B×T_v, 64, 22, 22]
Residual Block_3	$\begin{bmatrix}\text{Conv 2D}, 3^2, 128\\ \text{Conv 2D}, 3^2, 128\end{bmatrix} \times 2$	[B×T_v, 64, 22, 22]	[B×T_v, 128, 11, 11]
Residual Block_4	$\begin{bmatrix}\text{Conv 2D}, 3^2, 256\\ \text{Conv 2D}, 3^2, 256\end{bmatrix} \times 2$	[B×T_v, 128, 11, 11]	[B×T_v, 256, 6, 6]
Residual Block_5	$\begin{bmatrix}\text{Conv 2D}, 3^2, 512\\ \text{Conv 2D}, 3^2, 512\end{bmatrix} \times 2$	[B×T_v, 256, 6, 6]	[B×T_v, 512, 3, 3]
Aggregation	2D Global Average Pooling	[B×T_v, 512, 3, 3]	[B×T_v, 512, 1, 1]
Reshape	-	[B×T_v, 512, 1, 1]	[B, 512, T_v]

forces the model to learn better visual representations. A diagram of the architecture of our model is shown in Fig. 4.4c.

The performance of the our model can be seen in Tables 4.8 to 4.12. Due to the random nature of training we train 10 models for each experiment and we report the mean and standard deviation of the WER over the 10 runs. This is in contrast to previous works which report just a single value, most likely the best WER, and no standard deviation, and it provides a more robust estimate of the actual performance. However, in order to facilitate a fair comparison with other works, we also report the best WER of the 10 runs.

4.1.2 Architecture

The model consists of 4 modules, a front-end encoder (VSR encoder in Fig. 4.4c), a back-end encoder, a hybrid CTC and transformer decoder and two predictors. In particular, the encoder receives as input

Table 4.2: The architecture of the front-end encoder of the ASR model. The filter shapes are denoted by {Temporal Size, Channels} for 1D Convolutional Layers, respectively. The sizes correspond to [Batch Size, Channels, Sequence Length]. T_a denotes the length of audio waveforms.

Component Name	Layer Type	Input Size	Output Size
Stem_1	Conv 1D, 80, 64	[B, 1, T_a]	[B, 64, $T_a//4$]
Residual Block_2	$\begin{bmatrix}\text{Conv 1D}, 3, 64\\ \text{Conv 1D}, 3, 64\end{bmatrix} \times 2$	[B, 64, $T_a//4$]	[B, 64, $T_a//4$]
Residual Block_3	$\begin{bmatrix}\text{Conv 1D}, 3, 128\\ \text{Conv 1D}, 3, 128\end{bmatrix} \times 2$	[B, 64, $T_a//4$]	[B, 128, $T_a//8$]
Residual Block_4	$\begin{bmatrix}\text{Conv 1D}, 3, 256\\ \text{Conv 1D}, 3, 256\end{bmatrix} \times 2$	[B, 128, $T_a//8$]	[B, 256, $T_a//16$]
Residual Block_5	$\begin{bmatrix}\text{Conv 1D}, 3, 512\\ \text{Conv 1D}, 3, 512\end{bmatrix} \times 2$	[B, 256, $T_a//16$]	[B, 512, $T_a//32$]
Aggregation	1D Average Pooling, Stride 20	[B, 512, $T_a//32$]	[B, 512, $T_a//640$]

the raw images and maps them to visual speech representations which are fed to the back-end encoder. This is followed by a CTC and transformer decoder which generates the predicted characters. Finally, the features extracted from the middle position of the back-end encoder flow through two separate predictors to predict visual and acoustic speech representations from pre-trained VSR and ASR models, respectively.

The **front-end encoder** consists of a 3D convolutional layer with a kernel size of $5 \times 7 \times 7$ followed by a ResNet-18 [152, 17]. Let $B \times T \times H \times W$ be the input tensor to the visual front-end module, where B, T, H, and W correspond to batch size, number of frames, height and width, respectively. The visual features at the top of the residual blocks are aggregated along the spatial dimension by a global average pooling layer, resulting in a feature output of dimensions $B \times C \times T$, where C indicates the channel dimensionality. The Swish activation functions is used in all layers. The detailed architecture can be seen in Table 4.1.

The **back-end encoder** starts with a positional embedding module, followed by a stack of 12 conformer blocks. The positional embedding module is a linear layer, which projects the features from the output of ResNet-18 to a 256-dimensional space. The transformed features are further injected with relative position information [153]. In each conformer block, a feed-forward module, a self-attention module, a convolution module, and a second feed-forward module are stacked in order. Specifically, the feed-forward module is comprised of a linear layer, which projects the features to a higher 2048-dimensional space,

followed by a Rectified Linear Unit (ReLU) activation function, a dropout layer with a probability of 0.1, and a second linear layer with output dimension of 256. Half-step residual connections are also used in each feed-forward module. The self-attention module is capable of modeling global dependencies among elements. The module maps the query and a set of key-value pairs through an attention map, which focuses on different parts of the input. Instead of performing a single attention function, a multi-head mechanism is leveraged with different linear projections to a lower 64-dimensional space. The attention function is performed in parallel on each head and the outputs are concatenated into a 256-dimensional space and once again projected into the final values. The convolutional module, which excels at capturing local patterns efficiently, is composed of an 1D point-wise convolutional layer, Gated Linear Units (GLU) [154], an 1D depth-wise convolutional layer, a batch normalisation layer, a swish activation layer, a 1D point-wise convolutional layer, and a layer normalisation layer. The combination of self-attention and convolution is capable of better capturing both local and global temporal information compared to the standard transformer architecture [112].

The **decoder** is composed of an embedding module and a set of residual multi-head attention blocks. It takes as input the encoded sequence and the prefixes of the target sequence. First, the prefixes from index 1 to l - 1 are projected to embedding vectors, where l is the target length index. The absolute positional encoding [67] is also added to the embedding. Next, the embedding is fed to a stack of multi-head attention blocks. Each block consists of a self-attention module, an encoder-decoder attention module and a feed-forward module. Layer normalisation is added before each module. Specifically, the self-attention module is slightly different from the one in the encoder where future positions at its attention matrix are masked out, followed by an encoder-decoder attention, which helps the decoder to focus on the relevant part of the input. This attention receives the features from the previous self-attention module as Q and the features from the encoder as K and V ($K = V$). The features are further fed to a feed-forward module, which is the same as the one used in the encoder. Finally, a layer normalisation and a linear layer are added which predict the posterior distribution of the next generated token.

A **linear layer** with a softmax function, which maps the encoded features to the predicted character sequence is also used on top of the back-end encoder. This layer is trained with the Connectionist Temporal Classification (CTC) loss.

The **predictor** is a linear layer which takes as input the features at the middle block (6th) of the back-end

encoder and predicts the corresponding audio/visual features from the pre-trained ASR/VSR models. Separate predictors are employed for each prediction task. Both the input and output dimensions of the linear layer are 256.

4.1.3 Prediction-based Auxiliary Tasks

The standard approach to visual speech recognition relies on end-to-end training which allows the entire model to be optimised towards the desired target. This is an attractive property and has led to impressive results but also results in significant challenges in training such a large model. One solution which has been recently proposed is the use of auxiliary tasks in the form of additional losses applied to intermediate layers of the model [155, 156, 157]. This acts as regularisation which helps the model learn better representations and leads to better generalisation on test data.

Based on this observation we propose as an auxiliary task the prediction from intermediate layers of audio and visual representations learned by pre-trained ASR and VSR models (see Fig. 4.4c). This is inspired by the recent success of prediction tasks in self-supervised learning. In particular, good audio representations can be learned by predicting speech features such as Log power spectrum (LPS) and MFCCs or by using joint audio and visual supervision [158]. Similarly, visual speech representations can be learned by predicting audio features [159]. Hence, the proposed auxiliary task provides additional supervision to the intermediate layers of the model which in turns results in better visual representations and improved performance. Mathematically, this is formulated as a regression problem where the goal is to minimise the L1 distance between the predicted and pre-trained visual and audio features. This results in the following loss term added to loss function:

$$\begin{aligned}\mathcal{L}_{AUX} = \beta_a \left\| h_a(f^l(\mathbf{x}_v)) - g_a^l(\mathbf{x}_a) \right\|_1 \\ + \beta_v \left\| h_v(f^l(\mathbf{x}_v)) - g_v^l(\mathbf{x}_v) \right\|_1 \end{aligned} \tag{4.1}$$

where $\mathbf{x}_v$ and $\mathbf{x}_a$ are the visual and audio input sequences, respectively, g_v and g_a are the pre-trained visual and audio encoders, respectively. f is the subnetwork up to layer l whose intermediate representation is used as input to the audio and visual predictor h_a and h_v, respectively. β_a and β_v are the coefficients for

each loss term and $\|\cdot\|_1$ is the ℓ_1-norm.

The model performs VSR and at the same time attempts to predict audio and visual representations from intermediate layers. Hence, the final loss is simply the addition of the main VSR loss and the auxiliary loss as follows:

$$\mathcal{L} = \mathcal{L}_{VSR} + \mathcal{L}_{AUX} \tag{4.2}$$

$$\mathcal{L}_{VSR} = \alpha\mathcal{L}_{CTC} + (1 - \alpha)\mathcal{L}_{att} \tag{4.3}$$

where $\mathcal{L}_{VSR}$ is the loss of the hybrid CTC/attention architecture used. $\mathcal{L}_{CTC}$ is the CTC loss, $\mathcal{L}_{att}$ the loss of the attention mechanism and α controls the relative weight of each loss term. Further details about the losses can be found in section 4.1.4 in the Supplementary Information.

The significant impact of the auxiliary losses on performance can be seen in Table 4.5. Removing either loss, i.e., either the first or second term from equation 4.1, leads to an increase in the mean WER for both datasets. In case both losses are removed, i.e., no auxiliary loss is used, then the increase in the mean WER is even greater. Finally, the removal of the two losses and time masking results in a significant decrease in performance.

An ablation study on the effect of layer l where the auxiliary loss (equation 4.1) is attached is shown in Fig. 4.1. Layer 6 was found to be the optimal level based on the performance on the validation set. All results reported in all Tables are based on this configuration.

We investigate the effect of the layer l where the auxiliary loss (equation 4.1) is attached. The position of layer varies from 0 to 12 at intervals of 2. Layer 6 was found to be the optimal level on the validation set of LRS2. Results are presented in Figure 4.3.

4.1.4 Loss Functions

To map input sequences $\mathbf{x} = [x_1, ..., x_T]$ such as audio or visual streams to corresponding target characters $\mathbf{y} = [y_1, ..., y_L]$, we consider a hybrid CTC/attention architecture [160] in this paper, where T, L are the lengths of the input sequence and target character sequence, respectively. The CTC loss assumes conditional independence between the output predictions and the estimated sequence posterior has the form of $P_{CTC}(\mathbf{y}|\mathbf{x}) \approx \prod_{t=1}^{T} p(y_t|\mathbf{x})$. The CTC loss from equation 4.3 is defined as follows:

$$\mathcal{L}_{CTC} = -\log P_{CTC}(\mathbf{y}|\mathbf{x}) \tag{4.4}$$

An attention-based encoder-decoder model gets rid of this assumption by directly estimating the posterior on the basis of the chain rule and has a form of $P_{att}(\mathbf{y}|\mathbf{x}) \approx \prod_{l=1}^{L} p(y_l|y_{<l}, \mathbf{x})$. In this case the $\mathcal{L}_{att}$ from equation is:

$$\mathcal{L}_{att} = -\log P_{att}(\mathbf{y}|\mathbf{x}) \tag{4.5}$$

The objective function of speech recognition is performed by a linear combination of the CTC loss and a cross-entropy loss as shown in equation 4.3. The α value used in this work is 0.1.

A grid search was performed for the parameters β_a and β_v used in the auxiliary loss (equation 4.1). The values that resulted in the best performance in the validation set of the LRS2 dataset are the following: β_a = 0.4 and β_v = 0.4. These values are used for all experiments.

4.1.5 Using Additional Training Data

Using larger and larger training sets is a recent trend in the literature in order to reduce the WER. In order to investigate the impact of the amount of training data we train models on varying amounts of data. We start by training models using only the training set of each database (ninth row of Table 4.8 and sixth row of Table 4.9). It is not possible to train a model from scratch on the LRS2 and LRS3 datasets so we use curriculum learning. This means that we first use only short utterances and as training progresses we keep adding longer ones. Further details on curriculum learning can be found in section 4.1.6 in the Supplementary Information. Then we use a model trained for recognising 500 English words [115] on the

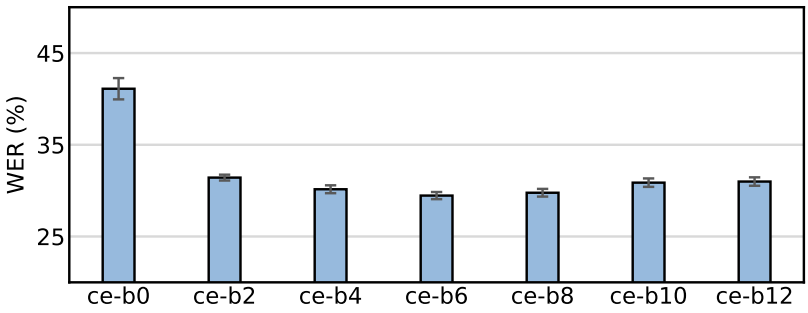


Figure 4.1: Performance of visual speech recognition on LRS2 test set based on features extracted from different layers. "ce-b0" to "ce-b12" refer to the layers from each conformer block from bottom to top.

LRW dataset for initialisation and we fine-tune it on the corresponding training sets of the LRS2 or LRS3 datasets (tenth row of Table 4.8 and seventh row of Table 4.9). Finally, we use the models trained on LRW + LRS3 and LRW + LRS2 as initialisation and fine-tune them further on LRS2 and LRS3, respectively (eleventh row of Table 4.8 and eigth row of Table 4.9). It is clear as we use more datasets for training the performance keeps improving. This is also the case for Spanish and Mandarin (seventh row of Table 4.10 and fourth row of Table 4.12) even when models trained on English are used for initialisation. However, the reduction in WER is smaller than in English probably due to language mismatch.

Finally, we use a subset of the AVspeech dataset as additional training data together with the automatically generated English transcriptions. Again, the WER is reduced in all languages (twelfth row of Table 4.8, ninth row of Table 4.9, last row of Table 4.10 and Table 4.12), despite using transcriptions which contain errors, with the smallest reduction observed in Mandarin. This is not surprising since Mandarin is much less similar to English than Spanish. These results are in line with the hypothesis that the reduction in the WER reported in recent works is mainly due to the larger datasets used for training.

4.1.6 Curriculum Learning

The end-to-end model was trained from scratch, resulting in poor performance on LRS2 and LRS3. This is likely due to the vast amount of very long utterances featured in LRS2 and LRS3, which makes learning from scratch especially challenging. We have found that the issue can be resolved by progressively training the end-to-end model, starting with short utterances and then using longer ones during training. This approach is commonly called curriculum learning (CL). In this paper, the model is initially trained with a subset of labelled training data, consisting of videos shorter than 100 frames. Then this model is

Table 4.3: Results of curriculum learning experiments on the LRS2 dataset.

Video length in frames	WER on the validation set	WER on the test set
Baseline VSR model		
0-100	65.1±0.2	52.7±0.8
0-150	54.0±0.7	44.2±0.5
0-300	46.0±0.6	36.3±0.4
0-450	43.6±0.5	34.3±0.5
0-600	42.4±0.4	33.7±0.4
VSR model with auxiliary workers		
0-100	51.9±0.3	41.5±0.5
0-150	46.2±0.4	36.1±0.3
0-300	43.3±0.2	34.4±0.2
0-450	42.6±0.3	34.6±0.5
0-600	42.0±0.3	33.4±0.3

used for initialisation when using utterances with up to 150 frames for training. This process is repeated for 3 more rounds where the length of training sequences is 300, 450, and 600 frames, respectively. As opposed to a model trained from scratch, this initialisation allows for faster training and more predictable results. In that way, we adopt this curriculum-based learning strategy, resulting in a significantly more efficient training process. As a result, as shown in the ninth row of Table 4.8 , the strategy reaches a WER of 33.65 ± 0.35, which pushes the state-of-the-art performance on LRS2, using the LRS2 dataset only. Similar pattern on LRS3 can also be found in the eighth row of Table 4.9. Results for each round of curriculum learning can be seen in Tables 4.3 and 4.4.

4.1.7 Time Masking

Data augmentation works by synthesising additional distorted training data with the goal of reducing over-fitting. In visual speech recognition, most existing works employ image transformations such as random cropping and horizontal flipping [161, 151, 24]. These spatial augmentations are helpful but they do not take into account the temporal nature of visual speech. Only few works exist which apply temporal augmentations like deleting or duplicating frames [16] or variable length augmentation [115].

In this work we propose the use of time masking which is commonly used in training ASR models [162].

Table 4.4: Results of curriculum learning experiments on the LRS3 dataset.

Video length in frames	WER on the test set
Baseline VSR model	
0-100	75.2±0.4
0-150	53.3±0.7
0-300	43.0±0.4
0-450	39.9±0.6
0-600	38.7±0.5
VSR model with auxiliary workers	
0-100	57.7±0.4
0-150	46.8±0.1
0-300	40.8±0.6
0-450	39.7±0.4
0-600	38.6±0.4

Table 4.5: Ablation study on the LRS2 dataset and LRS3 dataset. Models are trained on LRW+LRS2 and LRW+LRS3, respectively.

Method	WER on LRS2	WER on LRS3
Our model	29.5±0.4	35.8±0.5
- Audio auxiliary task	31.4±0.3	36.6±0.3
- Visual auxiliary task	30.6±0.5	36.9±0.5
- Audio auxiliary task, visual auxiliary task	33.2±0.5	37.8±0.6
- Time masking	32.6±0.5	38.5±0.5
- Audio auxiliary task, visual auxiliary task, time masking	35.0±0.5	39.1±0.4

It works by randomly masking n consecutive frames by replacing them with the mean sequence frame. This allows the model to more effectively use contextual information and can better disambiguate similar lip movements which correspond to different phonemes. It also makes the model more robust to short missing segments. Given that there is large variance in the video lengths, especially on the LRS2 and LRS3 datasets, the number of masks used is proportional to the length of the training sequence. Specifically, we use one mask per second, and for each mask, we randomly mask up to 40% of frames, where the masked segments is chosen using a uniform distribution. The impact of time masking is shown in the ablation study on the LRS2 and LRS3 datasets shown in Table 4.5. Training a model without time masking results

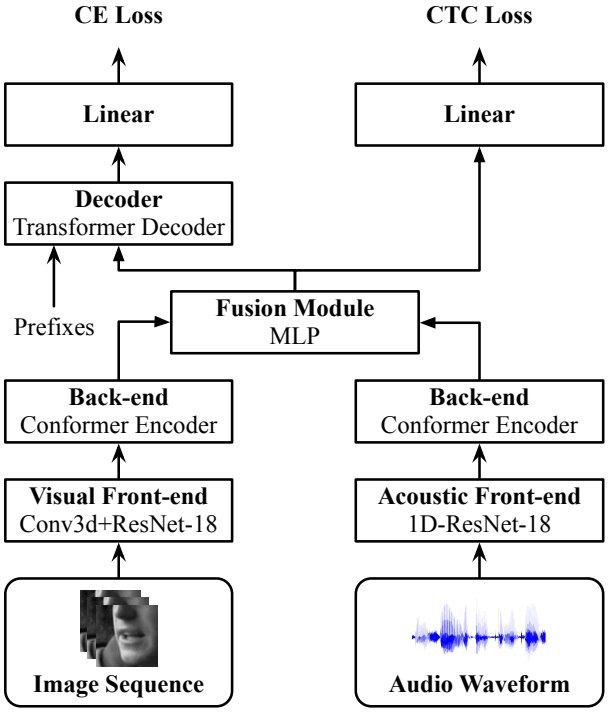


Figure 4.2: End-to-end audio-visual speech recognition architecture. The inputs are pixels and raw audio waveforms.

in a significant increase in the mean WER when compared to the full model.

4.1.8 Audio-Visual Fusion

We present a hybrid CTC/Attention model based on a ResNet-18 and Convolution-augmented transformer (Conformer), that can be trained in an end-to-end manner. In particular, features directly from the audio and visual encoders given raw pixels and audio waveforms are fed to a MLP for fusion. As shown in Figure 4.2, the acoustic and visual features from the back-end modules are then concatenated and projected to d_k-dimensional space by an MLP. The MLP is composed of a linear layer with an output size of $4 \times d_k$ followed by a batch normalization layer, ReLU, and a final linear layer with an output dimension d_k.

4.2 Experimental Setup

4.2.1 Performance Metrics

Word Error Rate (WER) is the most common metric used in speech recognition, which measures how close the predicted word sequence is to the target word sequence. Assuming S is the number of substitutions, D

Table 4.6: Investigation of the impact of hyperparameters and Language Model (LM) choices on the LRS2 dataset and LRS3 dataset.

Method	WER on LRS2	WER on LRS3
CM-seq2seq [151] - Baseline	37.8±0.5	44.9±0.8
+ Hyperparameter Optimisation	35.9±0.5	40.6±0.8
+ Improved LM	35.0±0.5	39.1±0.4

is the number of deletions, I is the number of insertions needed to get from the predicted to the target sequence and N is the number of words in the target sequence, then the metric can be defined as follows:

$$WER = \frac{S + D + I}{N} \tag{4.6}$$

Similarly to WER, we can define the character error rate (CER) which measures how close the predicted and target character sequences are. In this case, S, D, and I are computed at the character level and N is the total number of characters.

4.2.2 Preprocessing

We use the RetinaFace [135] face detector and the Face Alignment Network (FAN) [136] to detect 68 facial landmarks. Then, the faces are registered to a neutral reference frame using a similarity transformation to remove translation and scaling variations. A bounding box of 96×96, centered on the mouth center, is used to crop the mouth Region Of Interest (ROI). The cropped patch is further converted to gray-scale and normalised with respect to the overall mean and variance of the training set.

4.2.3 Hyper-Parameter Optimisation

Hyper-parameter optimisation aims at improving the performance of a model by fine-tuning the values of parameters which are used to control the training process or the model architecture. Some of the most common hyper-parameters which are usually optimised are the following: initial learning rate, learning rate decay parameters, number of layers, size of layers, dropout rate and the loss function weights which are used to combine the different loss terms. Additional hyper-parameters related to conformers are the number and size of the self attention heads. We performed hyper-parameter optimisation on the LRS2

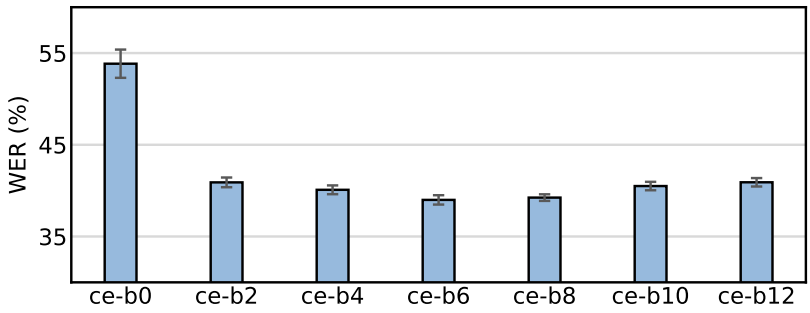


Figure 4.3: Performance of visual speech recognition on LRS2 validation set based on features extracted from different layers. "ce-b0" to "ce-b12" refer to the layers from each conformer block from bottom to top.

Table 4.7: Investigation of the impact of hyperparameters and Language Model (LM) choices on the validation set of LRS2 dataset.

Method	WER
CM-seq2seq [151] - Baseline	47.7±0.5
+ Hyperparameter Optimisation	45.6±0.4
+ Improved LM	44.1±0.5

dataset by attempting to reduce the WER on the validation set. Our conclusion was that the parameters used in the baseline model [151] were already optimal so no further improvement was observed.

The next step was to optimise other hyper-parameters which might not have been exhaustively optimised, like batch size related parameters. Each hyper-parameter is optimised independently based on the WER on the validation set of LRS2. We use the same hyper-parameters for all experiments. The main hyperparameter that was found to have a significant impact on performance was the batch size. We observed that increasing the batch size from 8 to 16 led to reduced WER on the validation set of the LRS2 dataset (see Table 4.7). There is also one more hyper-parameter which controls the batch size based on the length of the sequences. In other words, if some sequences are too long then the batch is halved. We found that increasing this threshold from 150 to 220 frames also improved the performance. We could not increase these two hyper-parameters even further due to GPU memory constraints but it is likely that the WER will be reduced even more.

4.2.4 Improving Language Models

A language model (LM) determines the probability of a given sequence of characters. It is used during decoding and favours sequences which are more likely to occur. In order to increase the capacity of the LM we use multiple text corpora for training. We also increase the number of sequences considered during decoding (beam size is set to 40). The impact of these changes can be seen in Table 4.6 where the WER is reduced for both English datasets.

The score from the language model (S_{LM}) is incorporated in decoding as shown in Eq. 4.7.

$$S = \lambda S_{CTC} + (1 - \lambda) S_{att} + \beta S_{LM} \tag{4.7}$$

where S_{CTC} and S_{att} are the scores of the CTC and decoder branch, respectively. λ and β correspond to the CTC and language models score weights.

4.2.5 Language Models

We train six monolingual transformer-based language model [163] for 50 epochs. The English language model is trained by combining the training sets of LibriSpeech (960 h) [164], pre-training and training sets of LRS2 [19] and LRS3 [52], TED-LIUM 3 [165], Voxforge (English) and Common Voice (English) [166], with a total of 166 million characters. The Mandarin language model is trained by combining the CMLR [53] and news2016zh, with a total of 153 million characters. The Spanish language model is trained by combining the Spanish corpus from Multilingual TEDx [55], Common Voice [166] and Multilingual LibriSpeech [167], with a total of 192 million characters. The Italian language model is trained by combining the Italian corpus from Multilingual TEDx [55], Common Voice [166] and Multilingual LibriSpeech [167], with a total of 252 million characters. The Portuguese language model is trained by combining the Portuguese corpus from Multilingual TEDx [55], Common Voice [166] and Multilingual LibriSpeech [167], with a total of 85 million characters. The French language model is trained by combining the French corpus from Multilingual TEDx [55], Common Voice [166] and Multilingual LibriSpeech [167], with a total of 945 million characters. In our work, we set λ and β from equation 4.1 to 0.1 and {English: 0.6, Mandarin: 0.3, Spanish: 0.4, Italian: 0.5, Portuguese: 0.3, French: 0.3}, respectively. The impact of the improved English language model on the validation set of the LRS2

dataset can be seen on Table 4.7.

4.2.6 Implementation

Our experiments were implemented using an open-source toolkit, ESPNet [168]. We train the models with the Adam optimizer [169] with $\beta_1 = 0.9, \beta_2 = 0.98$ and $\epsilon = 10^{-9}$. The learning rate is increased linearly in the first 25 000 steps, yielding a peak learning rate of 0.0004 and thereafter decreases proportionally to the inverse square root of the step number. The network is trained for 50 epochs with a batch size of 16. We use the model averaged over the last 10 checkpoints for evaluation. Details regarding the network architecture are provided in section 4.1.2 in the Supplementary Information.

4.2.7 Baseline VSR Model

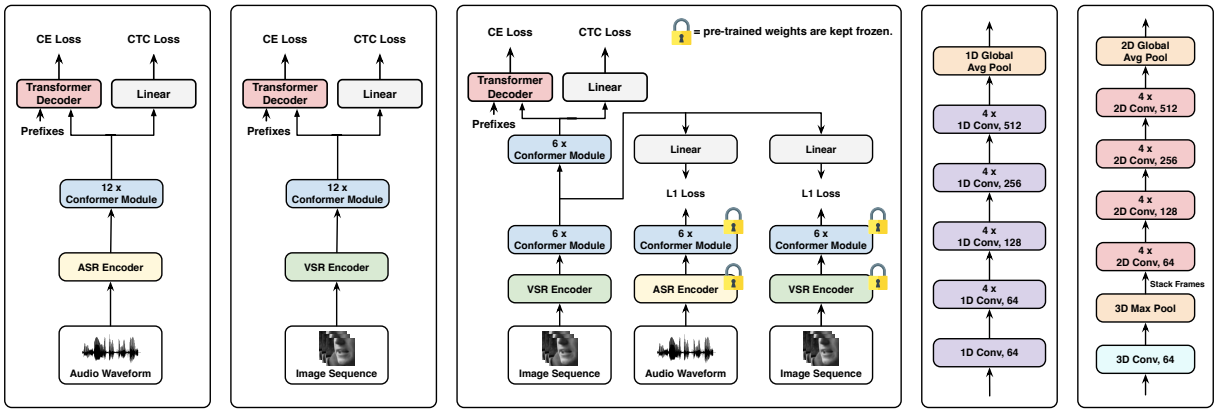
The baseline VSR model that we extend in this work is based on [151]. The model consists of a 3D convolutional layer with a receptive field of 5 frames, followed by a 2D ResNet-18 (Fig. 4.4e), a 12-layer Conformer model [112] and a transformer decoder [67] as shown in Fig. 4.4b. The model is trained end-to-end using a combination of the Connectionist Temporal Classification (CTC) loss with an attention mechanism. Data augmentation is also used during training in the form of random cropping and image flipping (applied to all frames in the same sequence). This model achieves the state-of-the-art VSR performance on the LRS2 and LRS3 datasets, when only publicly available data are used for training.

4.2.8 Baseline ASR Model

The baseline ASR model that we use is based on [151]. The model consists of an 1D ResNet-18 (Fig. 4.4d), a 12-layer Conformer model and a transformer decoder as shown in Fig. 4.4a. This model also follows the hybrid CTC/Attention architecture and is trained end-to-end. Time-masking is also used as data augmentation during training. This is at the moment the state-of-the-art ASR model on the LRS2 and LRS3 datasets.

4.3 Results

We present a model that takes a sequence of images as input and predicts the spoken words in that sequence. We show that the proposed method outperforms state-of-the-art methods trained on publicly available data



(a) (b) (c) (d) (e)

Figure 4.4: (a) Baseline ASR model, (b) Baseline VSR model, (c) Proposed model with prediction-based auxilliary tasks. The frame rate of extracted visual features and audio features is 25. (d) Architecture of ASR encoder. (e) Architecture of VSR encoder.

by a large margin. It also outperforms some methods which have been trained on non-publicly available datasets (which are an order of magnitude larger than the datasets we use in this work). Finally, we also show that our method does not work well only for English but it also significantly outperforms the state-of-the-art VSR methods for Mandarin, Spanish, Italian, French and Portuguese.

4.3.1 Results on LRS2

Results on LRS2, which is an English audio-visual dataset, are reported in Table 4.8. Our model outperforms all existing works by a large margin even when it is trained on smaller amounts of training data. In particular, we outperform the previous state-of-the-art [151], in terms of the best WER achieved, by 5 %. This is despite the fact that [151] is trained on a larger training set. When we use the same training set size as [151] our model results in a 9.2 % improvement. Finally, when we use additional training data an even larger improvement of 12.4 % is observed. Similarly, our approach results in a 22.8 % absolute improvement in the best WER over [51] which uses a training set with similar size to ours and also includes non-publicly available data.

4.3.2 Results on LRS3

Results on LRS3, which is an English audio-visual dataset, are presented in Table 4.9. Also in this case, our proposed approach significantly outperforms all existing works which are trained using publicly available datasets. In particular, our method leads to an 8.2 % absolute improvement, in terms of the best

Table 4.8: Results on the LRS2 dataset.

Method	Pre-training Set	Training Set	Training Sets Total Size (hours)	Mean±Std.	Best
Using Publicly Available Datasets					
MV-WAS [19]	-	LRS2	223	-	70.4
CTC/Att. [24]	LRW	LRS2	380	-	63.5
KD + CTC [59]	VoxCeleb2^{clean}+LRS3	LRS2	995	-	51.3
KD-seq2seq [170]	LRW+LRS3	LRS2	818	-	49.2
TDNN [171]	-	LRS2	223	-	48.9
CM-seq2seq [151]	LRW	LRS2	380	-	37.9
Ours	-	LRS2	223	**33.6±0.5**	**32.9**
Ours	LRW	LRS2	380	**29.5±0.4**	**28.7**
Ours	LRW+LRS3	LRS2	818	**27.6±0.2**	**27.3**
Ours	LRW+LRS3+AVSpeech	LRS2	1 459	**25.8±0.4**	**25.5**
Using Non-Publicly Available Datasets					
TM-seq2seq [51]	MVLRS+LRS3	LRS2	1 391	-	48.3

Table 4.9: Results on the LRS3 dataset.

Method	Pre-training Set	Training Set	Training Sets Total Size (hours)	Mean±Std.	Best
Using Publicly Available Datasets					
KD+CTC [59]	VoxCeleb2^{clean}	LRS3	772	-	59.8
KD-seq2seq [170]	LRW+LRS2	LRS3	818	-	59.0
CM-seq2seq [151]	LRW	LRS3	595	-	43.3
Ours	-	LRS3	438	**38.6±0.4**	**37.9**
Ours	LRW	LRS3	595	**35.8±0.5**	**35.1**
Ours	LRW+LRS2	LRS3	818	**34.9±0.2**	**34.7**
Ours	LRW+LRS2+AVSpeech	LRS3	1 459	**32.1±0.3**	**31.5**
Using Non-Publicly Available Datasets					
TM-seq2seq [51]	MVLRS+LRS2	LRS3	1 391	-	58.9
V2P [32]	-	LSVSR	3 886	-	55.1
RNN-T [33]	-	YT-31k	31 000	-	33.6
ViT3D-TM [34]	-	YT-90k	90 000	-	25.9
ViT3D-CM [172]	-	YT-90k	90 000	-	19.3

Table 4.10: Results on the CMLR dataset.

Method	Pre-training Set	Training Set	Training Sets Total Size (hours)	Mean±Std.	Best
LipCH-Net [149]	-	CMLR	61		34.0
CSSMCM [53]	-	CMLR	61	-	32.5
LIBS [173]	-	CMLR	61	-	31.3
CTCH [150]	-	CMLR	61	-	22.0
Ours	-	CMLR	61	**9.1±0.05**	**9.1**
Ours	LRW+LRS2+LRS3	CMLR	879	**8.2±0.06**	**8.1**
Ours	LRW+LRS2+LRS3+AVSpeech	CMLR	1 520	**8.1±0.05**	**8.0**

WER, over the state-of-the-art [151] when the same training data are used. As expected, a smaller absolute improvement of 5.4 % is reported when a smaller training set is used. In case of additional training data being available, a larger absolute improvement of 11.8 % is achieved.

There are also some works which rely on very large non-publicly available datasets for training. As a consequence, it is not clear if the reported improvement in WER is due to a better model or simply to the large amount of training data. Our approach outperforms all works which use up to 21 times more training data. More specifically, our best model, trained on 1 453 hours of video, leads to a 2.1 % absolute improvement over [33] which uses 31 000 hours of training data. However, it performs worse than [34] that presents a model trained on 90 000 hours, which is 62 times more training data than the publicly available training data our model is trained on.

4.3.3 Results on CMLR

Results on the CMLR dataset, which is a Mandarin audio-visual dataset, are shown in Table 4.10. We report performance in terms of character error rate (CER) instead of WER because Chinese characters are not separated by spaces. Our approach results in a significant reduction in the CER over all existing works. We achieve an absolute improvement of 12.9 % over the state-of-the-art [150]. The WER can be further reduced by 1.1 % by first pre-training our model on English and then fine-tuning it on the CMLR training set.

Table 4.11: Results on the Multilingual TEDx-Spanish (MT_{es}) dataset.

Method	Pre-training Set	Training Set	Training Sets Total Size (hours)	Mean±Std.	Best
CM-seq2seq [151]	LRW	CM_{es}+MT_{es}	244	66.4±0.8	65.2
Ours	LRW	CM_{es}+MT_{es}	244	**60.8±0.8**	**60.3**
Ours	LRW+LRS2+LRS3	CM_{es}+MT_{es}	905	**56.9±0.5**	**56.5**
Ours	LRW+LRS2+LRS3+AVSpeech	CM_{es}+MT_{es}	1 546	**56.6±0.3**	**56.3**

Table 4.12: Results on the CMU-MOSEAS-Spanish (CM_{es}) dataset.

Method	Pre-training Set	Training Set	Training Sets Total Size (hours)	Mean±Std.	Best
CM-seq2seq [151]	LRW	CM_{es}+MT_{es}	244	58.9±0.8	58.1
Ours	LRW	CM_{es}+MT_{es}	244	**51.5±0.8**	**50.4**
Ours	LRW+LRS2+LRS3	CM_{es}+MT_{es}	905	**47.4±0.2**	**47.2**
Ours	LRW+LRS2+LRS3+AVSpeech	CM_{es}+MT_{es}	1 546	**44.6±0.6**	**43.9**

4.3.4 Results on Spanish

Results on the Multilingual TEDx-Spanish dataset are shown in Table 4.11. We observe that our proposed approach results in a 5.6 % absolute reduction in the WER. A further reduction of 4.2 % can be achieved by using additional training data.

Results on the CMU-MOSEAS-Spanish dataset, which is an audio-visual Spanish dataset, are shown in Table 4.12. Given that this is a small dataset it is not possible to train an accurate model without using additional data. For this purpose, we first pre-train the model on English datasets and then fine-tune it on the training sets of CMU-MOSEAS and TEDx datasets using the Spanish videos only. Since this is a new dataset and there are no results from prior works, we have trained the end-to-end model presented in [151] to serve as the baseline. We observe that our proposed approach results in a 7.7 % absolute reduction in the WER. A further reduction of 6.5 % can be achieved by using additional training data.

4.3.5 Results on Italian

We manually clean the Italian corpus on Multilingual TEDx to exclude videos without visible speakers, resulting in a total of 26387 videos (45.8 hours) for training, 252 videos (0.4 hours) for validation and 309

Table 4.13: Results on the Multilingual TEDx-Italian (MT_{it}) dataset.

Method	Pre-training Set	Training Set	Training Sets Total Size (hours)	Mean±Std.	Best
CM-seq2seq [151]	LRW	MT_{it}	203	71.5±0.4	70.9
Ours	LRW	MT_{it}	203	**65.9±0.5**	**65.2**
Ours	LRW+LRS2+LRS3	MT_{it}	864	**58.7±0.3**	**58.2**
Ours	LRW+LRS2+LRS3+AVSpeech	MT_{it}	1 505	**57.9±0.7**	**57.4**

Table 4.14: Results on the Multilingual TEDx-Portuguese (MT_{pt}) dataset.

Method	Pre-training Set	Training Set	Training Sets Total Size (hours)	Mean±Std.	Best
CM-seq2seq [151]	LRW	CM_{pt}+MT_{pt}	256	70.2±0.3	69.7
Ours	LRW	CM_{pt}+MT_{pt}	256	**66.0±0.5**	**65.3**
Ours	LRW+LRS2+LRS3	CM_{pt}+MT_{pt}	917	**62.4±0.4**	**62.0**
Ours	LRW+LRS2+LRS3+AVSpeech	CM_{pt}+MT_{pt}	1 558	**62.1±0.6**	**61.5**

Table 4.15: Results on the CMU-MOSEAS-Portuguese (CM_{pt}) dataset.

Method	Pre-training Set	Training Set	Training Sets Total Size (hours)	Mean±Std.	Best
CM-seq2seq [151]	LRW	CM_{pt}+MT_{pt}	256	65.7±0.5	65.4
Ours	LRW	CM_{pt}+MT_{pt}	256	**57.2±0.7**	**56.6**
Ours	LRW+LRS2+LRS3	CM_{pt}+MT_{pt}	917	**53.1±0.2**	**52.8**
Ours	LRW+LRS2+LRS3+AVSpeech	CM_{pt}+MT_{pt}	1 558	**51.6±0.2**	**51.4**

videos (0.5 hours) for testing. Results on the Multilingual TEDx-Italian dataset are shown in Table 4.13. Our proposed approach results in an absolute drop of 5.6 % in the WER. A further reduction of 8 % can be achieved by using additional training data.

4.3.6 Results on Portuguese

We manually cleaned the Portuguese corpus on Multilingual TEDx to exclude videos where the speaker is not visible, resulting in a total of 52 395 videos (81.3 hours) for training, 532 videos (0.7 hours) for validation and 401 videos (0.6 hours) for testing. Results on the Multilingual TEDx-Portuguese dataset are shown in Table 4.14. We observe that our proposed approach results in a 4.2 % absolute reduction in

Table 4.16: Results on the Multilingual TEDx-French (MT_{fr}) dataset.

Method	Pre-training Set	Training Set	Training Sets Total Size (hours)	Mean±Std.	Best
CM-seq2seq [151]	LRW	CM_{fr}+MT_{fr}	257	84.0±0.7	83.2
Ours	LRW	CM_{fr}+MT_{fr}	257	**74.6±0.6**	**73.4**
Ours	LRW+LRS2+LRS3	CM_{fr}+MT_{fr}	918	**67.0±0.3**	**66.7**
Ours	LRW+LRS2+LRS3+AVSpeech	CM_{fr}+MT_{fr}	1 559	**67.0±0.6**	**66.2**

Table 4.17: Results on the CMU-MOSEAS-French (CM_{fr}) dataset.

Method	Pre-training Set	Training Set	Training Sets Total Size (hours)	Mean±Std.	Best
CM-seq2seq [151]	LRW	CM_{fr}+MT_{fr}	257	79.9±0.4	79.6
Ours	LRW	CM_{fr}+MT_{fr}	257	**68.4±0.5**	**67.5**
Ours	LRW+LRS2+LRS3	CM_{fr}+MT_{fr}	918	**60.1±0.3**	**59.5**
Ours	LRW+LRS2+LRS3+AVSpeech	CM_{fr}+MT_{fr}	1 559	**59.1±0.5**	**58.3**

the WER. A further reduction of 3.9 % can be achieved by using additional training data.

We divide the Portuguese corpus on CMU-MOSEAS [54] into 10 658 videos (17.8 hours) for training and 412 videos (0.7 hours) for testing, respectively. Results on the CMU-MOSEAS-Portuguese dataset are shown in Table 4.15. The proposed approach results in a 8.5 % absolute reduction in the WER. Using additional training data leads to a further reduction of 5.6 %.

4.3.7 Results on French

We manually cleaned the French corpus on Multilingual TEDx to exclude videos where the speaker is not visible, resulting in a total of 58 809 videos (84.9 hours) for training, 333 videos (0.4 hours) for validation and 235 videos (0.3 hours) for testing. Results on the Multilingual TEDx-French dataset are shown in Table 4.16. The proposed approach results in a 9.4 % absolute reduction in the WER. A further reduction of 7.6 % can be achieved by using additional training data.

We divide the French corpus on CMU-MOSEAS [54] into 8 880 videos (15.3 hours) for training and 513 videos (0.8 hours) for testing, respectively. Results on the CMU-MOSEAS-French dataset are shown in Table 4.17. We observe that our proposed approach results in a 11.5 % absolute reduction in the WER.

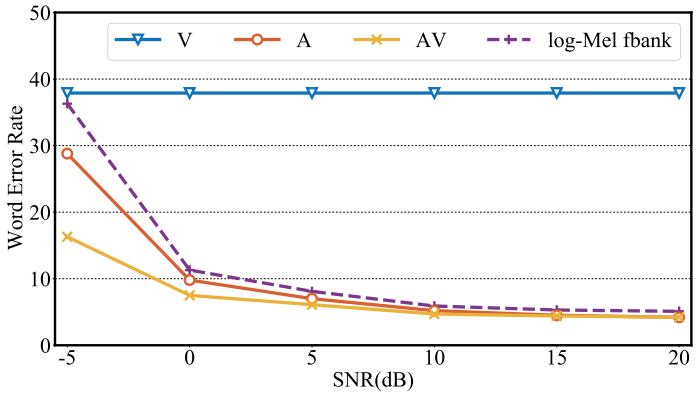


Figure 4.5: Word Error Rate (WER) as a function of the noise level. A: End-to-End audio model. V: End-to-End visual model, AV: End-to-End audio-visual model. log-Mel filter-bank: A conformer model trained with log-Mel filter-bank features.

Furthermore, as expected, the performance is improved by a large margin of 9.3 % when additional training data is included.

4.3.8 Comparison between Mean and Best WER/CER

In all results shown in Tables 4.8 to 4.12 we report both the mean and the best performance over 10 runs. We observe that the mean WER, which is more representative of the actual performance, is up to 0.8 % worse than the best WER. The only exception is the CMLR dataset (Table 4.10) where the mean and best CER are practically the same, mainly due to the large test set. This difference between the mean and best WER is something which should be taken into account when comparing different models, especially when the models are tested on relatively small test sets and the results are too close.

4.3.9 Audio-Visual Experiments

As shown in Table 4.18, the E2E audio-only model using audio waveforms for training achieves a WER of 4.3 %, resulting in an absolute improvement of 2.4 %. over the current state-of-the-art. For comparison purposes, we also run an experiment using 80-dimension log-Mel filter-bank features following [24, 160]. Similarly to the WavAugment [175], we augment the log-Mel filter-bank features via SpecAugment [162]. By replacing the raw audio features with the log-Mel filter-bank features, we observe the same performance, WER 4.3 %, which indicates deep acoustic speech representations based on the proposed temporal network

[†] We used a part of unlabelled AVSpeech dataset with machine-generated transcriptions for training. Details can be found in supplementary materials

Method	Training Data (Hours)	WER
Audio-only (↓)		
TM-seq2seq [51]	MVLRS (730) + LRS2&3$^{v0.4}$ (632)	9.7
CTC/Attention [24]	LRS2 (224)	8.3
CTC/Attention [174]	LibriSpeech (960) + LRS2 (224)	8.2
TDNN [171]	LRS2 (224)	6.7
Ours (filter-bank)	LRS2 (224)	**4.3**
Ours (raw A)	LRS2 (224)	**4.3**
Ours (raw A)	LRW (157) + LRS2 (224)	**3.9**
Audio-visual (↓)		
TM-seq2seq [51]	MVLRS (730) + LRS2&3$^{v0.4}$ (632)	8.5
CTC/Attention [24]	LRW (157) + LRS2 (224)	7.0
TDNN [171]	LRS2 (224)	5.9
Ours (raw A + V)	LRS2 (224)	**4.2**
Ours (raw A + V)	LRW (157) + LRS2 (224)	**3.7**

Table 4.18: Word Error Rate (WER) of the audio-only and audio-visual models on LRS2.

can be directly learnt from audio waveforms. To better investigate their differences, we conduct noisy experiments varying different levels of babble noise. The results are shown in Figure 4.5. It is interesting to observe that the performance of the raw audio model slightly outperforms the log-Mel filter-bank based over varying levels of babble noise with a maximum absolute margin of 7.5 % at -5 dB. This indicates deep speech representations are more robust to noise than the log-Mel filter-bank features. We further initialise the audio encoder with a model pre-trained on LRW then the WER drops to 3.9 %.

It is evident that the audio-visual model which directly learns from audio waveforms and raw pixels leads to a small improvement over the audio-only models. We also run audio-only, visual-only, and audio-visual experiments varying the SNR levels of babble noise. The results are shown in Fig 4.5. Note that both audio-only and audio-visual models are augmented with noise injection. It is clear that the audio-visual model achieves better performance than the audio-only model. The gap between raw audio-only and audio-visual models becomes larger by the presence of strong noise. This demonstrates that the audio-visual model is particularly beneficial when the audio modality is heavily corrupted by background noise.

4.4 Conclusion

In this chapter, we study the audio-visual speech recognition models for continuous speech. We presented a hybrid CTC/Attention architecture and performed end-to-end training. The architecture uses a CTC loss in combination with an attention-based model in order to force monotonic alignments and at the same time get rid of the conditional independence assumption. Furthermore, we present our approach for VSR and demonstrated that state-of-the-art performance can be achieved not only by using larger datasets, which is the current trend in the literature, but also by carefully designing a model. We demonstrate that the AVSR model significantly outperforms the ASR model, especially at high levels of noise. Additionally, we show that the model trained using raw audio can achieve better performance than the model trained using log-Mel filter-bank features by leveraging the proposed ResNet-18 1D backbone. Our work opens a new path for in-the-wild audio-visual speech recognition and demonstrates the potential of learning from raw streams, which is in contrast to the log-Mel filter-bank features used in modern ASR systems. We hope that our work can function as the foundation for future research. In the next chapter, we will study the problem of self-supervised cross-modal learning.

Chapter 5

Learning Visual Speech Representations from Audio

Contents

5.1 Methodology **90**

5.2 Experimental Setup **91**

5.3 Results **94**

5.4 Conclusion **97**

In Chapter 3 and 4, we have discussed the problem of AVSR for both isolated words and continuous speech. Apart from audio-visual fusion in audio-visual learning, the correlation nature in audio-visual pairs makes it possible to supervise each other to learn powerful representations. In this chapter, we study the problem of self-supervised audio-visual learning.

Self-supervised learning aims to leverage unlabelled data by extracting the training objective directly from the input itself, in an attempt to model meaningful representations of the proposed modality which capture its content and structure. In works adopting this methodology, this task is usually known as the "pretext task" and this initial training procedure is known as the "pre-training" stage. After pre-training, the network is trained on the "downstream task", which generally involves a smaller set of manually labelled data.

Pretext tasks for visual self-supervision include image colourisation [176], jigsaw puzzle solving [177], as

well as combinations of these and other tasks [178]. Self-supervised learning has also been explored in the speech community through works such as Contrastive Predicting Coding (CPC) [179] and wav2vec [180], which predict/discriminate future segments of audio samples; LIM (Local Info Max) [181], which maximises mutual information for the same speaker; and, more recently, PASE (Problem Agnostic Speech Encoder) [182, 183], which predicts established audio features such as STFT and MFCC.

Self-supervision has been adopted in the audio-visual domain. Recent approaches include audio-visual fusion [184, 185], clustering [186], and distillation [187]; cross-modal discrimination [188]; and cyclic translation between modalities [189]. Shukla *et al.* [190] focus on learning audio representations by facial reconstruction from waveform speech. Conversely, [191] predict frequency-based summaries of ambient sound from video, while other recent works apply audio-visual synchronisation [192, 193, 194] to learn visual embeddings. A task that can benefit from self-supervised learning is lip-reading. Current state-of-the-art lip-reading models rely on annotating hundreds of hours of visual speech data [33], which is costly. To solve this issue, Afouras *et al.* [195] propose using a pre-trained ASR model to produce machine-generated captions for unsupervised pre-training. This provides automatically labelled data but still relies on an ASR model trained on large amounts of labelled data.

In this chapter, we aim to leverage the vast amount of available audio-visual speech data to learn generic visual speech features and improve state-of-the-art lip-reading models by predicting audio features from visual speech. The targeted audio features are extracted from waveform audio without the need for additional labels using an established speech encoder (PASE+ [183]). Using the proposed approach, the learnt visual features are explicitly guided by audio which contains rich information about speech. This in turn can lead to learning visual features which are more suitable for speech recognition. After this training procedure, we apply our model for lip-reading on a transcribed visual speech dataset. For both tasks, we employ a 2D ResNet-18 with a 3D front-end layer, as proposed in [31], followed by the recently proposed conformer encoder [112].

Our research contributions are as follows: **1)** We present LiRA, which learns powerful visual speech representations by predicting acoustic features from raw video taken from large audio-visual datasets. **2)** We demonstrate that LiRA provides a good initialisation for fine-tuning lip-reading models which consistently outperforms training from scratch, and that this method is particularly beneficial for smaller labelled datasets. **3)** We show that LiRA outperforms previous self-supervised methods for word-level

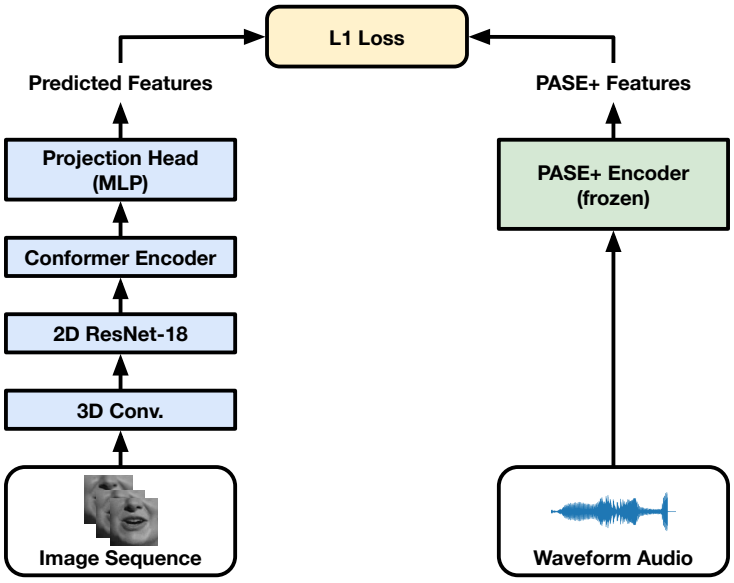


Figure 5.1: The high-level architecture of our model and our methodology for audio-visual self-supervised training.

lip-reading, achieving an accuracy of 88.1% on LRW by pre-training on unlabelled data. **4)** Finally, we leverage our self-supervised approach towards sentence-level lip-reading, and find that our fine-tuned model achieves state-of-the-art performance for LRS2.

5.1 Methodology

5.1.1 Pretext task

LiRA predicts PASE+ features from raw video and is composed of three distinct components. The first is the spatial encoder, which is a traditional 2D ResNet-18 preceded by a 3D front-end layer. The second component is the temporal encoder – the conformer – which receives as input the frame-wise features produced by the spatial encoder and returns a set of features of the same size. The conformer encoder combines traditional attention-based transformer blocks, which excel at capturing global temporal dependencies, with convolutional layers, which model local patterns efficiently [112]. The final component is the projection head (based on the MLP – Multi-Layer Perceptron – workers presented in [182]), which projects these representations into the predicted PASE+ features. To train the model, we apply an L1 loss between the generated embeddings and the features extracted from the pre-trained (frozen) PASE+ model, as shown in Figure 5.1. We would also like to mention that we have also experimented with predicting

MFCC features but the results were worse than predicting PASE+ features.

5.1.2 Downstream Task

To evaluate the visual speech representations, we run three variations of end-to-end lip-reading experiments. The training procedure is illustrated in Figure 5.2. LiRA-Supervised models are trained from scratch based on the same encoder as in the self-supervised training [196]. This serves as our baseline model since it is trained only with the labelled training data. LiRA-Frozen models are trained using LiRA features from the pre-trained encoder. This allows us to evaluate the visual representations learned during self-supervised learning. Finally, LiRA-FineTuned models use the same model as LiRA-Supervised but are initialised with the pre-trained encoder weights from the pretext task. By using this configuration, we can evaluate the model initialisation capabilities of the proposed self-supervised learning approach. For each of these methods, we adopt a separate model for each lip-reading task - six models in total. For word-level lip-reading, we use a Multi-Scale Temporal Convolutional Network (MS-TCN) [3] on top of the encoder, followed by a linear classifier for classification. For sentence-level lip-reading, we follow the state-of-the-art lip-reading model [196] on LRS2 and build a hybrid CTC/attention model. We use the same conformer encoder architecture as in the pre-training phase, followed by the transformer decoder for sequence-to-sequence training [197]. We also perform fine-tuning experiments using the pre-trained model.

5.2 Experimental Setup

5.2.1 Training Settings in the Pretext Task

The 3D front-end module preceding our ResNet consists of a convolutional layer with kernel size (5, 7, 7) followed by a max pooling layer. The conformer, on the other hand, is comprised of an initial embedding module – feed forward layer combined with layer normalisation, dropout (0.1), activation (ReLU – Rectified Linear Unit) and relative positional encoding (as proposed in [198]) – followed by 12 conformer blocks, as defined in [112]. The conformer blocks feature the following parameters: $d^{\text{ff}} = 2048$, $n^{\text{head}} = 4$, $d^{\text{q}} = 256$, $d^{\text{k}} = 256$, $d^{\text{v}} = 256$; where d^{ff} is the hidden dimension of the feed-forward modules, n^{head} is the number of self-attention heads, and d^{q}, d^{k}, d^{v} are the dimensions of the key (K), query (Q), and value (V) in the self-attention layers respectively. The MLP consists of a linear layer with a hidden

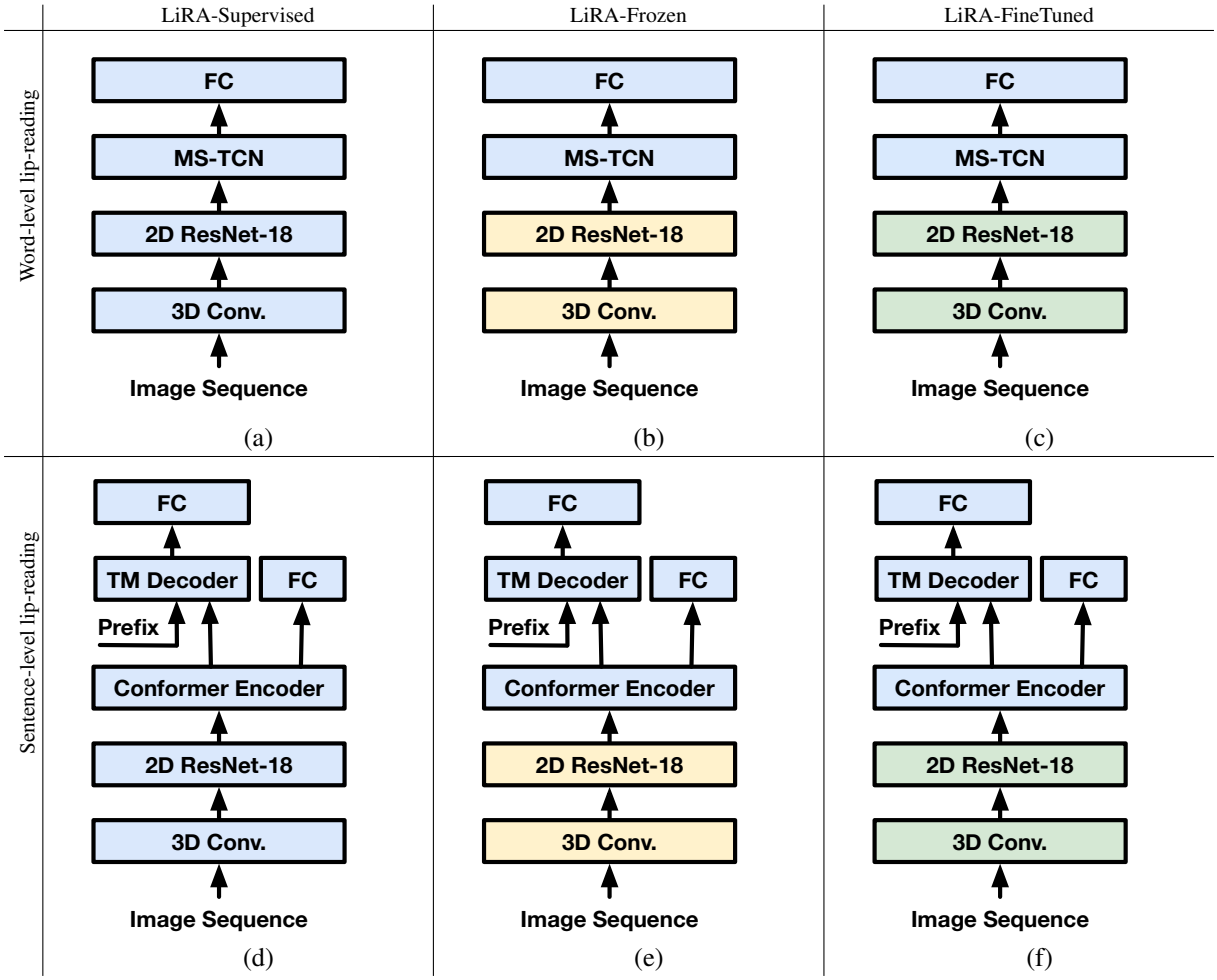


Figure 5.2: The variations of the end-to-end lip-reading architecture. The sub-figures in the top row ((a),(b),(c)) refer to the word-level lip-reading training procedures, while the sub-figures in the bottom row ((d),(e),(f)) refer to sentence-level lip-reading. From left to right, (a) and (d) denote training from scratch (the whole model is initialised randomly); (b) and (e) are feature extraction experiments based on visual features extracted from the pre-trained model; and (c) and (f) are fine-tuning experiments. Blue coloured blocks are trained from scratch on the downstream task; yellow coloured blocks are loaded from the pre-trained model and kept frozen during the downstream task; and green coloured blocks are loaded from the pre-trained model and are then fine-tuned for the downstream task. We abbreviate the following model layers: TM: Transformer, FC: Fully-Connected layer, MS-TCN: Multi-Scale Temporal Convolutional Network.

dimension of 256 units, ReLU activation, dropout, and a linear layer to project the representation to 256-dimensional latent space. For prediction, we average the PASE+ features, which are computed at 100 frames per second (fps), over time to match the frame rate of the input visual features (25 fps). We optimise our model using Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-9}$) combined with the Noam scheduler [197] (25 000 warm-up steps). The model is trained on LRS3 with a batch size of 32. For simplicity, we randomly sample 1 second from each clip and use it as the input to our network, discarding any utterances

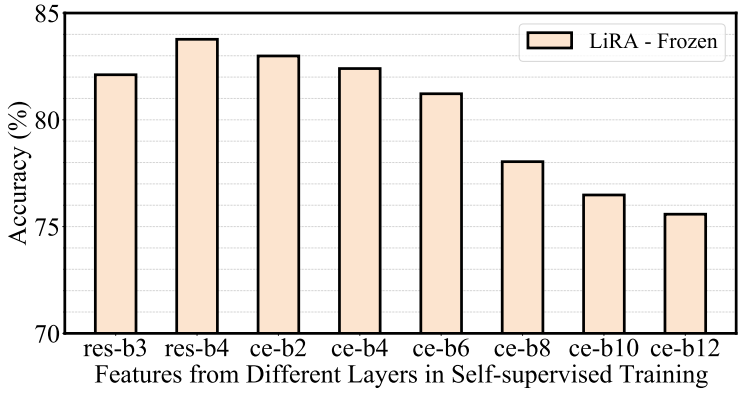


Figure 5.3: Accuracy of feature classification (LiRA-Frozen) on LRW based on features extracted from different layers after pre-training on LRS3 via self-supervision. "res-b3" and "res-b4" refer to the output of blocks 3 and 4 from the ResNet-18 respectively; and "ce-b2" to "ce-b12" refer to the layers from every two conformer blocks from bottom to top.

Table 1: A comparison of the performance between the baseline methods and ours (pre-trained on LRS3) on the LRW dataset.

Methods	Strategy	Acc. (%)
ResNet + BLSTM [31]	Supervised	83.0
Two-stream 3D CNN [199]	Supervised	84.1
ResNet + BLSTM [120]	Supervised	84.3
ResNet + DenseTCN [200]	Supervised	88.4
PerfectMatch [194]	Self-supervised	71.6
PT-CDDL [201]	Self-supervised	75.9
AV-PPC [202]	Self-supervised	84.8
LiRA-Supervised [115]	Supervised	87.4
LiRA-Frozen	Self-supervised	83.1
LiRA-FineTuned	Self-supervised	88.1

with less than 1 second in length.

5.2.2 Training Settings in Downstream Tasks

LiRA-Supervised In LiRA-Supervised, we train word-level (Figure 2a) and sentence-level lip-reading models (Figure 2d) from scratch. In particular, for the task of word-level lip-reading, we add a MS-TCN followed by a linear classifier with an output dimension of 500 on top of the encoder like [200]. A cross-entropy loss is employed to optimise the whole model using AdamW [139] with $\beta_1 = 0.9, \beta_2 = 0.999$, $\epsilon = 10^{-8}$ and a weight decay of 0.01 for 80 epochs with a batch size of 32. The initial learning rate is set

to 0.0003. For the task of sentence-level lip-reading, we use 12 multi-head attention blocks ($d^{\text{ff}} = 2048$, $n^{\text{head}} = 4$, $d^{\text{q}} = 256$, $d^{\text{k}} = 256$, $d^{\text{v}} = 256$) together with a linear layer on the top of conformer blocks like [196]. Following [160], we use a combination of CTC and cross-entropy loss to train a hybrid CTC/Attention architecture for 50 epochs with a batch size of 8. In this case, we use Adam with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$ with the first 25 000 steps for warm-up. The initial learning rate is set to 0.0004. At the decoding phase, we use a beam size of 20 for beam search. During decoding, we also apply a transformer-based language model trained on LRS2, LRS3, and Librispeech 960h [164] (16.2 million words in total). Due to graphic memory limitations, we exclude utterances with more than 600 frames during training.

LiRA-Frozen At the end of self-supervised training, the features extracted from the pre-trained frozen encoder are fed to a classifier for evaluation. For word-level lip-reading, we use a MS-TCN, followed by a linear layer with an output size of 500 for classification (Figure 2b). For the sentence-level lip-reading, the LiRA features are first fed to 12 conformer blocks, and then the encoded representations are used for CTC/attention joint training (Figure 2e).

LiRA-FineTuned We follow the same hyperparameter setting as LiRA-Supervised, but instead of training from scratch, we initialise the encoder with the pre-trained weights from the pretext task and then fine-tune the entire model for word-level lip-reading (Figure 2c) and sentence-level lip-reading (Figure 2f).

5.3 Results

5.3.1 VSR for Isolated Words

We first evaluate the performance of LiRA-Supervised by training the model from scratch. This leads to an accuracy of 87.6% on LRW which is very close to the state-of-art performance. For LiRA-Frozen, which is pre-trained on LRS3, the learnt visual speech representations are evaluated on word-level lip-reading by training an MS-TCN classifier on top of the frozen representations, as illustrated in Figure 2b. Feature extraction performance (LiRA-Frozen) for different layers is portrayed in Figure 5.3. We observe that the representations extracted from the last layer of the ResNet-18 achieve a maximum accuracy of 83.7 %, which outperforms the current state-of-the-art self-supervised method on LRW [201] by a large absolute margin of 7.8 %, as seen in Table 1. It is clear that the performance generally decreases as the layer

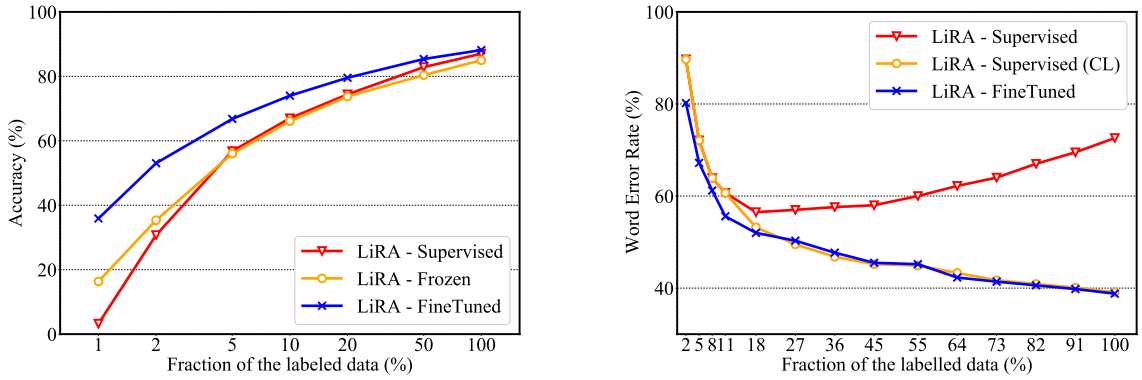


Figure 5.4: Effect of the size of training data on downstream task performance. (a): Accuracy of the end-to-end model as a function of the percentage of the training set (on a logarithmic scale) used for training on LRW. (b) WER achieved by the end-to-end model as a function of the percentage of labelled data used for training on LRS2. All LiRA-Frozen and LiRA-FineTuned models are pre-trained on LRS3 via self-supervision. LiRA-Frozen models are trained using features extracted from the last layer of the ResNet-18 in the pre-trained model, since it achieves the best performance as demonstrated in Figure 5.3. "CL" refers to the model being trained using curriculum learning. LRW and LRS2 contain 165 and 222 hours of labelled training data respectively.

becomes deeper, which may indicate that the features extracted in deeper layers are further tuned towards the pretext task and therefore fail to generalise as well for other tasks.

The performance of the 3 downstream scenarios while varying the amount of training data on LRW is shown in Figure 5.4a. We use LRS3 for self-supervised pre-training. We observe that the feature extraction approach leads to superior performance compared to LiRA-Supervised when using smaller fractions of the labelled training set (1-2 %) and achieves very similar performance for larger amounts of labelled data. This indicates that the pre-trained model learns useful visual features which work well also on LRW. By adopting this methodology, we can simply train the classification layers while the encoder remains frozen, and hence significantly reduce the training time of our model. If we fine-tune the full model, including the encoder, then the performance improves further as shown in Figure 5.4a.

We also observe that the gap between the performance of LiRA-FineTuned and LiRA-Supervised becomes smaller when we increase the amount of labelled data for training. This demonstrates that pre-training using the proposed self-supervised task is particularly beneficial when the labelled training set is very small. In the extreme case, where only 1 % of the labelled training data is used, LiRA-Supervised achieves an accuracy of 1.3 %. In contrast, we obtain 33.9 % accuracy when LiRA-FineTuned is trained using

Table 2: A comparison of the Word Error Rate (WER) between the baseline methods and ours (pre-trained on LRS3) on the LRS2 dataset. CL: Curriculum learning.

Methods	Strategy	WER. (%)
Hyb. CTC/Att. [24]	Supervised	63.5
Conv-seq2seq [203]	Supervised	51.7
TDNN [171]	Supervised	48.9
TM-seq2seq [204]	Supervised	48.3
Hyb. CTC/Att. [196]	Supervised	39.1
KD-seq2seq [195]	Unsupervised	51.3
LiRA-Supervised [196]	Supervised (CL)	39.1
LiRA-FineTuned	Self-supervised	38.8

the same amount of data. This is mainly due to the fact that the self-supervised training provides a good initialisation for network training. We also show that LiRA-Finetuned provides an absolute improvement of 0.5 % in accuracy over LiRA-Supervised when both are trained on full LRW. This demonstrates that LiRA-FineTuned consistently outperforms LiRA-Supervised, even for larger labelled training sets.

5.3.2 VSR for Continuous Speech

To investigate the performance of visual speech representations in a more challenging task, we run training from scratch (Figure 2d) and fine-tuning (Figure 2f) experiments on LRS2 after pre-training on LRS3. We present our results as a function of the fraction of labelled data used during training.

Results are shown in Figure 5.4b. It is evident that the performance of LiRA-FineTuned significantly outperforms the supervised baseline. We also observe that the performance of LiRA-Supervised is hard to optimise without a good initialisation. The performance becomes worse and worse as the training set increases beyond 18 % of the total amount of labelled data. This is likely due to the large variance in length for the videos in LRS2, which makes training from scratch especially difficult. To overcome this problem we use curriculum learning. We first train the model using 11 % of the labelled training set and then use this model for initialisation when training on the entire training set. This curriculum learning strategy allows us to properly train a model which results in a 39.1 % WER.

Fine-tuning the self-supervised model leads to a small improvement over the curriculum learning strategy resulting in a 38.8 % WER. This is the new state-of-the-art performance on the LRS2 dataset when no external labelled datasets are used for training. We also observe that it leads to a 9.5 % absolute

improvement compared to the previous state-of-the-art model [204], as reported in Table 2. Furthermore, as displayed in Figure 5.4, we are able to outperform the previous state-of-the-art of 48.3 % WER using 18× fewer labelled data – 76 hours (36 % of LRS2) vs 1 362 hours (MVLRS, LRS2, and LRS3).

5.4 Conclusion

In this chapter, we have presented an audio-visual self-supervised learning framework to enhance visual speech representations by cross-modal self-supervised learning. In particular, we train a visual model by predicting acoustic features from visual speech and observe that it could be adapted for lip-reading with remarkable success. By fine-tuning our models for this new task, we achieve an accuracy of 88.1 % on LRW and report a WER of 38.8 % on LRS2, which outperforms the models trained from supervised learning. We show that LiRA could function as a feature extractor or even provide a good initialisation for downstream tasks. Therefore, LiRA is powerful to learn effective speech representations, which could be a replacement for feature extraction for downstream tasks. In the next chapter, we will study the Lombard effect in an AVSR system.

Chapter 6

Investigating the Lombard Effect Influence on Audio-Visual Speech Recognition

Contents

6.1 Methodology 100

6.2 Experimental Setup 101

6.3 Results 103

6.4 Conclusion 106

Recently several AVSR models have been presented [2, 19, 23] which aim to augment the performance of acoustic speech recognisers. The main application of such systems is in noisy acoustic environments since the main assumption is that the visual signal is not affected by noise and can therefore enhance the performance of speech recognition systems. However, this assumption is not true due to the Lombard effect, i.e., the change in speaking style in noisy environments which aims to make speech more intelligible and affects both the acoustic characteristics of speech and the lip movements. In addition, such models are usually trained with clean speech which is artificially mixed with additive noise. This approach does not correspond to a realistic scenario where Lombard (and not plain) speech will be mixed with noise. This mismatch can potentially harm the performance of audio-only, video-only and audio-visual speech recognisers.

Few works have investigated the impact of the Lombard effect on audio-only speech recognition [205, 36, 206]. The main finding is that the performance of a model trained on plain speech mixed with noise

is significantly degraded when tested on noisy Lombard speech. This is true even when compensated Lombard speech is used, i.e., the Lombard utterances are normalised to the same energy as the plain speech utterances, although the performance drop is smaller in this case [205]. A similar performance degradation has also been reported for speaker recognition [207]. However, if noisy Lombard speech is used for training then a significant improvement is reported. It is also worth pointing out that the performance of a model trained and tested on noisy Lombard is higher than a model trained and tested on noisy plain speech [205].

Even fewer works have investigated the effect of the Lombard reflex on visual and audio-visual speech recognition and the results are not conclusive. Marxer et al. [205] report an improvement on the recognition of visual Lombard speech no matter if the model is trained on plain or Lombard speech. As expected the improvement is higher when visual Lombard speech is used for training. On the other hand, Heracleous et al. [208] reported a performance drop when there is a mismatch between training and testing conditions. The same conclusion was reached also when an AVSR system was used.

In this work, we investigate the impact of the Lombard effect on end-to-end audio-only, video-only and AVSR. To the best of our knowledge, this is the first work that studies the Lombard effect within the framework of deep end-to-end models which learn to extract features directly from the raw images and audio waveforms. This is in contrast with the majority of previous works which used MFCCs in combination with GMM-HMMs.

In addition, we also consider both multi-speaker and subject-independent scenarios. The former has been extensively studied in previous works [208, 205] and offers an insight on the impact of the Lomard effect. However, in a real scenario we are mainly interested in the performance on unseen speakers. Hence, we first conduct multi-speaker experiments in order to test the claims made by prior works. Then we also conduct subject-independent experiments in order to investigate the performance on unspeen speakers which has not been explored before.

Finally, we report results on sentence-level speech recognition. This is in contrast to previous works which mainly focus either on isolated words [208] or on specific words within a sentence [205]. We believe that the conclusions reached by this approach can be more useful for a practical speech recognition system where the goal will most likely be to recognise all words rather than recognise just isolated words.

We show that properly modelling Lombard speech during training leads to improved performance for audio-only, video-only and audio-visual speech recognition models in all experiments. We also show that in subject-independent experiments, including even a relatively small set of Lombard speech during training can significantly improve the performance of an audio-visual speech recogniser in real conditions, i.e., when testing on noisy Lombard speech. Finally, we show that the standard approach followed in the literature, where noise is mixed with plain speech for training and testing, overestimates the actual performance on noisy Lombard speech in a multi-speaker scenario. In case of a subject-independent scenario, the actual performance on noisy Lombard speech is overestimated for SNRs higher than -3dB but underestimated for lower SNRs. On the other hand, the visual performance is correctly estimated in all scenarios.

6.1 Methodology

6.1.1 Network Architecture

The end-to-end AVSR architecture is shown in Figure 6.1 and is similar to the one proposed in [2]. A CTC loss is added so the model can recognise continuous speech.

Visual Stream visual stream consists of a spatiotemporal convolutional layer, followed by a ResNet-18 [152] and a 2-layer BGRU. Specifically, the temporal-wise 3D convolutional layer has a kernel size of 5 frames. Then, frame-level features are extracted by ResNet-18. The output of ResNet-18 is fed to a 2-layer BGRU to model the temporal dynamics of visual features. Note that the outputs of the forward and backward GRU are concatenated together instead of added together. This means that although there are 128 GRU cells, the features produced by the GRU have a dimensionality of 256.

Audio Stream The audio stream consists of 5 temporal convolutional blocks, followed by a 2-layer BGRU and an average pooling layer. Each convolutional block includes a temporal convolutional layer, ReLU activation and batch normalisation. The first temporal convolutional layer uses a kernel of 5ms and a stride of 0.25ms to extract fine-scale spectral information. The output of the convolutional layers is fed to a 2-layer BGRU. Similarly to the visual stream, the outputs of the forward and backward BGRUs are concatenated. Finally, an averaging pooling layer is used to reduce the audio frame rate to the visual frame rate.

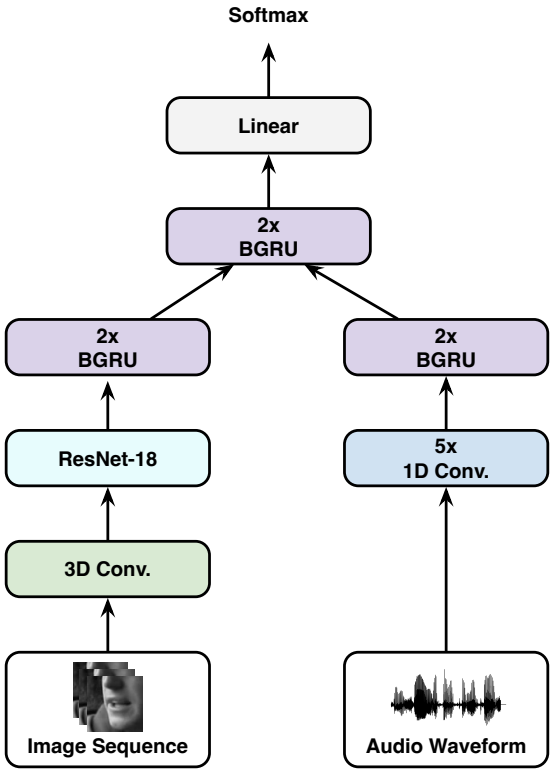


Figure 6.1: End-to-end AVSR architecture overview. Raw images and audio waveforms are fed to the visual and audio streams, respectively, which produce features at the same frame rate at the bottleneck layer. These features are fused together and fed into another 2-layer BGRU to model the temporal dynamics. CTC [4] is used as the loss function.

Fusion Layers Once the 256 audio features and 256 visual features are extracted, they are concatenated and fed into a 2-layer BGRU to model their temporal dynamics. Then a softmax layer follows which provides the characters probabilities for each frame.

6.2 Experimental Setup

6.2.1 Preprocessing

Video Preprocessing We use dlib [209] to detect and track facial landmarks for frontal faces and the face alignment library proposed in [136] for profile faces. The faces are first aligned using a neutral reference frame in order to normalise them for rotation and size differences. This is performed using an affine transform using 5 stable points, two eyes corners in each eye and the tip of the nose. Then the centre of the mouth is located based on the tracked points and a bounding box of 140 by 200 and 80 by 60 is used to extract the mouth ROI on frontal and profile faces, respectively.

Audio Preprocessing Lombard utterances have greater energy than plain speech utterances so for a given noise level their SNR is higher than noisy plain speech. So similarly to [205] we also generate 'compensated' Lombard speech, where the energy of Lombard speech is normalised to the same energy as plain speech. In this case, the SNR between Lombard and plain utterances is the same for a given noise level. To remove the artificial variability of the signals caused by the speaker-to-microphone distance, we follow the approach suggested in [205]. We normalise the non-Lombard and 'compensated' Lombard signals to the same root mean square (RMS) of 0.05. For the Lombard signals, we set the RMS to $0.05 \cdot \bar{x}^{L}_{rms}/\bar{x}^{NL}_{rms}$, where $\bar{x}^{L}_{rms}$ and $\bar{x}^{NL}_{rms}$ are the average RMS value on Lombard speech and non-Lombard speech corpus.

6.2.2 Data Augmentation

During training, two data augmentation methodologies are performed in raw images, random cropping and horizontal flipping. Specifically, each frontal mouth ROI is randomly cropped to a size of 130 by 190 and each profile mouth ROI is randomly cropped to a size of 75 by 55. During testing, the central patch is cropped. Horizontal flipping with a probability of 0.5 is used to increase the variation on training samples.

Babble noise at different levels is added into the audio waveforms during training. The SNR levels range from -15dB to 6dB with an interval of 3dB. One of the noise levels or the clean signal is selected under a uniform distribution, which enhance robustness to different noise levels. Note that this audio noise selection mechanism is only performed in the subject-independent experiments. In the multi-speaker experiments, we train SNR-specific models, i.e., the same noise level is used both for training and testing.

6.2.3 Training Settings

We firstly train each stream from scratch. An initial learning rate of 0.001 and a mini-batch of 64 are used for the audio stream and an initial learning rate of 0.0003 and a mini-batch of 10 are used for the visual stream. We train the audio stream for 400 epochs and the visual stream for 120 epochs separately. Once the audio and visual streams have been trained, their weights are fixed and the 2-layer BGRU used for fusion is trained with an initial learning rate of 0.0003 and a mini-batch of 10. Finally, the entire audio-visual model is fine-tuned for another 40 epochs.

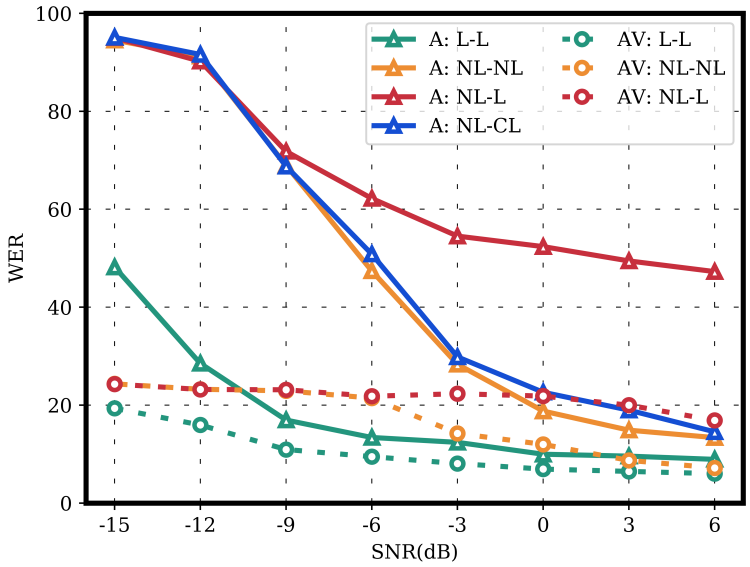


Figure 6.2: WER of the end-to-end models as a function of the noise level in a multi-speaker scenario. A: audio-only model, AV: audio-visual model, L: Lombard, NL: non-Lombard, CL: 'compensated' Lombard. X-Y indicates a model trained on X (L or NL) speech and tested on Y (L or NL or CL) speech. Best seen in colour.

Views	L-L	NL-L	NL-NL
WER (Frontal)	23.57	26.05	25.59

Table 6.1: Video-only results on a multi-speaker scenario. L: Lombard, NL: non-Lombard. X-Y indicates a model trained on X (L or NL) speech and tested on Y (L or NL) speech.

6.3 Results

6.3.1 Multi-Speaker Experiments

In this set of experiments, we investigate the impact of the Lombard effect in a multi-speaker scenario when end-to-end deep models are used for speech recognition. A similar study has been conducted in [205], but a traditional GMM-HMM approach was followed. For the purpose of this study, we use 30, 10 and 10 utterances from each subject for training, validation and testing, respectively.

We first train SNR-specific audio-only models for non-Lombard and Lombard speech. Results are shown in Figure 6.2. We notice that when we train a model on non-Lombard speech and test it on Lombard speech (red solid line), a significant drop in performance compared to testing on non-Lombard speech (orange

Views	L-L	NL-L	NL-NL
WER (Frontal)	25.00	27.84	27.66
WER (Profile)	39.45	47.61	47.47

Table 6.2: Video-only results on subject-independent experiments. L: Lombard, NL: non-Lombard. X-Y indicates a model trained on X (L or NL) speech and tested on Y (L or NL) speech.

solid line) is observed between -9dB and 6dB. This is consistent with the results presented in [205] and is mainly the consequence of the SNR mismatch between Lombard and plain speech. Since SNR-specific models are trained where the noise level is computed on the plain utterances the corresponding Lombard utterances have a higher SNR due to their higher energy. However, between -12dB and -15dB, there is no difference between the two training approaches. This is probably to the high levels of noise which do not allow for proper training of a plain speech recogniser and the WER is above 90%. When we test on 'compensated' Lombard speech, the results are still worse than non-Lombard speech (up to 4%). This is also consistent with [205]. This indicates that not only the SNR mismatch affects the performance but also the difference in acoustic characteristics between Lombard and non-Lombard speech, to a smaller extent though. When we train on Lombard speech, a significant improvement in performance is observed when we test on Lombard speech (green solid line) compared to training and testing on non-Lombard speech (orange solid line).

The results of video-only models are reported in Table 6.1. A slight improvement of 0.45% is reported in the case of NL-L over NL-NL. This is not entirely consistent with [205] who reported a greater improvement of 4.6%. We also show that L-L has an absolute improvement of 2.48% compared to NL-NL, which shows the benefit of properly modelling Lombard speech.

The results of audio-visual models are shown in Figure 6.2. As expected the audio-visual models have a lower WER compared to audio-only models across all noise levels. The same conclusions as in the case of audio-only models are drawn. It is worth pointing out again that when Lombard speech is properly modelled then a better performance is achieved.

6.3.2 Subject-Independent Experiments

Previous experiments considered SNR-specific and multi-speaker models. However, in real scenarios, we would like to have a model that works under different noise levels and on unseen subjects. To better

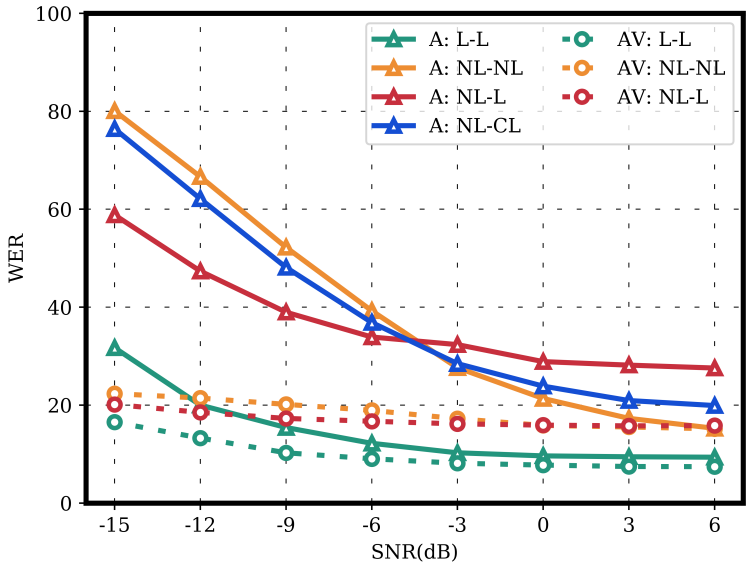


Figure 6.3: WER of the end-to-end as a function of the noise level in a subject-independent scenario. A: audio-only model, AV: audio-visual model, L: Lombard, NL: non-Lombard, CL: 'compensated' Lombard. X-Y indicates a model trained on X (L or NL) speech and tested on Y (L or NL or CL) speech. Best seen in colour.

investigate the impact of the Lombard effect in subject-independent experiments, the training, validation and test sets are divided into 36, 6 and 12 subjects, respectively. It is important to note that the same number of female and male speakers are included on validation and test sets.

The results of audio-only experiments are shown in Figure 6.3. The main difference with the multi-speaker experiments is that the performance on Lombard speech (for a model trained on non-Lombard speech) is better than the performance on non-Lombard speech between -15dB and -6 dB. This is probably due to the fact that during training all SNR levels are seen so the influence of the SNR mismatch between Lombard and plain speech is minimised. The same pattern is also observed for 'compensated' Lombard speech. This indicates that although at higher SNRs the performance of a model trained and tested on non-Lombard speech, which is the usual approach in the literature, overestimates the actual performance, in lower SNRs it actually underestimates it.

The video-only results are reported in Table 6.2. When we train and test a model on Lombard speech, an absolute improvement of 2.66% and 8.02% is observed in frontal faces and profile faces, respectively, over the NL-NL scenario. The performance of NL-L is very similar to NL-NL which reveals that the approach followed in the literature (NL-NL) provides a correct estimate of the actual performance (NL-L). We also

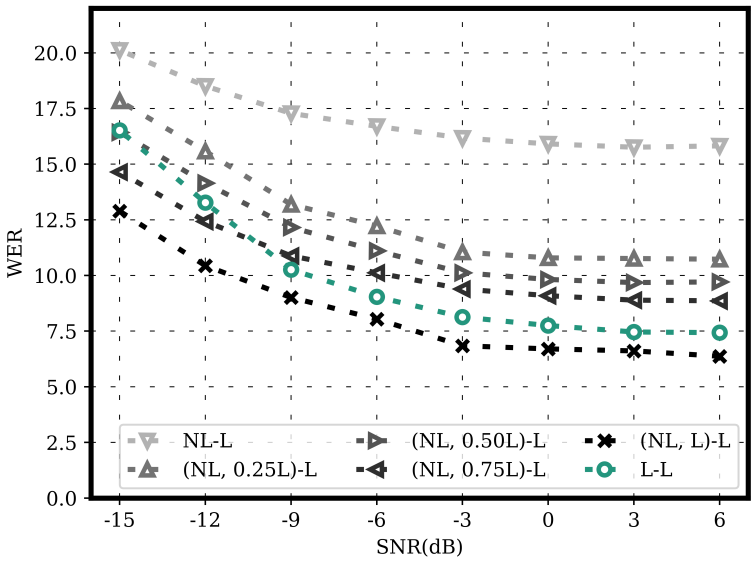


Figure 6.4: WER of the end-to-end audio-visual model as a function of the noise level in a subject-independent scenario. L: Lombard, NL: non-Lombard. (NL,0.25L)-L indicates the performance is reported using a model trained on non-Lombard and 25% Lombard speech and tested on Lombard speech. The other combinations follow the same pattern. Best seen in colour.

notice that the performance on profile faces is much worse due to less information being available as well as inaccurate tracking in profile videos. The results of audio-visual models are shown in Figure 6.3. Similarly to the multi-speaker scenario, the best performance is achieved when a model is trained and tested on Lombard speech.

Figure 6.4 shows the performance of an audio-visual model as a function of the percentage of Lombard speech combined with plain speech for training. It is clear that even when the Lombard speech utterances added to the training set account for 25% of plain speech the gap between NL-L and L-L is reduced to half. Also, when Lombard speech accounts for 50% of plain speech similar performance to the L-L scenario is achieved for very low SNRs.

6.4 Conclusion

In this chapter, we have investigated the impact of the Lombard effect on audio-only, video-only and audio-visual speech recognition. In contrast to the majority of previous works which used MFCCs or DCT-based features in combination with GMM-HMMs to predict specific words (letter and digit), we present the first work which leveraged end-to-end deep architectures for continuous speech. We evaluate

the performance on Lombard GRID in both multi-speaker and subject-independent scenarios. Since there exists an SNR mismatch in Lombard and non-Lombard speech, we show that the performance on Lombard speech in an audio-only model trained on non-Lombard speech is better than the performance on non-Lombard speech. More importantly, we show that training and testing on non-Lombard speech is a bad estimate for the performance on audio-only speech recognition. More specifically, it overestimates the actual performance at higher SNRs but underestimates it in lower SNRs. However, the performance on visual Lombard speech, for a model trained on visual non-Lombard speech, is similar to the performance on visual non-Lombard speech. Furthermore, by adding a small amount of Lombard speech to the training set, we show that the performance in a real scenario could be significantly improved. In future work, we will be interested in how different types of background noise affect the performance of AVSR models.

Chapter 7

Detecting Adversarial Attacks on Audio-Visual Speech Recognition

Contents

7.1 Methodology 110

7.2 Experimental Setup 112

7.3 Results 113

7.4 Conclusion 116

In Chapter 6, we proposed an audio-visual self-supervised approach based on the strong correlation between audio and visual sources, which has achieved state-of-the-art performance on both visual speech recognition for isolated words and continuous speech. However, recent studies [210, 43] show that deep networks are susceptible to adversarial attacks. Given any input $\mathbf{x}$ and a classifier $f(\cdot)$, an adversary tries to carefully construct a sample $\mathbf{x}^{\text{adv}}$ that is similar to $\mathbf{x}$ but $f(\mathbf{x}) \neq f(\mathbf{x}^{\text{adv}})$. The adversarial examples are indistinguishable from the original ones but can easily degrade the performance of deep classifiers.

Existing studies on adversarial attacks have mainly focused in the image domain [43, 44, 45, 46]. Recently, adversarial attacks in the audio domain have also been presented [211, 212]. One of the most prominent studies is the iterative optimisation-based attack [212], which directly operates on an audio clip and enables it to be transcribed to any phrase when a perturbation is added. Works on defense approaches against adversarial attacks can be divided into three categories: adversarial training [43], gradient masking [213]

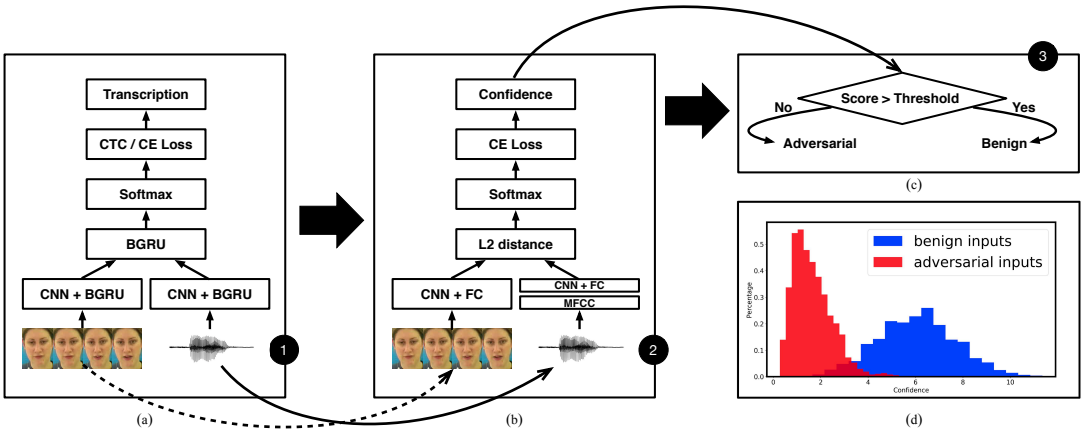


Figure 7.1: An overview of our proposed detection method. (a) A video and an audio clip are fed to the end-to-end AVSR model. They are also fed to the synchronisation network (b) which estimates a synchronisation confidence score used for determining if the audio-visual model has been attacked or not (c). The confidence distribution of 300 adversarial and benign examples from the GRID dataset is shown in (d).

and input transformation [214]. The first one adds adversarial examples in the training set whereas the second one builds a model which does not have useful gradients. Both of them require the model to be retrained, which can be computationally expensive. In contrast, the last one attempts to defend adversarial attacks by transforming the input.

Inspired by the idea of using temporal dependency to detect audio adversarial examples, we propose a simple and efficient detection method against audio-visual adversarial attacks in this chapter. The key idea is that the audio stream is highly correlated with the video of the face (and especially the mouth region). In case of an adversarial example, the added noise on the audio and video streams is expected to weaken the audio-visual correlation. Hence, we propose the use of audio-visual synchronisation as a proxy to correlation. In other words, we expect higher synchronisation scores for benign examples and lower scores for adversarial examples. The proposed detection method is tested on speech recognition attacks on models trained on the LRW [30] and GRID datasets [56]. Our results show that we can detect audio-visual adversarial attacks with high accuracy. This work was published in ICASSP 2021 [215].

In Section 7.1, we introduce the adversarial attacks in audio-only, visual-only and audio-visual models. We describe our synchronisation-based detection approach in Section 7.1.3, Then I present experimental results in Section 7.3.

Generated adversarial samples can be seen at `https://mpc001.github.io/av_adversarial_examples.html`

7.1 Methodology

7.1.1 Attacks

In this study, we consider two attack methods, Fast Gradient Sign Method (FGSM) [43] and the iterative optimisation-based attack [212]. FGSM, which is suitable for attacks on classification models, computes the gradient with respect to the benign input and each pixel can be updated to maximise the loss. Basic Iterative Method (BIM) [216] is an extension of FGSM by applying it multiple times with a small step size. Specifically, given a loss function $J(\cdot,\cdot)$ for training the classification model $f(\cdot)$, the adversarial noise $\mathbf{x}^{\text{adv}}$ is generated as follows:

$$\begin{aligned} \mathbf{x}_0^{\text{adv}} &= \mathbf{x} \\ \mathbf{x}_{N+1}^{\text{adv}} &= \text{Clip}_{\mathbf{x},\epsilon}\{\mathbf{x}_N^{\text{adv}} + \alpha \text{sign}(\nabla_{\mathbf{x}} J(f(\mathbf{x}_N^{\text{adv}}), y^{\text{true}})\} \end{aligned} \tag{7.1}$$

where α is the step size, $\mathbf{x}_N^{\text{adv}}$ is the adversarial example after N-steps of the iterative attack and y^{true} is the true label. After each step, pixel values in the adversarial images $\mathbf{x}^{\text{adv}}$ are clamped to the range $[\mathbf{x}-\epsilon, \mathbf{x}+\epsilon]$, where ϵ is the maximum change in each pixel value. This method was proposed for adversarial attacks on images but can also be applied to audio clips by crafting perturbation to the audio input.

The second type of attack [212] has been recently proposed and is suitable for attacks on continuous speech recognition models. Audio adversarial examples can be generated, which can be transcribed to any phrase but sound similar to the benign one. Specifically, the goal of this targeted attack is to seek an adversary input $\mathbf{x}^{\text{adv}}$, which is very close to the benign input $\mathbf{x}$, but the model decodes it to the target phrase z^{target}. The objective of the attack is the following:

$$\begin{aligned} \text{minimize} \quad & J(f(\mathbf{x}+\delta), z^{\text{target}}) \\ \text{such that} \quad & \|\delta\| < \epsilon \end{aligned} \tag{7.2}$$

where ϵ is introduced to limit the maximum change for each audio sample or pixel and δ is the amount of adversarial noise.

7.1.2 Audio-Visual Speech Recognition Threat Model

The architecture is shown in Figure 1a. We use the end-to-end audio-visual model that was proposed in [217]. The video stream consists of spatiotemporal convolution [17], a modified ResNet18 network and a 2-layer BGRU network whereas the audio stream consists of a 5-layer CNN and a 2-layer BGRU network. These two streams are used for feature extraction from raw modalities. The top two-layer BGRU network further models the temporal dynamics of the concatenated feature.

According to the problem type, two different loss functions are applied for training. The multi-class cross entropy loss, where each input sequence is assigned a single class, is suitable for word-level speech recognition. The CTC loss is used for sentence-level classification. This loss transcribes directly from sequence to sequence when the alignment between inputs and target outputs is unknown. Given an input sequence $\mathbf{x} = (x_1, ..., x_T)$, CTC sums over the probability of all possible alignments to obtain the posterior of the target sequence.

7.1.3 Synchronisation-based Detection Method

Chung et al. [218, 219] introduced the SyncNet model, which is able to predict the synchronisation error when raw audio and video streams are given. This error is quantified by the synchronisation offset and confidence score. A sliding window approach is used to determine the audio-visual offset. For each 5-frame video window, the offset is found when the distance between the visual features and all audio features in a $\pm$ 1 second range is minimised. The confidence score for a particular offset is defined as the difference between the minimum and the median of the Euclidean distances (computed over all windows). Audio and video are considered perfectly matched if the offset approaches to zero with a high level of confidence score.

In this work, we aim to explore if such synchronisation is affected by adversarial noise. The detection method is shown in Figure 1b and 1c. In the detection model, we measure the temporal consistency between the audio and video streams via a model trained for audio-visual synchronisation. For benign audio and video streams, the confidence score should be relatively high since audio and video are aligned and therefore highly synchronised. However, for adversarial audio and video examples, the confidence score is expected to be lower. The added perturbation, which aims to alter the model toward the target

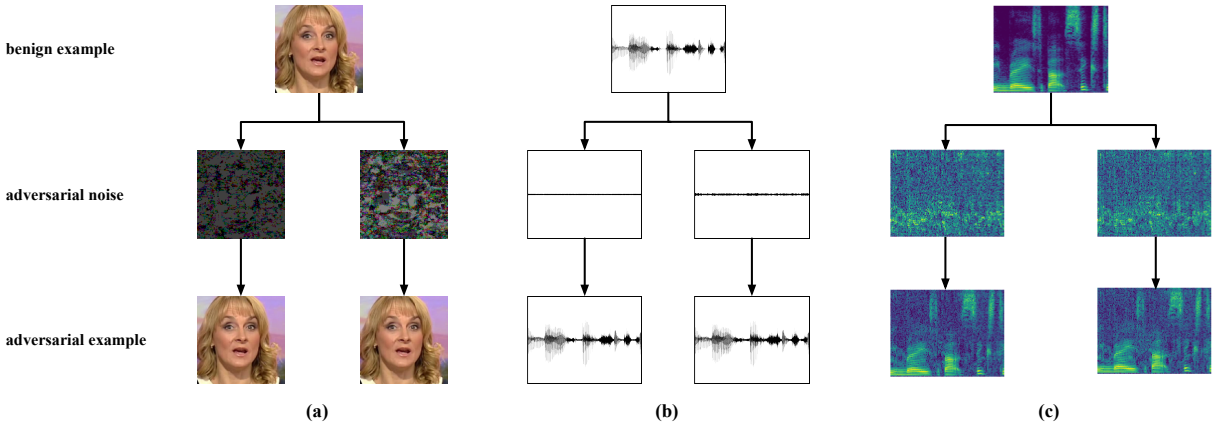


Figure 7.2: One example using basic iterative attack on the LRW dataset. Benign examples, adversarial noise examples, and adversarial examples are illustrated from top to bottom. (a) Raw images (ϵ^V=4, ϵ^V=8), (b) audio waveforms (ϵ^A=256, ϵ^A=512), and (c) audio log-spectrum (ϵ^A=256, ϵ^A=512) are presented from left to right. It is noted that the adversarial visual noise has been scaled with a ratio of 64 for a better illustration since the maximum distortion (ϵ^V=8) is 2 pixels.

transcription, reduces the correlation between the two streams, hence they are less synchronous. Figure 1d. shows the confidence distribution of 300 benign and adversarial examples from the GRID dataset.

7.2 Experimental Setup

7.2.1 Attacks

We evaluate our proposed method using two adversarial attacks on both modalities. We assume the parameters of audio-visual models are known to the attacker.

Attacks against Speech Recognition for Isolated Words Attacks such as FGSM and BIM are suitable for word recognition models trained on the LRW dataset. For FGSM, we consider three values for ϵ^A used in the audio stream (256, 512, 1024) and three values for ϵ^V for the video stream (4, 8, 16). For BIM, the step size α^V was set to 1 in the image domain, which means the value of each pixel is changed by 1 at each iteration. The step size α^A in the audio domain is set to 64. We follow the number of iterations setting suggested by [216], which is selected to be $\min(\epsilon^V + 4, 1.25\epsilon^V)$.

Attacks against Speech Recognition for Continuous Speech For attacking a speech recognition model trained on GRID we use a recently proposed targeted attack [212]. The maximum distortion allowed as

Pixel values are in the range of [0, 255]. Audio samples are in the range of [-32768, 32767].

defined by ϵ (see Eq. 7.2) is limited in $\{256, 512, 1024\}$, $\{4, 8, 16\}$ for audio and video, respectively, and is reduced during iterative optimisation. We implement the attack with 800 iterations. In our studies, 10 random utterances are selected as target utterances. 300 adversarial examples are randomly selected for each target utterance.

7.2.2 Evaluation Metrics

We use the Euclidean distance (L_2) for measuring the similarity between two images. We also use the L_∞ norm to measure the maximum change per pixel. For audio samples we follow [212] and convert the L_∞ norm to the scale of Decibels (dB): $\text{dB}(\mathbf{x}) = \max_i 20 \cdot \log_{10}(x_i)$, where x_i is an arbitrary audio sample point from the audio clip $\mathbf{x}$. The audio distortion is specified as the relative loudness to the benign audio, which can be defined as $\text{dB}_{\mathbf{x}}(\delta) = \text{dB}(\delta) - \text{dB}(\mathbf{x})$.

The Area Under the Curve (AUC) score is used for evaluating the detection approach. We compute the synchronisation confidence score in benign and adversarial examples and by varying the threshold we compute the Receiver Operating Characteristic (ROC) curve.

Finally, in order to compare how this approach would work in a real scenario, we select the threshold (from Figure 1c) which maximises the average F_1 score of adversarial and benign classes on the validation set. Then we use this threshold to compute the average F_1 score on the test set.

7.3 Results

7.3.1 AVSR for Isolated Words

Detection results for attacks on word-level speech recognition are shown in Table 7.1. In the presence of adversarial noise, the Top-1 Accuracy drops from 97.20% [1] to below 40% using FGSM. As ϵ^A and ϵ^V increase the accuracy drops (from 38.27% for the lowest levels of noise to 10.40% for the highest noise levels). On the other hand, the AUC and F1 scores increase, since the highest levels of noise make detection easier. Similar conclusions can be drawn when BIM is used. Accuracy varies between 0% and 7% depending on the noise level, the AUC varies between 0.77 and 0.90 and the F1 scores between 0.71

[1] This is the performance of the model trained on the LRW dataset when benign examples are fed to it.

Attacks (Configuration)	Top-1 Acc.	Distortion		Measures	
		L_2^V	L_∞^A(dB)	AUC	F_1
FGSM (ϵ^A=1024, ϵ^V=16)	10.40%	3.46	-19.26	0.99	0.95
FGSM (ϵ^A=512, ϵ^V=16)	21.87%	3.46	-25.28	0.96	0.89
FGSM (ϵ^A=256, ϵ^V=16)	32.80%	3.46	-31.30	0.90	0.82
FGSM (ϵ^A=1024, ϵ^V=8)	12.40%	1.73	-19.26	0.98	0.94
FGSM (ϵ^A=512, ϵ^V=8)	24.40%	1.73	-25.28	0.94	0.86
FGSM (ϵ^A=256, ϵ^V=8)	34.73%	1.73	-31.30	0.86	0.78
FGSM (ϵ^A=1024, ϵ^V=4)	15.20%	0.87	-19.26	0.98	0.93
FGSM (ϵ^A=512, ϵ^V=4)	27.53%	0.87	-25.28	0.93	0.85
FGSM (ϵ^A=256, ϵ^V=4)	38.27%	0.87	-31.30	0.83	0.76
BIM (ϵ^A=1024, ϵ^V=16)	0.00%	1.66	-19.26	0.90	0.82
BIM (ϵ^A=512, ϵ^V=16)	0.00%	1.66	-25.28	0.89	0.81
BIM (ϵ^A=256, ϵ^V=16)	0.00%	1.70	-31.30	0.84	0.76
BIM (ϵ^A=1024, ϵ^V=8)	0.00%	1.07	-23.34	0.85	0.77
BIM (ϵ^A=512, ϵ^V=8)	0.00%	1.07	-25.28	0.85	0.77
BIM (ϵ^A=256, ϵ^V=8)	0.00%	1.08	-31.30	0.81	0.74
BIM (ϵ^A=1024, ϵ^V=4)	0.07%	0.67	-29.36	0.78	0.72
BIM (ϵ^A=512, ϵ^V=4)	0.07%	0.67	-29.36	0.78	0.72
BIM (ϵ^A=256, ϵ^V=4)	0.07%	0.67	-31.30	0.77	0.71

Table 7.1: Results for the proposed adversarial attack detection approach on word recognition models trained on the LRW dataset.L_∞^V is 1, 2 and 4 pixels when ϵ^V is 4, 8 and 16, respectively.

and 0.82. We should also mention that although adversarial noise is imperceptible for all values of ϵ^V it becomes more and more perceptible as ϵ^A increases.

It is clear from Table 7.1 that for both types of attacks the distortion is smaller when ϵ^A and ϵ^V decrease and as a consequence detection becomes harder: both AUC and F_1 scores go down. However, such attacks are less successful since the classification rate goes up.

We also notice that when the attack is stronger, e.g., BIM is used instead of FSGM, the classification rate goes down, i.e., the attack is more successful, and at the same time the distortion (L_2^V) becomes smaller. Consequently, detection becomes more difficult and this is reflected in the lower AUC and F_1 scores for BIM than FGSM.

7.3.2 AVSR for Continuous Speech

In this section we consider two types of attacks on continuous speech recognition: 1) partially targeted attacks, where the WER between the transcribed result and target phrase is up to 50%, and 2) fully targeted

Threshold		Success	Distortion			Measures	
ϵ^A	ϵ^V	Rate	L_2^V	L_∞^V	L_∞^A(dB)	AUC	F_1
1024	8	100%	3.14	0.019	-43.34	0.84	0.75
512	8	94%	3.38	0.021	-43.93	0.84	0.75
256	8	67%	3.63	0.022	-46.77	0.83	0.75
1024	4	99%	1.54	0.010	-40.14	0.79	0.71
512	4	78%	0.82	0.010	-41.14	0.78	0.71
256	4	42%	1.98	0.012	-45.63	0.74	0.68

Table 7.2: Average results over 10 utterances of the proposed audio-visual synchronisation detection on partially targeted adversarial attacks on continuous speech recognition models trained on GRID. The success rate is the proportion of adversarial examples with WER less than 50%. ($\epsilon^A \in \{256, 512, 1024\}$, $\epsilon^V \in \{4, 8\}$)

Threshold		Success	Distortion			Measures	
ϵ^A	ϵ^V	Rate	L_2^V	L_∞^V	L_∞^A(dB)	AUC	F_1
1024	8	77%	3.26	0.020	-35.22	0.90	0.82
512	8	36%	3.88	0.024	-39.29	0.89	0.81
256	8	8%	4.15	0.026	-43.37	0.87	0.81
1024	4	66%	1.73	0.011	-34.12	0.87	0.78
512	4	19%	2.13	0.013	-38.08	0.84	0.77
256	4	2%	2.17	0.013	-43.85	0.83	0.77

Table 7.3: Average results over 10 utterances of the proposed audio-visual synchronisation detection on fully targeted adversarial attacks on continuous speech recognition models trained on GRID. The success rate is the proportion of adversarial examples with WER = 0%. ($\epsilon^A \in \{256, 512, 1024\}$, $\epsilon^V \in \{4, 8\}$)

Target	Success	Distortion			Measures	
Phrase	Rate	L_2^V	L_∞^V	L_∞^A(dB)	AUC	F_1
bbaazp	81%	1.842	0.011	-41.56	0.78	0.71
bwbonn	70%	1.877	0.012	-40.71	0.79	0.72
lgwysa	62%	1.956	0.012	-40.48	0.80	0.72
lraces	78%	1.795	0.011	-41.38	0.77	0.70
pbapoa	91%	1.821	0.011	-41.51	0.78	0.71
prbaos	81%	1.734	0.011	-41.55	0.77	0.72
prbzts	87%	1.673	0.010	-41.87	0.77	0.70
sgifoa	72%	1.791	0.011	-40.97	0.79	0.71
srixfn	76%	1.824	0.011	-40.12	0.80	0.70
swipfn	83%	1.700	0.011	-41.22	0.78	0.71

Table 7.4: Results of the proposed audio-visual synchronisation detection on partially targeted adversarial attacks on continuous speech recognition models trained on GRID. The WER between transcribed and target phrases is up to 50%. The success rate is the proportion of adversarial examples with WER less than 50%. ($\epsilon^A = 512$, $\epsilon^V = 4$)

attacks where the goal of the attack is that the transcribed result is the same as the desired target phrase (WER = 0%). We also limit the values of ϵ^V to 4 and 8 since $\epsilon^V = 16$ results in very perceptible adversarial examples especially in the case of fully targeted attacks.

Average detection results over 10 utterances for partially targeted attacks on sentence-level speech recognition are shown in Table 7.2. It is clear that the success rate is fairly high, over 90% in most cases. Only when ϵ^A is 256 and ϵ^V is 4 then the attack is much less successful with a success rate of 42%. At the same time the detection rates are quite high for most combinations of the two thresholds, varying between 0.74 and 0.84 for AUC and 0.68 to 0.75 for F1 score.

Average detection results over 10 utterances for fully targeted attacks on sentence-level speech recognition are shown in Table 7.3. In this case the success rates are much lower than the partially targeted attack due to the difficulty of the task. Relatively high success rates are observed when ϵ^V is either 4 or 8 and ϵ^A is 1024 which results in more perceptible adversarial examples. In addition the generated audio and video adversarial examples are more distorted than the ones generated by the partially targeted attacks. In turn, this leads to higher AUC scores, between 0.83 and 0.90, and F1 scores, between 0.77 and 0.82.

Results per sentence for the partially targeted attack when ϵ^V is 4 and ϵ^A is 512 are shown in Table 7.4. Although the success rates vary a lot (from 62% to 91%) depending on the sentence, detection measures AUC and F1 are similar for all sentences. We also observe that the maximum distortions applied to the audio and video signals are similar in most cases.

7.4 Conclusion

In this chapter, we propose to leverage audio-visual synchronisation as a detection method of adversarial attacks. In contrast to previous work focusing on exploring the detection methods of attacks against audio-only models or visual-only models, we have investigated the detection methods in end-to-end audio-visual models. Specifically, we hypothesise that the synchronisation confidence score would be lower in adversarial than benign examples and demonstrate that this could be used for detecting adversarial attacks. To verify this hypothesis, adversarial attacks are first applied to word-level classification and continuous speech recognition models, respectively. For the former, we apply both FGSM and the iterative

bbaazp: bin blue at a zero please, bwbonn: bin white by o nine now, lgwysa: lay green with y seven again, lraces: lay red at c eight soon, pbapoa; place blue at p one again, prbaos: place red by a one soon, prbzts: place red by z two soon, sgifoa: set green in f one again, srixfn: set red in x four now, swipfn: set white in p five now.

optimisation-based attack on the LRW dataset. For the latter, we apply both partially targeted attacks and fully targeted attacks based on [212] on the GRID dataset. We empirically show that our methods could detect adversarial attacks with a high detection rate. Furthermore, we present per-utterance results in partially targeted attack experiments, showing that the maximum distortions applied to the audio and visual streams are similar in most cases. For future work, It would be interesting to note that a more traditional non-deep learning model which cannot be attacked could be investigated. Furthermore, we are interested in developing better audio-visual synchronisation methods for the detection of adversarial attacks. Another avenue for follow-up research will be how to leverage the temporal dependency in the detection of deep fake videos.

Chapter 8

Conclusion

Contents

8.1 Summary 118

8.2 Applications 120

8.3 Challenges 121

8.4 Ethical Considerations 122

8.5 Future Work 124

8.1 Summary

This thesis investigates the problem of AVSR in realistic scenarios. We first study AVSR of isolated words in Chapter 3. Most previous state-of-the-art models [12, 13, 17, 2] in this field relied on multiple training phases, which resulted in complex training procedures. To address this issue, we propose a VSR model that can be trained in an end-to-end fashion and therefore greatly simplifies the training process. We empirically show that our model could help not only improve performance but also reduce the training time. On top of this pipeline, we introduce a new temporal model named MS-TCN to learn from multi-scale temporal dimensions, which significantly increase the recognition accuracy. Two different variants of TCN are also introduced and evaluated, which are DC-TCN and DS-TCN, respectively. The former variant (DC-TCN) captures the temporal information in a dense favour and therefore has demonstrated more accurate VSR performance, while the latter one (DS-TCN) utilise depth-wise convolution to accelerate the computational speed at the cost of slightly reduced accuracy. Additionally, we also investigate how

different data augmentation techniques can improve the performance and enhance the robustness of AVSR models against noises. The knowledge distillation technique is also employed to learn more accurate models.

In Chapter 4, we take a further step from word-level AVSR to continuous speech recognition. In this chapter, we present an AVSR model based on a ResNet-18 and Conformer that can be trained in an end-to-end manner. The audio and visual encoders learn to extract features directly from raw image pixels and audio waveforms, respectively, and the extracted features were subsequently fed into conformer models for temporal aggregation, while the audio-visual fusion is performed through a MLP. The evaluation on continuous speech recognition datasets demonstrates that state-of-the-art performance can be achieved through the combinations of 1) training on large-scale datasets, which is the current trend in AVSR community, and 2) the application of a carefully designed end-to-end model. We emphasise the importance of hyper-parameter optimisation and data augmentation techniques like time-masking in the training stage, which are crucial to learn more accurate and more robust models. We also propose a new architecture based on auxiliary tasks, i.e. the VSR model also needs to learn from audio visual representations obtained by pre-trained ASR and VSR models. Last but not least, we provide empirical evidence that using larger datasets can improves the performance, which is in line with recent works in this field. Our approach outperforms all existing VSR baselines trained on publicly available datasets in English, Spanish and Mandarin by a large margin.

Instead of collecting and annotating a large number of audio-visual pairs, which is costly and tedious, we study how to leverage large amounts of unlabelled audio-visual data to learn better speech representations in a novel self-supervised framework. In Chapter 5, we propose a novel framework to learn visual speech representations from audio. Specifically, we train a ResNet+Conformer model to predict acoustic features from unlabelled visual speech. Experimental results showed that our approach significantly outperforms other self-supervised methods on the LRW dataset and achieves state-of-the-art performance on LRS2 using only a fraction of the total labelled data.

We study the impact of the Lombard effect on end-to-end ASR, VSR, and AVSR models in Chapter 6. Experiments are performed under multi-speaker and subject-independent scenarios. We show that it is beneficial to properly model Lombard speech. We also show that training and testing on noisy plain speech, which is commonly used in the literature, is a good estimate for the performance on visual

Lombard speech but a bad estimate for the performance of audio-only speech recognition. Furthermore, we propose to include Lombard speech in the training set to compensate the performance gap between plain speech and Lombard speech in the training set.

Inspired by the idea of using temporal dependency to detect audio adversarial examples in [47], we investigate the usage of audio-visual synchronisation as a detection method of adversarial attacks in Chapter 7. The key idea is that the audio stream is highly correlated to the video of the mouth ROI. We hypothesise that the synchronisation confidence score would be lower in adversarial than benign examples and demonstrated that this could be used for detecting adversarial attacks. The proposed detection method is evaluated on speech recognition attacks on models trained on the LRW and GRID datasets. Our experimental results show that the proposed method could detect audio-visual adversarial attacks with high accuracy.

8.2 Applications

Speech is the most commonly used human communication method and consists of an audio signal and the corresponding mouth movements. Speech perception is also bimodal as demonstrated by the McGurk effect [220] where the perception of a sound may change depending on the lip movements shown to the observers. In addition, it has been shown that the addition of visual speech information to a word recognition task performed by normal hearing adults is equivalent to increasing the signal-to-noise ratio (SNR) by 15 dB compared to audio-only recognition [221]. Hence, one of the main applications of VSR is to enhance the performance of ASR models in noisy environments. VSR models are not significantly affected by acoustic noise[††] and can be integrated into an audio-visual speech recognition (AVSR) model to compensate for the performance drop of ASR models. Several AVSR architectures have been proposed [51, 151, 33, 24, 171, 222, 223] which show that the improvement over ASR models is greater as the noise level increases, i.e., the SNR is lower. The same VSR architectures can also be used to improve the performance of audio-based models in a variety of applications like speech enhancement [224], speech separation [58, 225], voice activity detection [226], active speaker detection [227] and speaker diarisation [228].

[††] Due to the Lombard effect [35] speakers adapt their speaking style in noisy environments. This affects the lip movements

There is also a number of applications based exclusively on VSR. Silent Speech Interfaces (SSI) [229], which can enable speech communication to take place when an audible speech signal is not available, can be developed with the help of VSR systems. This means that a speaker would be able to mouth words instead of vocalising them. This technology has the potential to transform the lives of speech impaired people. Patients who have lost the ability to speak (aphonia) or have difficulty in speaking (dysphonia) due to tracheostomy, laryngectomy, stroke or injury might find it hard to communicate with others. The use of SSI can alleviate this by providing an alternative way of communication and at the same time reduce the stress caused by the sudden loss of their voice. The use of SSI can also be useful in cases where speaking is not allowed, e.g., in a meeting, and can provide privacy in public conversations.

VSR technology also opens up opportunities to automatically transcribe video content which was recorded without audio, like silent movies, CCTV footage or video captured by older webcams, and would otherwise require significant manual effort or might have even been impossible. It can also be used as a useful tool in face forgery detection [230]. Most face manipulation approaches add inconsistencies in mouth movements, which might not always be perceptible by humans, but they can easily be detected by properly trained VSR models. Finally, there is a new form of VSR which has become popular recently and generates audio, instead of text, directly from the input video [231, 232, 233, 234]. This is essentially a combination of a standard VSR model with a text-to-speech model but has two important advantages: 1) It does not require any transcribed dataset and can be trained with vast amounts of unlabelled audio-visual data, 2) It is faster and can potentially be used in real-time applications as it removes the constraint of recognising a complete word before generating the corresponding speech signal. This new approach is especially useful for audio inpainting applications since it can automatically fill in audio gaps from video.

8.3 Challenges

Despite the great advances in VSR there are still a number of challenges that need to be solved before the full potential of this technology can be achieved. First of all, visual ambiguities which arise from the fact that different phonemes correspond to similar lip movements is one of the most important reasons for the significant performance gap between ASR and VSR models. Designing VSR systems which can resolve some of these ambiguities by relying more on the context, like the time masking augmentation proposed

but to a much less extent than the impact of noise to the acoustic signal.

in this work, might close this gap. In addition, VSR systems are sensitive to visual noise like lighting changes, occlusions, motion blur and compression. Reduced and/or mismatched resolution and frame rate between training and test conditions can also affect the performance. There is some evidence that VSR systems are robust to small or moderate amounts of noise and less robust to reduced resolution [235, 236] but further studies are needed to establish the impact of each noise type.

Another challenge is that a VSR model should be person-independent and pose-invariant. However, it is well known that deep networks rely heavily on texture [237]. This can potentially degrade the performance since unknown test subjects and head pose can significantly affect the appearance of the mouth. This is typically addressed by training the VSR models on a large number of subjects with varying poses. Some preliminary works on pose-invariant [238] and subject-independent [239] VSR has shown that this can be addressed in a more principled way and this is another area which deserves further attention. Similarly, multi-view VSR [240, 241] can be beneficial but it is not clear yet which lip views are optimal and how they should be combined. The availability of multiple cameras in meeting rooms, cars and in modern smartphones opens up a new opportunity for improving VSR systems.

The vast majority of VSR systems have focused on plain English speech. However, it is known that lip movements are affected by the context where speech is produced and the type of speech. There is evidence that lip movements tend to increase in silent speech [242] and also when speech is produced in noise (the Lombard effect) [41, 243]. Despite studies which show a performance drop when VSR models [244, 29, 245, 208] are tested on such conditions this area still remains unexplored. Finally, the development of non-English VSR systems which take into account the unique characteristics of each language also remains an open challenge.

8.4 Ethical Considerations

It is important to note that VSR is a dual-use technology, which means it can have a positive impact on society as well as negative. Although our objective is to build VSR systems that will be beneficial for the society, like the applications mentioned above, this technology can also be misused. One example is that it can be deployed for surveillance via CCTV or even with smartphone cameras which raises privacy concerns. A potential side effect of this is that it might discourage people from speaking in public if they believe that their conversation can be intercepted by anyone carrying a camera. Sophisticated surveillance

using VSR technology might not be possible at the moment, especially via CCTV due to the low quality of images compared to the high quality data used during training, but it should not be ignored. Cameras and VSR systems are getting better so it might become a serious privacy concern rather soon.

Commercial applications of VSR technology are still at a very early stage. One of the very few examples is a smartphone application which aims to help speech impaired patients communicate and it is currently being trialled in UK NHS hospitals. This is being developed by Liopa [246] which also works on keyword spotting from CCTV footage. Hence, we argue that appropriate government regulations about VSR systems, which address the privacy concerns and potential misuse, are necessary at this early stage before the technology is fully commercialised. This will allow the proper auditing of every new application before it reaches the market and its risks and merits can be properly communicated to the users and the public. Otherwise VSR systems may have the same fate as face recognition technology which was commercialised without proper regulation being in place. As a consequence, a ban on using face recognition was introduced in several cities [247, 248, 249] and some companies either stopped offering such services or put restrictions on their use [250, 251, 252, 253] when the ethical concerns became widely known.

It should also be pointed out that VSR technology might be biased against specific age groups, gender, cultural backgrounds or non-native speakers. Most of the publicly available datasets have been collected from TV programs, TED talks or YouTube videos. Hence, it is very likely that some groups are underrepresented, e.g., younger people when data are collected from TV programs or older people when data are collected from YouTube. Similarly, it is likely that people from specific cultural backgrounds or non-native speakers are also underrepresented. This will lead to VSR models which are less accurate for all these groups. Since demographic information is not available for any publicly available dataset used for training VSR models it is not easy to verify if such biases exist. Recently, some datasets have been released, like Casual Conversations [254], which contain such information so it might be possible to test VSR models for potential biases. However, this does not alleviate the issue, it just reveals the existence of bias. VSR models need to be trained on demographically diverse data including non-native speakers to ensure similar performance across different user groups. This will lead to VSR systems whose accuracy is not lower for some users because their age, gender, cultural background or accent is underrepresented in the training data.

8.5 Future Work

The central research topic of this thesis is *how to leverage both audio and visual signals for better speech recognition?* This topic can be further extended along five different directions outlined below.

As discussed in in Chapter 3 and Chapter 4, there are still unsolved challenges in audio-visual fusions. A first thought is to design an adaptive fusion mechanism that learns the weights of different data modality based on the noise levels. Inspiring ideas can be found in works such as cross-modal alignment between acoustic and visual encoders [23]. We should also notice that the noisy environments in most works are artificially created by adding random noises to the audio signals, which can be different from the noisy distribution in realistic scenarios. This issue can be potentially addressed by collecting noisy audio data from various conditions. Besides, it is also interesting to distinguish how different types of background noises can affect the performance of AVSR models. Furthermore, [255] introduced an additional loss to penalise the differences of clean and noisy representations in multiple layers. Inspired by this work, it would be interesting to investigate how much performance gains brings when a cumulative penalty is applied.

Cross-modal learning is explored in Chapter 5, and we can further investigate whether cross-modal distillation can improve the performance of audio-visual speech recognition models. As shown in Chapter 5, cross-modal supervised learning can introduce benefits, but will self-supervised learning between audio and visual signals still work positively? This is an interesting idea to investigate. Given the extent of modern audio-visual corpora, we believe it would be promising to leverage self-supervised learning towards other visual tasks such as emotion recognition and speaker recognition in the future. Moreover, independently pre-training audio and visual models is complex, and it would be interesting to develop end-to-end systems by jointly learning audio-visual representations like AV-HuBERT [256].

We have studied the Lombard effect influence on E2E audio-visual speech recognition in Chapter 6. The empirical results showed that including Lombard speech in the training set can compensate for the performance gap between non-Lombard speech and Lombard speech. Michelsanti *et al.* [245] showed that the speech enhancement system trained with Lombard speech outperformed the one trained with non-Lombard speech in terms of both estimated speech quality and estimated speech intelligibility. Nonetheless, collecting speech in Lombard conditions is costly. It would be interesting to investigate how

to reconstruct Lombard speech from a video of a spoken non-Lombard speech.

We have shown that the usage of audio-visual synchronisation could be considered as a detection of the method for adversarial attacks in Chapter 7. To achieve high accuracy of detection, we would like to investigate an approach that can effectively measure the correlation between audio and visual streams. The consistency between audio and visual signal can be a key to extend our system to the detection against fake videos.

The ability to perform multi-lingual analysis is also an attractive feature to add to our pipeline. Most current VSR systems rely on visual speech datasets in Mandarin, English and Spanish, while the support for other languages can be essential. How to tackle with the follow-up problem, e.g. the data sparseness problems in low-resource corpora, is also worth considerations. A multilingual self-supervised VSR, which sounds appealing and attractive, can be the pursuit of the next generation AVSR applications.

Bibliography

[1] P. Ma, R. Mira, S. Petridis, B. W. Schuller, and M. Pantic, "Lira: Learning visual speech representations from audio through self-supervision," in *Proceedings of the 22nd Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 3011–3015, 2021.

[2] S. Petridis, T. Stafylakis, P. Ma, F. Cai, G. Tzimiropoulos, and M. Pantic, "End-to-end audiovisual speech recognition," in *Proceedings of the 43rd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6548–6552, 2018.

[3] B. Martinez, P. Ma, S. Petridis, and M. Pantic, "Lipreading using temporal convolutional networks," in *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6319–6323, 2020.

[4] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pp. 369–376, 2006.

[5] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. W. Senior, "Recent advances in the automatic recognition of audiovisual speech," *Proceedings of the IEEE*, vol. 91, no. 9, pp. 1306–1326, 2003.

[6] S. Dupont and J. Luettin, "Audio-visual speech modeling for continuous speech recognition," *IEEE Transactions on Multimedia*, vol. 2, no. 3, pp. 141–151, 2000.

[7] S. Petridis and M. Pantic, "Prediction-based audiovisual fusion for classification of non-linguistic vocalisations," *IEEE Transactions on Affective Computing*, vol. 7, no. 1, pp. 45–58, 2015.

[8] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pp. 689–696, 2011.

[9] D. Hu, X. Li, and X. Lu, "Temporal multimodal learning in audiovisual speech recognition," in *Proceedings of the 29th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3574–3582, 2016.

[10] H. Ninomiya, N. Kitaoka, S. Tamura, Y. Iribe, and K. Takeda, "Integration of deep bottleneck features for audio-visual speech recognition," in *Proceedings of the 16th Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2015.

[11] Y. Mroueh, E. Marcheret, and V. Goel, "Deep multimodal learning for audio-visual speech recognition," in *Proceedings of the 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2130–2134, 2015.

[12] Y. Takashima, R. Aihara, T. Takiguchi, Y. Ariki, N. Mitani, K. Omori, and K. Nakazono, "Audio-visual speech recognition using bimodal-trained bottleneck features for a person with severe hearing loss," in *Proceedings of the 17th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 277–281, 2016.

[13] S. Petridis and M. Pantic, "Deep complementary bottleneck features for visual speech recognition," in *Proceedings of the 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2304–2308, 2016.

[14] S. Petridis, Y. Wang, Z. Li, and M. Pantic, "End-to-end audiovisual fusion with lstms," in *Proceedings of the International Conference on Auditory-Visual Speech Processing (AVSP)*, pp. 36–40, 2017.

[15] M. Wand, J. Koutnik, and J. Schmidhuber, "Lipreading with long short-term memory," in *Proceedings of the 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6115–6119, 2016.

[16] Y. Assael, B. Shillingford, S. Whiteson, and N. De Freitas, "Lipnet: End-to-end sentence-level lipreading," *Preprint at arXiv*, 2016.

[17] T. Stafylakis and G. Tzimiropoulos, "Combining residual networks with LSTMs for lipreading," in *Proceedings of the 18th Annual Conference of International Speech Communication Association (INTERSPEECH)*, vol. 9, pp. 3652–3656, 2017.

[18] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. Nicolaou, B. Schuller, and S. Zafeiriou, "Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network," in *Proceedings of the 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5200–5204, 2016.

[19] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Lip reading sentences in the wild," in *Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3444–3453, 2017.

[20] I. Anina, Z. Zhou, G. Zhao, and M. Pietikäinen, "Ouluvs2: A multi-view audiovisual database for non-rigid mouth motion analysis," in *Proceedings of the 11th IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, pp. 1–5, 2015.

[21] A. Koumparoulis and G. Potamianos, "MobiLipNet: Resource-efficient deep learning based lipreading," in *Proceedings of the 20th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 2763–2767, 2019.

[22] N. Shrivastava, A. Saxena, Y. Kumar, P. Kaur, R. R. Shah, and D. Mahata, "Mobivsr: A visual speech recognition solution for mobile devices," *Proceedings of the 20th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 2753–2757, 2019.

[23] G. Sterpu, C. Saam, and N. Harte, "Attention-based audio-visual fusion for robust automatic speech recognition," in *Proceedings of 20th International Conference on Multimodal Interaction (ICMI)*, pp. 111–115, 2018.

[24] S. Petridis, T. Stafylakis, P. Ma, G. Tzimiropoulos, and M. Pantic, "Audio-visual speech recognition with a hybrid ctc/attention architecture," in *Proceedings of the IEEE Spoken Language Technology (SLT) Workshop*, pp. 513–520, 2018.

[25] S. J. Cox, R. W. Harvey, Y. Lan, J. L. Newman, and B. Theobald, "The challenge of multispeaker lip-reading," in *Proceedings of the International Conference on Auditory-Visual Speech Processing (AVSP)*, pp. 179–184, 2008.

[26] P. Lucey, G. Potamianos, and S. Sridharan, "Patch-based analysis of visual speech from multiple views," in *Proceedings of the International Conference on Auditory-Visual Speech Processing (AVSP)*, pp. 69–74, 2008.

[27] A. Z. K. Frisky, C. Wang, A. Santoso, and J. Wang, "Lip-based visual speech recognition system," in *Proceedings of the International Carnahan Conference on Security Technology (ICCST)*, pp. 315–319, 2015.

[28] D. Lee, J. Lee, and K. Kim, "Multi-view automatic lip-reading using neural network," in *Proceedings of the 13th Asian Conference on Computer Vision (ACCV) Workshops*, vol. 10117, pp. 290–302, 2016.

[29] S. Petridis, J. Shen, D. Cetin, and M. Pantic, "Visual-only recognition of normal, whispered and silent speech," in *Proceedings of the 43rd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6219–6223, 2018.

[30] J. S. Chung and A. Zisserman, "Lip reading in the wild," in *Proceedings of the 13th Asian Conference on Computer Vision (ACCV)*, vol. 10112, pp. 87–103, 2016.

[31] T. Stafylakis and G. Tzimiropoulos, "Combining residual networks with LSTMs for lipreading," in *Proceedings of the 18th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 3652–3656, 2017.

[32] B. Shillingford, Y. Assael, M. W. Hoffman, T. Paine, C. Hughes, U. Prabhu, H. Liao, H. Sak, K. Rao, L. Bennett, *et al.*, "Large-scale visual speech recognition," in *Proceedings of the 20th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 4135–4139, 2019.

[33] T. Makino, H. Liao, Y. Assael, B. Shillingford, B. Garcia, O. Braga, and O. Siohan, "Recurrent neural network transducer for audio-visual speech recognition," in *Proceedings of the IEEE Automatic Speech Recognition and Understanding (ASRU) Workshop*, pp. 905–912, 2019.

[34] D. Serdyuk, O. Braga, and O. Siohan, "Audio-visual speech recognition is worth 32×32×8 voxels," *Preprint at arXiv*, 2021.

[35] E. Lombard, "Le signe de l'elevation de la voix," *Ann. Mal. de L'Oreille et du Larynx*, pp. 101–119, 1911.

[36] J.-C. Junqua, "The lombard reflex and its role on human listeners and automatic speech recognizers," *The Journal of the Acoustical Society of America*, vol. 93, no. 1, pp. 510–524, 1993.

[37] A. L. Pittman and T. L. Wiley, "Recognition of speech produced in noise," *Journal of Speech, Language, and Hearing Research*, 2001.

[38] N. Alghamdi, S. Maddock, R. Marxer, J. Barker, and G. J. Brown, "A corpus of audio-visual lombard speech with frontal and profile views," *The Journal of the Acoustical Society of America*, vol. 143, no. 6, pp. EL523–EL529, 2018.

[39] W. V. Summers, D. B. Pisoni, R. H. Bernacki, R. I. Pedlow, and M. A. Stokes, "Effects of noise on speech production: Acoustic and perceptual analyses," *The Journal of the Acoustical Society of America*, vol. 84, no. 3, pp. 917–928, 1988.

[40] M. Garnier, L. Ménard, and B. Alexandre, "Hyper-articulation in lombard speech: An active communicative strategy to enhance visible speech cues?," *The Journal of the Acoustical Society of America*, vol. 144, no. 2, pp. 1059–1074, 2018.

[41] J. Šimko, Š. Beňuš, and M. Vainio, "Hyperarticulation in lombard speech: Global coordination of the jaw, lips and the tongue," *The Journal of the Acoustical Society of America*, vol. 139, no. 1, pp. 151–162, 2016.

[42] E. Vatikiotis-Bateson, A. V. Barbosa, C. Y. Chow, M. Oberg, J. Tan, and H. C. Yehia, "Audiovisual lombard speech: reconciling production and perception.," in *Proceedings of the International Conference on Auditory-Visual Speech Processing (AVSP)*, p. 41, 2007.

[43] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[44] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation (TEVC)*, vol. 23, pp. 828–841, 2019.

[45] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 284–293, 2018.

[46] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," in *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[47] Z. Yang, B. Li, P. Chen, and D. Song, "Characterizing audio adversarial examples using temporal dependency," in *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

[48] E. K. Patterson, S. Gurbuz, Z. Tufekci, and J. N. Gowdy, "CUAVE: A new audio-visual database for multimodal human-computer interface research," in *Proceedings of the 27th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2017–2020, 2002.

[49] I. A. Matthews, T. F. Cootes, J. A. Bangham, S. J. Cox, and R. W. Harvey, "Extraction of visual features for lipreading," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 198–213, 2002.

[50] S. Yang, Y. Zhang, D. Feng, M. Yang, C. Wang, J. Xiao, K. Long, S. Shan, and X. Chen, "LRW-1000: A naturally-distributed large-scale benchmark for lip reading in the wild," in *Proceedings of the 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, pp. 1–8, 2019.

[51] T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Deep audio-visual speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[52] T. Afouras, J. S. Chung, and A. Zisserman, "LRS3-TED: a large-scale dataset for visual speech recognition," *Preprint at arXiv*, 2018.

[53] Y. Zhao, R. Xu, and M. Song, "A cascade sequence-to-sequence model for chinese mandarin lip reading," in *Proceedings of the ACM International Conference on Multimedia in Asia (MM Asia)*, 2019.

[54] A. B. Zadeh, Y. Cao, S. Hessner, P. P. Liang, S. Poria, and L. Morency, "CMU-MOSEAS: A multimodal language dataset for spanish, portuguese, german and french," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1801–1812, 2020.

[55] E. Salesky, M. Wiesner, J. Bremerman, R. Cattoni, M. Negri, M. Turchi, D. W. Oard, and M. Post, "The Multilingual TEDx Corpus for Speech Recognition and Translation," in *Proceedings of the 22nd Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 3655–3659, 2021.

[56] 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*, vol. 120, pp. 2421–2424, 2006.

[57] K. Vougioukas, S. Petridis, and M. Pantic, "End-to-end speech-driven realistic facial animation with temporal gans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) workshop*, 2018.

[58] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein, "Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 112:1–112:11, 2018.

[59] T. Afouras, J. S. Chung, and A. Zisserman, "ASR is all you need: Cross-modal distillation for lip reading," in *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2143–2147, 2020.

[60] J. S. Bridle and M. D. Brown, "An experimental automatic word-recognition system," *Proceedings of the IEEE*, 1974.

[61] P. Mermelstein, "Distance measures for speech recognition, psychological and instrumental," *Pattern Recognition and Artificial Intelligence*, pp. 374–388, 1976.

[62] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[63] R. B. Blackman, J. W. Tukey, and T. Teichmann, "The measurement of power spectra," *Physics Today*, vol. 13, no. 2, pp. 52–54, 1960.

[64] O. Essenwanger, *Elements of Statistical Analysis*. General Climatology, Springer Science & Business Media, 1986.

[65] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, 1937.

[66] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[67] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 5998–6008, 2017.

[68] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th Neural Information Processing System (NIPS)*, pp. 3104–3112, 2014.

[69] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, "End-to-end dense video captioning with masked transformer," in *Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8739–8748, 2018.

[70] N. C. Camgöz, O. Koller, S. Hadfield, and R. Bowden, "Sign language transformers: Joint end-to-end sign language recognition and translation," in *Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10020–10030, 2020.

[71] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[72] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *CoRR*, vol. abs/1508.01211, 2015.

[73] N. Jaitly, Q. V. Le, O. Vinyals, I. Sutskever, D. Sussillo, and S. Bengio, "An online sequence-to-sequence model using partial conditioning," in *Proceedings of the 29th Neural Information Processing System (NIPS)*, pp. 5067–5075, 2016.

[74] C. Raffel, M. Luong, P. J. Liu, R. J. Weiss, and D. Eck, "Online and linear-time attention by enforcing monotonic alignments," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, vol. 70, pp. 2837–2846, 2017.

[75] H. Sak, M. Shannon, K. Rao, and F. Beaufays, "Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping," in *Proceedings of the 18th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 1298–1302, 2017.

[76] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proceedings of 28th Advances in Neural Information Processing System (NIPS)*, pp. 577–585, 2015.

[77] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4774–4778, IEEE, 2018.

[78] G. Pundak and T. Sainath, "Lower frame rate neural network acoustic models," in *Proceedings of the 17th Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2016.

[79] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[80] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1412–1421, 2015.

[81] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *CoRR*, vol. abs/1410.5401, 2014.

[82] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition.," in *Proceedings of the 18th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 939–943, 2017.

[83] E. Battenberg, J. Chen, R. Child, A. Coates, Y. G. Y. Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu, "Exploring neural transducers for end-to-end speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 206–213, IEEE, 2017.

[84] Y. He, R. Prabhavalkar, K. Rao, W. Li, A. Bakhtin, and I. McGraw, "Streaming small-footprint keyword spotting using sequence-to-sequence models," in *Proceedings of the IEEE Automatic Speech Recognition and Understanding (ASRU) Workshop*, pp. 474–481, 2017.

[85] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K. C. Sim, T. Bagby, S. Chang, K. Rao, and A. Gruenstein, "Streaming end-to-end speech recognition for mobile devices," in *Proceedings of the 44th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6381–6385, 2019.

[86] J. L. Elman, "Finding structure in time," *Cogn. Sci.*, vol. 14, no. 2, pp. 179–211, 1990.

[87] P. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[88] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proceedings of the 22nd Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 1045–1048, 2010.

[89] A. Graves, A. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6645–6649, 2013.

[90] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 802–810, 2015.

[91] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[92] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proceedings of the 30th International Conference on Machine Learning*, vol. 28 of *JMLR Workshop and Conference Proceedings*, pp. 1310–1318, 2013.

[93] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735–1780, 11 1997.

[94] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[95] A. H. Waibel, T. Hanazawa, G. E. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 3, pp. 328–339, 1989.

[96] K. Yamaguchi, K. Sakamoto, T. Akabane, and Y. Fujimoto, "A neural network for speaker-independent isolated word recognition," in *The First International Conference on Spoken Language Processing, ICSLP 1990, Kobe, Japan, November 18-22, 1990*, ISCA, 1990.

[97] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proceedings of the 16th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 3214–3218, 2015.

[98] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, 2016.

[99] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *Proceedings of the 9th ISCA Speech Synthesis Workshop*, p. 125, 2016.

[100] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[101] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 40, no. 4, pp. 834–848, 2017.

[102] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2881–2890, 2017.

[103] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[104] N. Zeghidour, N. Usunier, G. Synnaeve, R. Collobert, and E. Dupoux, "End-to-end speech recognition from the raw waveform," in *Proceedings of the 19th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 781–785, 2018.

[105] T. Parcollet, M. Morchid, and G. Linares, "E2e-sincnet: Toward fully end-to-end speech recognition," in *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7714–7718, 2020.

[106] P. Golik, Z. Tüske, R. Schlüter, and H. Ney, "Convolutional neural networks for acoustic modeling of raw time signal in LVCSR," in *Proceedings of the 16th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 26–30, 2015.

[107] T. N. Sainath, R. J. Weiss, A. W. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform cldnns," in *Proceedings of the 16th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 1–5, 2015.

[108] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *Proceedings of the 44th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5791–5795, 2019.

[109] C. Wang, M. Li, and A. J. Smola, "Language models with transformers," *CoRR*, vol. abs/1904.09408, 2019.

[110] S. Karita, X. Wang, S. Watanabe, T. Yoshimura, W. Zhang, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, and R. Yamamoto, "A comparative study on transformer vs RNN in speech applications," in *Proceedings of the IEEE Automatic Speech Recognition and Understanding (ASRU) Workshop*, pp. 449–456, 2019.

[111] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "Sound event detection of weakly labelled data with cnn-transformer and automatic threshold optimization," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 2450–2460, 2020.

[112] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Proceedings of the 21st Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 5036–5040, 2020.

[113] Y. Gong, Y. Chung, and J. R. Glass, "AST: audio spectrogram transformer," *CoRR*, vol. abs/2104.01778, 2021.

[114] P. Ma, Y. Wang, J. Shen, S. Petridis, and M. Pantic, "Lip-reading with densely connected temporal convolutional networks," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2857–2866, 2021.

[115] P. Ma, B. Martinez, S. Petridis, and M. Pantic, "Towards practical lipreading with distilled and efficient models," in *Proceedings of the 46th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7608–7612, 2021.

[116] P. Ma, Y. Wang, S. Petridis, J. Shen, and M. Pantic, "Training strategies for improved lip-reading," in *Proceedings of the 47th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8472–8476, 2022.

[117] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3377–3381, 2013.

[118] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, "Audio-visual speech recognition using deep learning," *Applied Intelligence*, vol. 42, no. 4, pp. 722–737, 2015.

[119] S. Petridis and M. Pantic, "Deep complementary bottleneck features for visual speech recognition," in *Proceedings of the 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[120] T. Stafylakis, M. H. Khan, and G. Tzimiropoulos, "Pushing the boundaries of audiovisual word recognition using residual networks and lstms," *Computer Vision and Image Understanding*, vol. 176, pp. 22–32, 2018.

[121] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born-again neural networks," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 1602–1611, 2018.

[122] I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[123] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proceedings of the 14th European Conference on Computer Vision (ECCV)*, pp. 630–645, 2016.

[124] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708, 2017.

[125] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3684–3692, 2018.

[126] D. Guo, S. Wang, Q. Tian, and M. Wang, "Dense temporal convolution network for sign language translation," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 744–750, 2019.

[127] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7132–7141, 2018.

[128] 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*, vol. abs/1704.04861, 2017.

[129] N. Ma, X. Zhang, H. Zheng, and J. Sun, "ShufflenetV2: practical guidelines for efficient CNN architecture design," in *Proceedings of the 15th European Conference on Computer Vision (ECCV)*, pp. 122–138, 2018.

[130] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1251–1258, 2017.

[131] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 2613–2617, 2019.

[132] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proceedings of the 27th Neural Information Processing System (NIPS) Workshops*, 2014.

[133] H. Bagherinezhad, M. Horton, M. Rastegari, and A. Farhadi, "Label refinery: Improving ImageNet classification through label progression," *arXiv:1805.02641*, 2018.

[134] C. Yang, L. Xie, S. Qiao, and A. L. Yuille, "Training deep neural networks in generations: A more tolerant teacher educates better students," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pp. 5628–5635, 2019.

[135] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-stage dense face localisation in the wild," in *Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5203–5212, 2020.

[136] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks)," in *Proceedings of the 16th IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[137] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

[138] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, vol. 37 of *JMLR Workshop and Conference Proceedings*, pp. 448–456, 2015.

[139] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

[140] T. Afouras, J. S. Chung, and A. Zisserman, "Lrs3-ted: a large-scale dataset for visual speech recognition," in *arXiv preprint arXiv:1809.00496*, 2018.

[141] P. Ma, S. Petridis, and M. Pantic, "Visual speech recognition for multiple languages in the wild," *CoRR*, vol. abs/2202.13084, 2022.

[142] X. Weng and K. Kitani, "Learning spatio-temporal features with two-stream deep 3d cnns for lipreading," in *Proceedings of the British Machine Vision Conference (BMVC)*, p. 269, 2019.

[143] Y. Zhang, S. Yang, J. Xiao, S. Shan, and X. Chen, "Can we read speech beyond the lips? rethinking roi selection for deep visual speech recognition," in *Proceedings of the 15th IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, 2020.

[144] D. Feng, S. Yang, S. Shan, and X. Chen, "Learn an effective lip reading model without pains," *CoRR*, vol. abs/2011.07557, 2020.

[145] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520, 2018.

[146] Y. Liu, X. Jia, M. Tan, R. Vemulapalli, Y. Zhu, B. Green, and X. Wang, "Search to distill: Pearls are everywhere but not the eyes," in *Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7536–7545, 2020.

[147] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. W. Senior, "Recent advances in the automatic recognition of audiovisual speech," *Proceedings of the IEEE*, vol. 91, no. 9, pp. 1306–1326, 2003.

[148] S. Dupont and J. Luettin, "Audio-visual speech modeling for continuous speech recognition," *IEEE Transactions on Multimedia*, vol. 2, no. 3, pp. 141–151, 2000.

[149] X. Zhang, H. Gong, X. Dai, F. Yang, N. Liu, and M. Liu, "Understanding pictograph with facial features: End-to-end sentence-level lip reading of chinese," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pp. 9211–9218, 2019.

[150] S. Ma, S. Wang, and X. Lin, "A transformer-based model for sentence-level chinese mandarin lipreading," in *Proceedings of the 5th IEEE International Conference on Data Science in Cyberspace*, pp. 78–81, 2020.

[151] P. Ma, S. Petridis, and M. Pantic, "End-to-end audio-visual speech recognition with conformers," in *Proceedings of the 46th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7613–7617, 2021.

[152] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 29th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[153] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*, pp. 2978–2988, 2019.

[154] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International Conference on Machine Learning*, vol. 70, pp. 933–941, 2017.

[155] C. Liu, F. Zhang, D. Le, S. Kim, Y. Saraf, and G. Zweig, "Improving rnn transducer based asr with auxiliary tasks," in *Proceedings of the IEEE Spoken Language Technology (SLT) Workshop*, pp. 172–179, 2021.

[156] S. Toshniwal, H. Tang, L. Lu, and K. Livescu, "Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition," in *Proceedings of the 18th Annual Conference of International Speech Communication Association*, p. 3532–3536, 2017.

[157] J. Lee and S. Watanabe, "Intermediate loss regularization for ctc-based speech recognition," in *Proceedings of the 46th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6224–6228, 2021.

[158] A. Shukla, S. Petridis, and M. Pantic, "Learning speech representations from raw audio by joint audiovisual self-supervision," in *Proceedings of the 37th International Conference on Machine Learning (ICML) Workshop*, 2020.

[159] P. Ma, R. Mira, S. Petridis, B. W. Schuller, and M. Pantic, "LiRA: Learning Visual Speech Representations from Audio Through Self-Supervision," in *Proceedings of the 22nd Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 3011–3015, 2021.

[160] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[161] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[162] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proceedings of the 20th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 2613–2617, 2019.

[163] K. Irie, A. Zeyer, R. Schlüter, and H. Ney, "Language modeling with deep transformers," in *Proceedings of the 20th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 3905–3909, 2019.

[164] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proceedings of the 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015.

[165] F. Hernandez, V. Nguyen, S. Ghannay, N. A. Tomashenko, and Y. Estève, "TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation," in *International Conference on Speech and Computer*, vol. 11096, pp. 198–208, 2018.

[166] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 4218–4222, 2020.

[167] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "MLS: A large-scale multilingual dataset for speech research," in *Proceedings of the 21st Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 2757–2761, 2020.

[168] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proceedings of the 19th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 2207–2211, 2018.

[169] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 12 2014.

[170] S. Ren, Y. Du, J. Lv, G. Han, and S. He, "Learning from the master: Distilling cross-modal advanced knowledge for lip reading," in *Proceedings of the 34th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13325–13333, 2021.

[171] J. Yu, S. Zhang, J. Wu, S. Ghorbani, B. Wu, S. Kang, S. Liu, X. Liu, H. Meng, and D. Yu, "Audio-visual recognition of overlapped speech for the LRS2 dataset," in *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6984–6988, 2020.

[172] D. Serdyuk, O. Braga, and O. Siohan, "Transformer-based video front-ends for audio-visual speech recognition." Preprint at `https://arxiv.org/abs/2201.10439`, 2022.

[173] Y. Zhao, R. Xu, X. Wang, P. Hou, H. Tang, and M. Song, "Hearing lips: Improving lip reading by distilling speech recognizers," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 6917–6924, 2020.

[174] C. Li and Y. Qian, "Listen, watch and understand at the cocktail party: Audio-visual-contextual speech separation," in *Proceedings of the 21st Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 1426–1430, 2020.

[175] E. Kharitonov, M. Rivière, G. Synnaeve, L. Wolf, P.-E. Mazaré, M. Douze, and E. Dupoux, "Data augmenting contrastive learning of speech representations in the time domain," in *Proceedings of the IEEE Spoken Language Technology (SLT) Workshop*, pp. 215–222, 2021.

[176] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proceedings of the 14th European Conference on Computer Vision (ECCV)*, vol. 9907, pp. 649–666, 2016.

[177] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proceedings of the 14th European Conference on Computer Vision (ECCV)*, vol. 9910, pp. 69–84, 2016.

[178] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," in *Proceedings of the 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1920–1929, 2019.

[179] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018.

[180] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proceedings of the 20th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 3465–3469, 2019.

[181] M. Ravanelli and Y. Bengio, "Learning speaker representations with mutual information," in *Proceedings of the 20th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 1153–1157, 2019.

[182] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," in *Proceedings of the 20th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 161–165, 2019.

[183] M. Ravanelli, J. Zhong, S. Pascual, P. Swietojanski, J. Monteiro, J. Trmal, and Y. Bengio, "Multi-task self-supervised learning for robust speech recognition," in *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6989–6993, 2020.

[184] S. Petridis and M. Pantic, "Prediction-based audiovisual fusion for classification of non-linguistic vocalisations," *IEEE Transactions on Affective Computing*, vol. 7, no. 1, pp. 45–58, 2016.

[185] S. Petridis, M. Pantic, and J. Cohn, "Prediction-based classification for audiovisual discrimination between laughter and speech," in *Proceedings of the 9th IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, pp. 619–626, 2011.

[186] H. Alwassel, D. Mahajan, L. Torresani, B. Ghanem, and D. Tran, "Self-supervised learning by cross-modal audio-video clustering," *CoRR*, vol. abs/1911.12667, 2019.

[187] A. J. Piergiovanni, A. Angelova, and M. S. Ryoo, "Evolving losses for unsupervised video representation learning," in *Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 130–139, 2020.

[188] A. Owens and A. A. Efros, "Audio-visual scene analysis with self-supervised multisensory features," in *Proceedings of the 15th European Conference on Computer Vision (ECCV)*, vol. 11210, pp. 639–658, 2018.

[189] H. Pham, P. P. Liang, T. Manzini, L. Morency, and B. Póczos, "Found in translation: Learning robust joint representations by cyclic translations between modalities," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pp. 6892–6899, 2019.

[190] A. Shukla, S. Petridis, and M. Pantic, "Does visual self-supervision improve learning of speech representations?," *CoRR*, vol. abs/2005.01400, 2020.

[191] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba, "Learning sight from sound: Ambient sound provides supervision for visual learning," *Int. J. Comput. Vis.*, vol. 126, no. 10, pp. 1120–1137, 2018.

[192] R. Arandjelovic and A. Zisserman, "Look, listen and learn," in *Proceedings of the 16th IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 609–617, 2017.

[193] B. Korbar, D. Tran, and L. Torresani, "Cooperative learning of audio and video models from self-supervised synchronization," in *Proceedings of the 31st Neural Information Processing System (NIPS)*, pp. 7774–7785, 2018.

[194] S. Chung, J. S. Chung, and H. Kang, "Perfect match: Self-supervised embeddings for cross-modal retrieval," *J. Sel. Top. Signal Process.*, vol. 14, no. 3, pp. 568–576, 2020.

[195] T. Afouras, J. S. Chung, and A. Zisserman, "ASR is all you need: Cross-modal distillation for lip reading," in *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2143–2147, 2020.

[196] P. Ma, S. Petridis, and M. Pantic, "End-to-end audio-visual speech recognition with conformers," in *Proceedings of the 46th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[197] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 30th Neural Information Processing System (NIPS)*, pp. 5998–6008, 2017.

[198] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*, pp. 2978–2988, 2019.

[199] X. Weng and K. Kitani, "Learning spatio-temporal features with two-stream deep 3D CNNs for lipreading," in *Proceedings of the British Machine Vision Conference (BMVC)*, p. 269, 2019.

[200] P. Ma, Y. Wang, J. Shen, S. Petridis, and M. Pantic, "Lip-reading with densely connected temporal convolutional networks," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2856–2865, 2021.

[201] S. Chung, H. Kang, and J. S. Chung, "Seeing voices and hearing voices: learning discriminative embeddings using cross-modal self-supervision," in *Proceedings of the 21st Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 3486–3490, 2020.

[202] M. K. Tellamekala, M. F. Valstar, M. Pound, and T. Giesbrecht, "Audio-visual predictive coding for self-supervised visual representation learning," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pp. 9912–9919, 2020.

[203] X. Zhang, F. Cheng, and S. Wang, "Spatio-temporal fusion based convolutional sequence learning for lip reading," in *Proceedings of the 17th IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 713–722, 2019.

[204] T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Deep audio-visual speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[205] R. Marxer, J. Barker, N. Alghamdi, and S. Maddock, "The impact of the lombard effect on audio and visual speech recognition systems," *Speech Communication*, vol. 100, pp. 58–68, 2018.

[206] A. Wakao, K. Takeda, and F. Itakura, "Variability of lombard effects under different noise conditions," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, vol. 4, pp. 2009–2012, 1996.

[207] J. H. Hansen and V. Varadarajan, "Analysis and compensation of lombard speech across noise type and levels with application to in-set/out-of-set speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 2, pp. 366–378, 2009.

[208] P. Heracleous, C. T. Ishi, M. Sato, H. Ishiguro, and N. Hagita, "Analysis of the visual lombard effect and automatic recognition experiments," *Computer Speech & Language*, vol. 27, no. 1, pp. 288–300, 2013.

[209] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[210] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.

[211] M. Alzantot, B. Balaji, and M. Srivastava, "Did you hear that? adversarial examples against automatic speech recognition," in *Proceedings of the 5th Conference and Workshop on Neural Information Processing Systems (NIPSW)*, 2017.

[212] N. Carlini and D. Wagner, "Audio adversarial examples: targeted attacks on speech-to-text," in *Proceedings of the 39th IEEE Security and Privacy (SP) Workshops*, pp. 1–7, 2018.

[213] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proceedings of the 37th IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, 2016.

[214] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: detecting adversarial examples in deep neural networks," in *Proceedings of the Network and Distributed Systems Security (NDSS) Symposium*, 2019.

[215] P. Ma, S. Petridis, and M. Pantic, "Detecting adversarial attacks on audio-visual speech recognition," in *Proceedings of the 46th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6403–6407, 2021.

[216] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proceedings of 5th International Conference on Learning Representations (ICLR) Workshops*, 2017.

[217] P. Ma, S. Petridis, and M. Pantic, "Investigating the lombard effect influence on end-to-end audio-visual speech recognition," in *Proceedings of the 20th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 4090–4094, 2019.

[218] J. S. Chung and A. Zisserman, "Out of time: automated lip sync in the wild," in *Proceedings of the 13th Asian Conference on Computer Vision (ACCV)*, pp. 251–263, 2016.

[219] S. Chung, J. S. Chung, and H. Kang, "Perfect match: Improved cross-modal embeddings for audio-visual synchronisation," in *Proceedings of the 44th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3965–3969, 2019.

[220] H. McGurk and J. MacDonald, "Hearing lips and seeing voices," *Nature*, vol. 264, no. 5588, pp. 746–748, 1976.

[221] W. H. Sumby and I. Pollack, "Visual contribution to speech intelligibility in noise," *The Journal of the Acoustical Society of America*, vol. 26, no. 2, pp. 212–215, 1954.

[222] W. Yu, S. Zeiler, and D. Kolossa, "Fusing information streams in end-to-end audio-visual speech recognition," in *Proceedings of the 46th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3430–3434, 2021.

[223] G. Sterpu, C. Saam, and N. Harte, "How to teach DNNs to pay attention to the visual modality in speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1052–1064, 2020.

[224] T. Afouras, J. S. Chung, and A. Zisserman, "The conversation: Deep audio-visual speech enhancement," in *Proceedings of the 19th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 3244–3248, 2018.

[225] R. Gao and K. Grauman, "Visualvoice: Audio-visual speech separation with cross-modal consistency," in *Proceedings of the 34th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15495–15505, 2021.

[226] T. Yoshimura, T. Hayashi, K. Takeda, and S. Watanabe, "End-to-end automatic speech recognition integrated with ctc-based voice activity detection," in *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6999–7003, 2020.

[227] Y. J. Kim, H. Heo, S. Choe, S. Chung, Y. Kwon, B. Lee, Y. Kwon, and J. S. Chung, "Look who's talking: Active speaker detection in the wild," in *Proceedings of the 22nd Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 3675–3679, 2021.

[228] J. S. Chung, J. Huh, A. Nagrani, T. Afouras, and A. Zisserman, "Spot the conversation: Speaker diarisation in the wild," in *Proceedings of the 21st Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 299–303, 2020.

[229] B. Denby, T. Schultz, K. Honda, T. Hueber, J. M. Gilbert, and J. S. Brumberg, "Silent speech interfaces," *Speech Communication*, vol. 52, no. 4, pp. 270–287, 2010.

[230] A. Haliassos, K. Vougioukas, S. Petridis, and M. Pantic, "Lips don't lie: A generalisable and robust approach to face forgery detection," in *Proceedings of the 34th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5039–5049, 2021.

[231] A. Ephrat and S. Peleg, "Vid2speech: speech reconstruction from silent video," in *Proceedings of the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5095–5099, 2017.

[232] R. Mira, K. Vougioukas, P. Ma, S. Petridis, B. W. Schuller, and M. Pantic, "End-to-end video-to-speech synthesis using generative adversarial networks," *Preprint at arXiv*, 2021.

[233] D. Michelsanti, O. Slizovskaia, G. Haro, E. Gómez, Z.-H. Tan, and J. Jensen, "Vocoder-Based Speech Synthesis from Silent Videos," in *Proceedings of the 21st Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 3530–3534, 2020.

[234] K. Prajwal, R. Mukhopadhyay, V. P. Namboodiri, and C. Jawahar, "Learning individual speaking styles for accurate lip to speech synthesis," in *Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13796–13805, 2020.

[235] L. Dungan, A. Karaali, and N. Harte, "The impact of reduced video quality on visual speech recognition," in *Proceedings of the 25th IEEE International Conference on Image Processing (ICIP)*, pp. 2560–2564, 2018.

[236] H. L. Bear, R. Harvey, B.-J. Theobald, and Y. Lan, "Resolution limits on visual speech recognition," in *Proceedings of the 21st IEEE International Conference on Image Processing (ICIP)*, pp. 1371–1375, 2014.

[237] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," in *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

[238] S. Cheng, P. Ma, G. Tzimiropoulos, S. Petridis, A. Bulat, J. Shen, and M. Pantic, "Towards pose-invariant lip-reading," in *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4357–4361, 2020.

[239] M. Wand and J. Schmidhuber, "Improving speaker-independent lipreading with domain-adversarial training," in *Proceedings of the 18th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 3662–3666, 2017.

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

[241] D. Lee, J. Lee, and K.-E. Kim, "Multi-view automatic lip-reading using neural network," in *Proceedings of the 13th Asian Conference on Computer Vision (ACCV)*, pp. 290–302, 2016.

[242] K. Bicevskis, J. de Vries, L. Green, J. Heim, J. Božič, J. D'Aquisto, M. Fry, E. Sadlier-Brown, O. Tkachman, N. Yamane, *et al.*, "Effects of mouthing and interlocutor presence on movements of visible vs. non-visible articulators," *Canadian acoustics= Acoustique canadienne*, vol. 44, no. 1, p. 17, 2016.

[243] M. Garnier, L. Ménard, and G. Richard, "Effect of being seen on the production of visible speech cues. a pilot study on lombard speech," in *Proceedings of the 13th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 611–614, 2012.

[244] P. Ma, S. Petridis, and M. Pantic, "Investigating the Lombard Effect Influence on End-to-End Audio-Visual Speech Recognition," in *Proceedings of the 20th Annual Conference of International Speech Communication Association (INTERSPEECH)*, pp. 4090–4094, 2019.

[245] D. Michelsanti, Z.-H. Tan, S. Sigurdsson, and J. Jensen, "Deep-learning-based audio-visual speech enhancement in presence of lombard effect," *Speech Communication*, vol. 115, pp. 38–50, 2019.

[246] "Liopa - the world's only startup focused on automated lipreading via visual speech recognition." https://liopa.ai/. [Online; accessed 24-November-2021].

[247] "Maine passes the strongest state facial recognition ban yet." https://www.theverge.com/2021/6/30/22557516/maine-facial-recognition-ban-state-law, 2021. [Online; accessed 24-November-2021].

[248] "Facial Recognition Laws Are (Literally) All Over the Map." https://www.wired.com/story/facial-recognition-laws-are-literally-all-over-the-map/, 2019. [Online; accessed 24-November-2021].

[249] "13 Cities Where Police Are Banned From Using Facial Recognition Tech." https://innotechtoday.com/13-cities-where-police-are-banned-from-using-facial-recognition-tech/, 2020. [Online; accessed 24-November-2021].

[250] "An Update On Our Use of Face Recognition." https://about.fb.com/news/2021/11/update-on-use-of-face-recognition/, 2021. [Online; accessed 24-November-2021].

[251] "Amazon will block police indefinitely from using its facial-recognition software." https://edition.cnn.com/2021/05/18/tech/amazon-police-facial-recognition-ban/index.html, 2021. [Online; accessed 24-November-2021].

[252] "Microsoft won't sell police its facial-recognition technology, following similar moves by Amazon and IBM." https://www.washingtonpost.com/technology/2020/06/11/microsoft-facial-recognition/, 2020. [Online; accessed 24-November-2021].

[253] "IBM abandons 'biased' facial recognition tech." https://www.bbc.com/news/technology-52978191, 2020. [Online; accessed 24-November-2021].

[254] C. Hazirbas, J. Bitton, B. Dolhansky, J. Pan, A. Gordo, and C. Canton-Ferrer, "Casual conversations: A dataset for measuring fairness in AI," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2289–2293, 2021.

[255] D. Liang, Z. Huang, and Z. C. Lipton, "Learning noise-invariant representations for robust speech recognition," in *Proceedings of the IEEE Spoken Language Technology (SLT) Workshop*, pp. 56–63, 2018.

[256] B. Shi, W.-N. Hsu, K. Lakhotia, and A. Mohamed, "Learning audio-visual speech representation by masked multimodal cluster prediction," in *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022.