Visual recognition of human communication

D.Phil. Thesis

Robotics Research Group
Department of Engineering Science
University of Oxford

Supervisor:
Professor Andrew Zisserman

Joon Son Chung
St. Catherine's College

2017

Abstract

The objective of this work is visual recognition of speech and gestures. Solving this problem opens up a host of applications, such as transcribing archival silent films, or resolving multi-talker simultaneous speech, but most importantly it helps to advance the state of the art in speech recognition by enabling machines to take advantage of the multi-modal nature of human communications. However, visual recognition of speech and gestures is a challenging problem, in part due to the lack of annotations and datasets, but also due to the inter- and intra-personal variations, and in the case of visual speech, ambiguities arising from homophones.

Training a deep learning algorithm requires a lot of training data. We propose a method to automatically collect, process and generate a large-scale audio-visual corpus from television videos temporally aligned with the transcript. To build such dataset, it is essential to know 'who' is speaking 'when'. We develop a ConvNet model that learns joint embedding of the sound and the mouth images from unlabelled data, and apply this network to the tasks of audio-to-video synchronisation and active speaker detection. Not only does this play a crucial role in building the dataset that forms the basis of much of the research done in this thesis, the method learns powerful representations of the visual and auditory inputs which can be used for related tasks such as lip reading. We also show that the methods developed here can be extended to the problem of generating talking faces from audio and still images.

We then propose a number of deep learning models that are able to recognise visual speech at word and sentence level. In both scenarios, we also demonstrate recognition performance that exceeds the state of the art on public datasets; and in the case of the latter, the lip reading performance beats a professional lip reader on videos from BBC television. We also demonstrate that if audio is available, then visual information helps to improve speech recognition performance.

Next, we present a method to *recognise* and *localise* short temporal signals in image time series, where strong supervision is not available for training. We propose image encodings and ConvNet-based architectures to first recognise the signal, and then to localise the signal using back-propagation. The method is demonstrated for localising spoken words in audio, and for localising signed gestures in British Sign Language (BSL) videos.

Finally, we explore the problem of speaker recognition. Whereas previous works for speaker identification have been limited to constrained conditions, here we build a new large-scale speaker recognition dataset collected from 'in the wild' videos using an automated pipeline. We propose a number of ConvNet architectures that outperforms traditional baselines on this dataset.

Declaration

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Joon Son Chung

Acknowledgements

I would like to thank my supervisor Professor Andrew Zisserman for his guidance, support, advice and encouragement. I would also like to thank my collaborators, Relja Arandjelović, Giles Bergel, Alexandra Franklin, Amir Jamaludin, Arsha Nagrani, Andrew Senior and Oriol Vinyals for all their contributions. Special thanks to Rob Cooper and Matt Haynes at BBC Research for help in obtaining the datasets. I also thank everyone in VGG for making it such a nice environment to work in. I would like to thank my family for all their support and understanding.

Contents

List of abbreviations

Abbreviation	Explanation
ASR	Automatic Speech Recognition
BLEU	Bilingual Evaluation Understudy
BSL	British Sign Language
CCA	Canonical Correlation Analysis
CNN	Convolutional Neural Network
CoIA	Co-Inertia Analysis
CTC	Connectionist Temporal Classification
DBF	Deep Bottleneck Feature
DCT	Discrete Cosine Transform
DNN	Deep Neural Network
ED	Edit Distance
FFT	Fast Fourier Transform
GIF	GA-based Informative Feature
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
LDA	Linear Discriminant Analysis
LSTM	Long Short Term Memory
MFCC	Mel-Frequency Cepstral Coefficients
MIL	Multiple Instance Learning
MLP	Multi Layer Perceptrons
OCR	Optical Character Recognition
PLP	Probabilistic Logic Programming
POI	Person of Interest
ReLU	Rectified Linear Units
ROC	Receiver Operating Characteristic
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
SIFT	Scale-Invariant Feature Transform
SSD	Single Shot MultiBox Detector
UBM	Universal Background Model

Supplementary materials

Supplementary materials (including video examples) can be downloaded from the VGG website (`http://www.robots.ox.ac.uk/~vgg`), and are also available from the Oxford Research Archive, DOI 10.5287/bodleian:r1bmVwpOn.

Chapter 1

Introduction

1.1 Objective and motivation

Humans use many signals to communicate with each other – speech is the primary signal for many people, but mouth shapes (McGurk and MacDonald, 1976), facial expressions (Morris, 1994), body gestures (Goldin-Meadow, 1999) and haptics (Field, 2004) also play a crucial role in human communication and speech understanding.

So suppose you are having a conversation with someone at a party, or in a noisy train station. Humans have a remarkable ability to make use of both visual and auditory inputs to understand what is said, particularly in environments with background noise.

On the other hand, the research community in speech perception has mostly focussed on the audio part of the inherently multimodal signal. Speech perception in noisy environments is an extremely difficult task, particularly without the visual information. Therefore while current automatic speech recognition (ASR) systems can achieve very high accuracy in noiseless environments, they often struggle in presence of background noise.

The objective of this thesis is to bridge this gap.

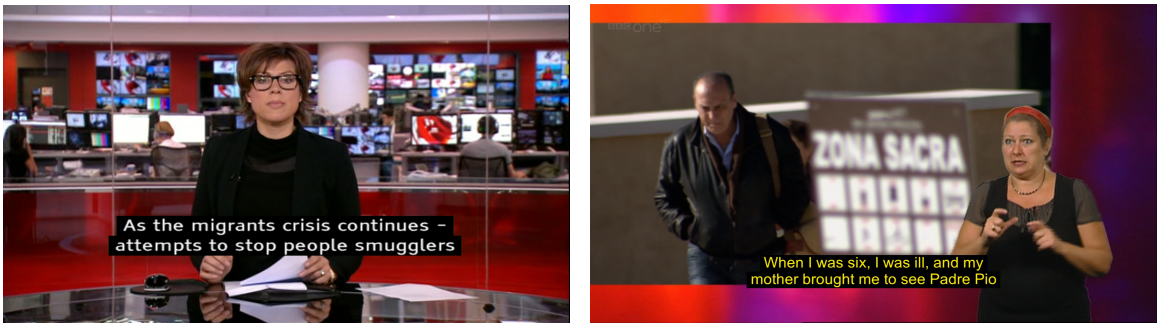

Figure 1.1: **Left:** regular news with subtitles; **Right:** sign interpreted TV with subtitles.

Not only would this be convenient to normal-hearing users of ASR-based services that are increasingly powering our everyday lives, this is extremely beneficial to the hard of hearing and the elderly for whom the visual expressions and gestures are important modes of communication. The ability to recognise visual communications also opens up a host of other applications, including resolving simultaneous multi-talker speech, and transcribing archival silent films.

The research into image, video and speech recognition have seen tremendous progress in recent years, due to the advances in deep learning. However, the vast majority of research conducted covers specific areas such as object recognition in images, and speech recognition; the progress related to visual recognition of speech and languages have been limited, somewhat due to the insufficient availability of training data.

Fortunately, there are vast amounts of data available on television and the Internet that can be used by machines to learn from, such as that shown in Figure 1.1. This thesis develops methods that can learn to recognise and understand visual speech and gestures from this large-scale data. Specific contributions are described in Section 1.3.

1.2 Key challenges

Ambiguities in audio-visual speech recognition. Visual speech recognition is a notoriously difficult task, and it is understood that even the best lip readers can only

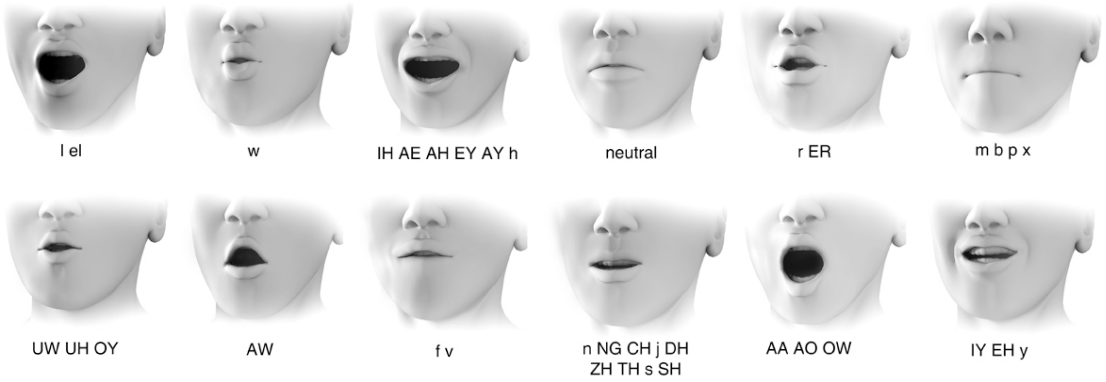

Figure 1.2: A single viseme can represent several different phonemes. Image adapted from Paulus (2013).

decipher less than half the words in most scenarios (Marschark and Spencer, 2010). This is due to the ambiguity at the word level arising from homophones – different characters that produce exactly the same lip sequence (*e.g.* 'p' and 'b') as can be seen in Figure 1.2. So suppose you see somebody say '*as I walk in the* ***park****, I hear my dog* ***bark***', you can only differentiate 'park' and 'bark' by the context. To a lesser extent, there are pairs of words in English that sound very similar, but look different on the lips, for example 'beam' and 'bean'. This presents two challenges: the first is how to combine the visual and auditory information, and the second is how to model complex long-term dependencies such as linguistic context.

Variability in speech and gestures. Automated speech and gesture recognition is challenging because of high intra- and inter-person variability (O'Shaughnessy, 2008). For example in speech, there are a number of sources for this variability, including: (i) regional accents; (ii) speaker physiology; (iii) speaking style; (iv) speed; (v) emotional state. Co-articulation also makes the tasks more challenging, since the transition from the previous and to the next word can affect the mouth shape in speech, and the trajectory of body parts in gestures.

Lack of training data. The most common form of machine learning is supervised learning. Many notable applications of machine learning in recent years have also been in this area, thanks to the availability of large datasets such as ImageNet (Deng

et al., 2009). However, such readily-available datasets do not exist in the field of visual speech or gesture recognition. This raises two challenges: (i) how to generate a corpus of labelled data from existing videos (*e.g.* from television or YouTube) with minimum human intervention; (ii) how to make use of abundant unlabelled data, when the amount of labelled data is limited.

Weak supervision in sign language recognition. Traditionally for machine learning applications relating to time series, the supervision is strong – either time aligned or, if not aligned, then complete. For example in audio-visual speech recognition, there is a one-to-one correspondence between the video and the audio, and hence the text in the transcript. However such correspondence does not exist in sign language videos. Therefore, training to recognise gestures in sign language from TV subtitles is extremely challenging, as the supervision would be both weak (the signs are not temporally aligned with the audio) and noisy (the occurrence of a word in the transcript does not imply that it will be signed). The question to consider here is how to design models that can learn to recognise short temporal signals from training data with such weak supervision.

1.3 Contributions

In this section, we list the main contributions made in this thesis.

Building a large-scale audio-visual corpus. (Chapter 3) Here, we develop a multi-stage pipeline for fully automated large-scale data collection from TV broadcasts. With this we generate a dataset with over a million word instances, spoken by over a thousand different people. This data is used in various experiments throughout this thesis.

Learning a synchronization network for lip motion and audio. (Chapter 4) We propose a two-stream ConvNet architecture and a self-supervised training strategy

that enables a joint embedding of the sound and the mouth images to be learnt from unlabelled data. The trained network is used to determine the *lip-sync error* in a video. We also apply the network to the task of active speaker detection, and we set a new state-of-the-art on a standard benchmark dataset.

Deep learning for word-level lip reading. (Chapter 5) In this chapter, we present various deep learning architectures that are able to effectively learn and recognise hundreds of words from the large-scale dataset generated in Chapter 3. We also demonstrate a recognition performance that exceeds the state of the art on a standard public benchmark dataset.

Deep learning for sentence-level lip reading. (Chapter 6) In this chapter, we develop an attention-based model that is able to lip read unconstrained natural language sentences, from 'in the wild' videos. This lip reading performance beats a professional lip reader on videos from BBC television, and we also demonstrate that if audio is available, then visual information helps to improve speech recognition performance.

Generating talking faces from audio. (Chapter 7) Here, we show that the self-supervised methods of Chapter 4 can be extended to the problem of generating videos of talking faces. The method takes still images of the target face and an audio speech segment as inputs; and produces a video of the target face lip synched with the audio. To achieve this we propose an encoder-decoder CNN model that uses a joint embedding of the face and audio to generate synthesised talking face video frames.

Word spotting in audio and sign language. (Chapter 8) This chapter presents a method to *recognise* and *localise* short temporal signals in image time series, where strong supervision is not available for training. We propose image encodings and ConvNet-based architectures to first recognise the signal, and then to localise the signal using back-propagation. The method is demonstrated for localising spoken

words in audio, and for localising signed gestures in British Sign Language (BSL) videos.

Speaker identification from audio. (Chapter 9) In this final chapter, we depart from the problem of 'what' is being said, and explore methods for recognising 'who' is saying it. Most of the previous works for speaker identification are limited to constrained conditions, due to the synthetic nature of existing datasets. This chapter presents a fully automated pipeline to generate a large scale text-independent speaker identification dataset collected 'in the wild', and propose a number of ConvNet-based architectures that outperform traditional baselines for speaker recognition.

1.4 Publications

The research conducted in this thesis has resulted in a number of peer-reviewed publications listed below in chronological order.

- J. S. Chung, A. Zisserman **"Signs in time: Encoding human motion as a temporal image"** ECCV Workshop, 2016. (Chung and Zisserman, 2016c)
- J. S. Chung, A. Zisserman **"Lip Reading in the Wild"** ACCV, 2016. *Oral presentation. Best student paper award.* (Chung and Zisserman, 2016a)
- J. S. Chung, A. Zisserman **"Out of time: automated lip sync in the wild"** ACCV Workshop, 2016. (Chung and Zisserman, 2016b)
- J. S. Chung, A. Senior, O. Vinyals, A. Zisserman **"Lip Reading Sentences in the Wild"** CVPR, 2017. *Oral presentation.* (Chung et al., 2017b)
- A. Nagrani[†], J. S. Chung[†], A. Zisserman **"VoxCeleb: a large-scale speaker identification dataset"** Interspeech, 2017. *Oral presentation. Best student paper award.* (Nagrani et al., 2017)

- J. S. Chung†, A. Jamaludin†, A. Zisserman **"You said that?"** BMVC, 2017. *Oral presentation.* (Chung et al., 2017a)

- J. S. Chung, A. Zisserman **"Lip Reading in Profile"** BMVC, 2017. (Chung and Zisserman, 2017)

† represents equal contributions between authors.

Chapter 2

Background

In this chapter we review the development in methods and applications relating to this thesis.

2.1 Deep learning

In this section we describe recent developments in deep learning, which have dramatically improved the state-of-the-art in many applications, such as image (Krizhevsky et al., 2012, Simonyan and Zisserman, 2015) and speech recognition (Graves and Schmidhuber, 2005, Graves et al., 2006a).

Deep learning can be seen as a form of hierarchical learning, where algorithms make use of multiple levels of simple but non-linear transformations to gradually transform the raw input into high level concepts. With multiple levels of such simple transformations, very powerful and complex functions can be learnt.

Neural networks are modelled after interactions of neurons in a human brain. For example, the architecture of image recognition CNNs is related to the mammalian visual cortex that has a layered structure (Hubel and Wiesel, 1962), and processes images in layers of increasing complexity – the first layer distinguishes basic attributes like lines and curves, while brain regions further down are sensitive to higher level

concepts, for instance, if the image is of a dog or a cat.

Deep learning methods require little engineering by hand, but generally require a large dataset to train from, and also substantial processing power.

2.1.1 Training with backpropagation.

A neural network consists of multiple layers of simple modules, which are subject to learning. These networks can be trained by stochastic gradient descent (SGD) using the backpropagation procedure. This means that the classifier is jointly trained with the feature extractors.

The backpropagation procedure involves using the chain rule to iteratively compute gradients for each layer, working backwards from the objective. Having computed the gradients for each layer, it is trivial to compute the gradients with respect to the weights of each module.

2.1.2 Convolutional Neural Networks (CNN).

Research has shown that ConvNets (LeCun et al., 1989, 1998) are easier to train and generalise better than other types of feedforward networks for many applications, particularly in computer vision.

ConvNets are obtained by stacking multiple layers of convolutional filter banks, each followed by a non-linear response function. Each convolutional layer takes an input which is a feature map, and outputs a new feature map which is connected to the input through a set of weights called filter bank. This is then passed through a non-linear activation function, most popular being ReLU (Glorot et al., 2011). All units in a feature map share the same filter bank – this design is beneficial for two reasons: reduction in the number of parameters, and invariance to the spatial location in the

input image and its features.

Convolutional layers are usually used together with pooling layers, for which popular functions are max pooling and average pooling. The pooling layers help downsample the feature maps to reduce dimensionality, and also help build invariance to the spatial location of the image.

Successful applications of CNNs go back several decades for a limited number of domains such as digit recognition (LeCun et al., 1989); however the breakthrough in performance came in recent years for many application such as object (Krizhevsky et al., 2012, Simonyan and Zisserman, 2015) and face (Parkhi et al., 2015, Schroff et al., 2015) recognition, thanks to the increased availability of computing power and datasets.

2.1.3 Recurrent Neural Networks (RNN).

Recurrent neural networks are powerful systems that are able to handle sequential inputs, such as speech and language. RNNs takes the inputs in a sequence, and maintains a 'state' in their hidden units that implicitly stores the relevant information about the history of the past inputs. However, for a long time, they were of limited use in the research community as they were difficult to train, mainly due to the vanishing and exploding gradient problems.

With advances in their architecture (El Hihi and Bengio, 1996, Hochreiter and Schmidhuber, 1997) and training strategies (Pascanu et al., 2013, Sutskever, 2013), RNNs have seen increasing success in many sequence-related applications. In particular, the introduction of Long Short Term Memory (LSTM) modules (Hochreiter and Schmidhuber, 1997) have proved very important, since its explicit memory module was good at modelling long-term dependencies, and also the design was less susceptible to the problem of vanishing and exploding gradients.

RNN architectures can take many different forms, but of particular relevance to the methods presented in this thesis are sequence-to-sequence models (Sutskever et al., 2014). This class of models are made up of an encoder network that generates a representation of the input sequence as the final state vector, and a decoder network that produces an output sequence, one element at a time. The network is often used with an attention mechanism (Bahdanau et al., 2015, Chorowski et al., 2015) that allows the decoder to 'peek into' the input at every output step. These have proven very effective for a number of applications, including machine translation (Bahdanau et al., 2015) and automatic speech recognition (Chan et al., 2015).

2.1.4 Self-supervised learning

Supervised learning has been the most prevalent paradigm in recent computer vision methods, but there is also a good deal of previous work on self-supervised representation learning, where raw data is used as its own source of supervision. One of the earlier adaptations is the work on auto-encoders (Hinton and Salakhutdinov, 2006), and there are a number of more recent applications on learning representations via data imputation. The work on predicting co-occurrence (Isola et al., 2015) and context by inpainting (Pathak et al., 2016) fall into this category. Zhang et al. (2016) uses cross-channel self-supervision to train a model that predicts RGB images given only single-channel greyscale images. Of close relevance to the methods developed in this thesis is the work of Wang and Gupta (2015) which trains Siamese networks (Chopra et al., 2005) to learn image representations from video, using the assumption that two patches from a continuous video track is probably of the same object.

2.2 Applications

This section reviews the application of machine learning methods that are relevant to this thesis, namely vision, speech and language.

2.2.1 Audio-visual speech recognition

Visual speech recognition. Research on visual speech recognition (lip reading) has a long history. A thorough survey of shallow (*i.e.* not deep learning) methods is given in the recent review (Zhou et al., 2014b), and will not be repeated in detail here. Many of the existing works in this field have followed similar pipelines which first extract spatio-temporal features around the lips (either motion-based, geometric-feature based or both), and then align these features with respect to a canonical template. For example, Pei et al. (2013), which holds state-of-the-art on many datasets, extracts the patch trajectory as a spatio-temporal feature, and then aligns these features to reference motion patterns.

A number of recent works have used deep learning methods to tackle problems related to lip reading. Koller et al. (2015) train an image classifier CNN to discriminate *visemes* (mouth shapes, visual equivalent of *phonemes*) on a sign language dataset where the signers mouth words. Similar CNN-based methods have been proposed by Noda et al. (2014) to predict *phonemes* in spoken Japanese. In the context of word recognition, Tamura et al. (2015) have used deep bottleneck features (DBF) to encode *shallow* input features such as LDA and GIF (Ukai et al., 2012). Similarly Petridis and Pantic (2016) use DBF to encode the image for every frame, and train a LSTM classifier to generate a word-level classification. Wand et al. (2016) use an LSTM with HOG input features to recognise short phrases. The shortage of training data in lip reading presumably contributes to the continued popularity of shallow features.

A recent work (Assael et al., 2016) uses a CNN and LSTM-based network and Connectionist Temporal Classification (CTC) (Graves et al., 2006b) to compute the labelling. This reports strong speaker-independent performance on the constrained grammar and 51 word vocabulary of the GRID dataset (Cooke et al., 2006). However, the method, suitably modified, should be applicable to longer, more general sentences.

Audio-visual speech recognition. The problems of audio-visual speech recognition (AVSR) and lip reading are closely linked, and as such, many of the existing works are based on similar methods. A number of papers have attempted to predict *phonemes* as a proxy task to speech recognition – for example, Mroueh et al. (2015) employ feed-forward Deep Neural Networks (DNNs) to perform phoneme classification on a non-public audio-visual dataset.

The use of HMMs together with hand-crafted or pre-trained visual features have proved popular – Tamura et al. (2015) encodes input images using DBF; Galatas et al. (2012) used DCT; and Noda et al. (2015) uses a CNN pre-trained to classify phonemes; all three combine these features with HMMs to classify spoken digits or isolated words. As with lip reading, there has been little attempt to develop AVSR systems that generalise to real-world settings.

Speech recognition. There is a wealth of literature on 'shallow' speech recognition systems that utilise separate components for acoustic and language-modelling functions (*e.g.* hybrid DNN-HMM systems), that we will not review here. We restrict this review to methods that can be trained end-to-end.

For the most part, prior work can be divided into two types. The first type uses CTC (Graves et al., 2006b), where the model typically predicts framewise labels and then looks for the optimal alignment between the framewise predictions and the output sequence. The weakness is that the output labels are not conditioned on each other.

The second type is sequence-to-sequence models (Sutskever et al., 2014) that first read all of the input sequence before starting to predict the output sentence. A number of papers have adopted this approach for speech recognition (Chorowski et al., 2014, 2015), and of which the most related to the work in this thesis is that of Chan et al. (2015) which proposes an elegant sequence-to-sequence method to transcribe audio

signal to characters. They utilise a number of the latest sequence learning tricks such as scheduled sampling (Bengio et al., 2015) and attention (Chorowski et al., 2015). We take many inspirations from this work.

Audio-visual datasets. One of the major obstacles to progress in this field has been the lack of suitable datasets (Zhou et al., 2014b). Table 2.1 gives a summary of existing public datasets. The amount of available data is far from sufficient to train scalable and representative models that will be able to generalise beyond the controlled environments and the very limited domains (*e.g.* digits and the alphabet).

Name	**Env.**	**Output**	**I/D**	**# class**	**# subj.**	**Best perf.**
AVICAR [a]	In-car	Digits	D	10	100	37.9% [h]
AVLetter [b]	Lab	Alphabet	I	26	10	43.5% [j]
CUAVE [c]	Lab	Digits	I	10	36	83.0% [k]
GRID [d]	Lab	Sent.	D	8.5*	34	79.6% [l]
OuluVS1 [e]	Lab	Phrases	I	10	20	89.7% [m]
OuluVS2 [f]	Lab	Phrases	I	10	52	73.5% [n]
OuluVS2 [f]	Lab	Digits	D	10	52	-
Modality [g]	Lab	Sent.	D	182	35	-

Table 2.1: Existing lip reading datasets. **I** for **I**solated (one word, letter or digit per recording); **D** for **D**elimited recording. The reported performance is on speaker-independent experiments. (* For GRID (Cooke et al., 2006), there are 51 classes in total, but the first word in a phrase is restricted to 4, the second word 4, etc. 8.5 is the average number of possible classes at each position in the phrase.) [a] Lee et al. (2004); [b] Matthews et al. (2002); [c] Patterson et al. (2002); [d] Cooke et al. (2006); [e] Zhao et al. (2009); [f] Anina et al. (2015b); [g] Czyzewski et al. (2017); [h] Fu et al. (2008); [j] Zhao et al. (2009); [k] Papandreou et al. (2009); [l] Wand et al. (2016); [m] Pei et al. (2013); [n] **?**.

2.2.2 Audio-to-video synchronisation

There is a large body of work on the audio to video synchronisation problem. However the majority of these are based on methods that are not available to the television receiver (*e.g.* embedding timestamps in the transport stream). Instead, we focus our review on computer vision methods that only rely on the audio-visual data.

Intermediate representations. *Phoneme* recognition is often used as a proxy task for solving the lip-sync problem. Lewis (1991), linear prediction is used to provide

phoneme recognition from audio, and the recognised phonemes are associated with mouth positions to provide lip-sync video. Morishima et al. (2002) classify the face parameters into *visemes*, and uses the *viseme* to *phoneme* mapping to obtain the synchronisation. Although Koster et al. (1994) and McAllister et al. (1997) do not explicitly classify the sounds into phonemes, their approaches are similar to those above in that they develop models by having the speaker record a set of vowels. Both Koster et al. (1994) and McAllister et al. (1997) correlate face parameters such as jaw position to the FFT of the sound signal. Zoric and Pandzic (2005) have used neural networks to tackle the problem. A multi-layer feedforward neural network is trained to predict the *viseme* from MFCC input vectors. A parametric face model is used for the visual processing.

Embeddings. More recent papers have attempted to find correpondence between speech and visual data without such labels. A number of approaches are based on canonical correlation analysis (CCA) (Bredin and Chollet, 2007, Sargin et al., 2007) or co-inertia analysis (CoIA) (Rúa et al., 2009) of audio and visual features (*e.g.* geometric parameters or 2D DCT features). The most related work to ours is that of Marcheret et al. (2015) that uses a Deep Neural Network (DNN)-based classifier to determine the time offset based also on pre-defined visual features (speech class likelihoods, bottleneck features, etc.) rather than learning the visual features directly.

2.2.3 Sign language recognition

Most of the work done on sign language recognition is at word-level, *i.e.* a single classification is produced for an input stream. It is therefore a particular application of action recognition.

One-shot learning. The amount of strongly labelled data in sign language gestures is limited, hence many papers have proposed one-shot methods where only single training example for every class is learnt. The method usually involves nearest neigh-

bour matching on a feature or combination of features. The best performing model from Krishnan and Sarkar (2013) extracts frame-wise HoG features, and applies dynamic time warping (Vintsyuk, 1968) to find the nearest gesture in the training examples. Wan et al. (2013) take a similar one-shot approach, but instead uses RGB-D frames to extract 3D EMoSIFT (a variant of SIFT) (Wan et al., 2014) features. Although it has been shown that these methods can produce a reasonable performance in person-specific gesture recognition, it is difficult for these models to generalise to other signers given the variability in how signs are performed. A number of papers also have attempted to learn sign language gestures using RGB-D data from Kinect (Cooper et al., 2012, Zafrulla et al., 2011), but we do not use the depth data in this project as the quantity of available data is limited.

Weakly supervised learning. A number of papers have attempted to learn gestures using the weakly labelled data from subtitled sign language TV broadcasts. Words are likely to appear in the subtitles where they are signed, though this is not always the case. Therefore, Buehler et al. (2009) split the data to 'positive' sequences where a word is likely to occur, and 'negative' where it is not. Using the pose estimate and the hand shape, a multiple instance learning method is used to find the signs that occur most frequently within the positive sequences. The extracted signs can be used to train a signer-independent classifier (Buehler et al., 2010). Cooper and Bowden (2009) use a similar approach using the positive and the negative bags, but learns only from the pose estimates, and uses a priori mining instead of MIL to extract the signs.

One-shot learning with weakly labelled reservoir. Pfister et al. (2014a) show that the performance of a gesture classifier can be improved by combining one-shot learning with a 'reservoir' of weakly supervised examples. The authors first train a one-shot detector using the Global Alignment Kernel (Cuturi, 2011) instead of dynamic time warping. Pose estimates and HoG of the hands are used as the features.

The detector is then used to find new samples of the learned sign - but this is restricted to a temporal window provided by the weak supervision. It is easier to learn an accurate classifier using this method than learning solely from the weakly labelled data. The new samples are then used to train a stronger classifier.

End-to-end learning with CNN. There are a number of recent CNN-based methods that are used to recognise signed gestures. Koller et al. (2016a,b) starts from a CNN pre-trained on a large database of labelled hand shapes, and combines this with an HMM to train a system that can recognise signs from continuous videos. The model is trained on German sign language (GSL) sentences with 'gloss' annotation (a morpheme-by-morpheme 'transcription' using English words; *e.g.* the sentence 'You are not cooking today' would be annotated as 'TODAY YOU-all neg-COOK'.) Similarly, Camgoz et al. (2017) proposes a CNN-LSTM model trained with CTC loss to recognise signs from continuous sentences. These methods can be trained end-to-end, however cannot be used for translation, since they rely on the monotonic alignment between the signs and the gloss sentence.

Datasets. There are a number of benchmark datasets in sign language recognition. Table 2.2 gives a summary of the datasets, of which WTH-PHOENIX and SIGNUM are the most popular. Both datasets have been transcribed with gloss annotation, which requires a significant amount of manual effort, hence the limitation in size. A number of BSL datasets also exist (such as that introduced by Pfister et al. (2014a)), but these lack gloss annotation, and are not publicly available.

Name	**Language**	**Output**	**# samples**	**# class**	**# subj.**
DGS Kinect 40 [a]	German	Words	3K	40	15
RWTH-PHOENIX [b]	German	Sentences	46K	1,200	9
SIGNUM [c]	German	Sentences	33K	450	25
Boston ASL LVD [d]	American	Words	10K	3,300	6
LSA64 [e]	Argentinian	Words	3K	64	10

Table 2.2: Existing sign language recognition datasets. [a] Lee et al. (2004); [b] Forster et al. (2012); [c] Von Agris et al. (2008); [d] Thangali et al. (2011); [e] Ronchetti et al. (2016).

2.2.4 Speaker identification and verification

Traditional methods. For a long time, speaker identification was the domain of Gaussian Mixture Models (GMMs) trained on low dimensional feature vectors (Reynolds and Rose, 1995, Reynolds et al., 2000). The state of the art in more recent times involves both the use of joint factor analysis (JFA) based methods which model speaker and channel subspaces separately (Kenny, 2005), and i-vectors which attempt to model both subspaces into a single compact, low-dimensional space (Dehak et al., 2011). Although state of the art in speaker recognition tasks, these methods all have one thing in common – they rely on a low dimensional representation of the audio input, such as Mel Frequency Cepstrum Coefficients (MFCCs). However, not only does the performance of MFCCs degrade rapidly in real world noise (Hansen et al., 2001, Yapanel et al., 2002), but by focusing only on the overall spectral envelope of short frames, MFCCs may be lacking in speaker-discriminating features (such as pitch information).

Deep learning methods. A number of recent papers have used neural network based methods for speaker identification. The most relevant to our methods is Lukic et al. (2016) which trains a vision-style CNN on voice spectrograms to identify speakers on the TIMIT database. Chen and Salman (2011) uses a Siamese feedforward DNN to discriminatively compare two voices, however this relies on pre-computed MFCC features rather than learning the features in an end-to-end manner. Yella et al. (2014) also uses a similar feedforward network to discriminatively train a speaker recognition system – here, the network also learns the features instead of using MFCCs. Heigold et al. (2016) proposes an end-to-end model with a LSTM feature extractor, and demonstrate significant benefits over a feedforward DNN on text-dependent speaker recognition task.

Datasets. Many existing datasets are obtained under controlled conditions, for ex-

ample: forensic data intercepted by police officials (van der Vloed et al., 2014), data from telephone calls (Hennebert et al., 2000), speech recorded live in high quality environments such as acoustic laboratories (Garofolo et al., 1993, Millar et al., 1994), or speech recorded from mobile devices (McCool and Marcel, 2009, Woo et al., 2006). The Forensic Comparison dataset (Morrison et al., 2015) consists of more natural speech but has been manually processed to remove extraneous noises and crosstalk. All the above datasets are also obtained from single-speaker environments, and are free from audience noise and overlapping speech.

Datasets obtained from multi-speaker environments include those from recorded meeting data (Carletta et al., 2005, Janin et al., 2003), or from audio broadcasts (Bell et al., 2015). These datasets usually contain audio samples under less controlled conditions. Some datasets contain artificial degradation in an attempt to mimic real world noise, such as those developed using the TIMIT dataset (Garofolo et al., 1993): NTIMIT, (transmitting TIMIT recordings through a telephone handset) and CTIMIT, (passing TIMIT files through cellular telephone circuits). The only freely available dataset curated from multimedia is the Speakers in the Wild (SITW) dataset (McLaren et al., 2016), which contains speech samples of 299 speakers across unconstrained or 'wild' conditions. This is a valuable dataset, but to create it the speech samples have been hand-annotated. Scaling it further, for example to thousands of speakers across tens of thousands of utterances, would require the use of a service such as Amazon Mechanical Turk (AMT). In the computer vision community AMT like services have been used to produce very large-scale datasets, such as ImageNet (Russakovsky et al., 2015).

Table 2.3 summarises existing speaker identification datasets. Besides lacking real world conditions, most of these datasets have been collected with great manual effort, other than (Bell et al., 2015) which was obtained by mapping subtitles and transcripts to broadcast data.

Name	Cond.	Free	# POI	# Utter.
ELSDSR [a]	Clean Speech	✓	22	198
MIT Mobile [b]	Mobile Devices	-	88	7,884
SWB [c]	Telephony	-	3,114	33,039
POLYCOST [d]	Telephony	-	133	1,285‡
ICSI Meeting Corpus [e]	Meetings	-	53	922
Forensic Comparison [f]	Telephony	✓	552	1,264
ANDOSL [g]	Clean speech	-	204	33,900
TIMIT [h] †	Clean speech	-	630	6,300
SITW [i]	Multi-media	✓	299	2,800
NIST SRE [j]	Clean speech	-	2,000+	∗
VoxCeleb	Multi-media	✓	**1,251**	**145,000**

Table 2.3: Comparison of existing speaker identification datasets. **Cond.:** Acoustic conditions; **POI:** Person of Interest; **Utter.:** Approximate number of utterances. †And its derivatives. ‡Number of telephone calls. ∗ varies by year. [a] Feng and Hansen (2005); [b] Woo et al. (2006); [c] Godfrey et al. (1992); [d] Hennebert et al. (2000); [e] Janin et al. (2003); [f] Morrison et al. (2015); [g] Millar et al. (1994); [h] Fisher et al. (1986); [i] McLaren et al. (2016); [j] Greenberg et al. (2013).

2.2.5 Visual speech synthesis

There are various works that proposed methods to generate or synthesise videos of talking heads from either audio or text sources.

The majority of the existing works are based on frame reselection from a video. Fan et al. (2015) propose a method based on a bi-directional LSTM that selects a target mouth region from a dictionary of saved target frames. The lower half of the face from the selected frame is then blend back into the background face. Similarly, Charles et al. (2016) train a model to select visemes based on the phonetic label of the audio, and enforces visual smoothness by matching the visual features of the last frame of one viseme to the first frame of the next, optimized using the Viterbi algorithm. The recent work of Suwajanakorn et al. (2017) train a recurrent neural network to predict the coordinates of key facial landmarks for every frame given the audio, and fills the texture based on the landmarks using frame reselection. The paper proposes a series of post-processing steps like video re-timing and jaw smoothing to produce realistic images.

Taylor et al. (2017) also use a phonetic-based method – the audio is first transcribed into a phonetic sequence, from which the animation parameters are generated. The final image here is however generated by a CG animation model, rather than by frame reselection.

Also of relevance is the work of Garrido et al. (2015) which describes a method to transfer the mouth shapes from the video of the dubber to the face in the target video using a 3D model. However, this method requires the video footage of the dubber's mouth saying the speech segment, whereas our method learns the relationship between the sound and the mouth shapes.

Natural image synthesis using CNNs. Visual speech synthesis is closely related to the problem of image synthesis, which has seen significant advances in recent years.

Generative adversarial networks (GANs) proposed by Goodfellow et al. (2014) have proven extremely effective in generating natural-looking images. A GAN has two modules, one network (typically a deconvolutional network) to generate an image from a latent variable, and another to discriminate if the image is real or fake. An extension to this model is Conditional GANs (Mirza and Osindero, 2014), which can generate images conditioned on auxiliary information, such as a class label.

Another successful approach is based on RNNs – in a method dubbed PixelRNN, van den Oord et al. (2016b) proposes training an LSTM to predict the value of the next pixel, given the previous pixel, starting from a corner. The method comes with a high computation cost as each state needs to be computed sequentially. The paper also introduces a CNN-based variant named PixelCNN, which reduces computational complexity by using standard convolutional layers to capture a bounded receptive field and computing features for all pixel positions at once. Conditional PixelCNN (van den Oord et al., 2016a) extends this architecture such that the model can be conditioned on any vector (*e.g.* descriptive labels or latent embeddings), which is

relevant to the work on talking face generation in this thesis.

The recent work of Chen and Koltun (2017) proposes Cascaded Refinement Network that generates realistic-looking images from pixel-wise semantic layout. They use a 'content representation' loss function that has been used previously in image style transfer works (Gatys et al., 2016) – the loss forces the network to match activations of a pre-trained CNN between the generated image and the ground truth, which demonstrates significant benefits over an image-space loss.

2.3 Conclusion

This chapter covered a selection of relevant literature on deep learning methods and their applications. We have observed an annual increase in performance in computer vision and speech recognition as new architectures and training regimes have been developed (Chan et al., 2015, Graves et al., 2006a, He et al., 2015, Hu et al., 2017, Krizhevsky et al., 2012, Simonyan and Zisserman, 2015, Szegedy et al., 2015), and have also witnessed deep learning methods being applied to new tasks such as synthesising natural images (Goodfellow et al., 2014). In this thesis, we build on these advances to develop new deep learning-based methods for recognising human communications.

Chapter 3

Building a large-scale audio-visual corpus

This chapter describes our multi-stage pipeline for automatically collecting and processing a large-scale audio-visual speech recognition dataset. We start from television programs from the BBC, since they contain all of the key ingredients to make the dataset – the audio of spoken sentences, the corresponding videos of the mouth motions, and the text in the closed caption (subtitles). Using the fully automated pipeline we have been able to extract thousands of hours of spoken text covering an extensive vocabulary of thousands of different words, with over 1 million word instances, and over a thousand different speakers. This corpus is used for a range of experiments throughout this thesis.

The key ideas are to: (i) obtain a temporal alignment of the spoken audio with a text transcription (broadcast as subtitles with the program). This in turn provides the time alignment between the visual face sequence and the words spoken; (ii) obtain a spatio-temporal alignment of the lower face for the frames corresponding to the word sequence; and, (iii) determine that the face is speaking the words (*i.e.* that the words are not being spoken by another person in the shot). The pipeline is summarised in Figure 3.2 and the individual stages are discussed in detail in the following sections.

Figure 3.1: A sample of speakers in our dataset.

3.1 Program selection

We require programs that have a changing set of talking heads, so choose news and current affairs, rather than dramas with a fixed cast. Table 3.1 lists the programs. There is a significant variation of format, viewpoint and content across the programs – from the regular news (*e.g.* BBC News, World News) where a single speaker is talking directly at the camera most of the time, to panel debates and talk shows (*e.g.* Question Time, The One Show) where the speakers might often shift their attention or look at each other, and so are seen in profile. There are a few people who appear repeatedly in the videos (*e.g.* news presenter or the host), but the large majority of participants change every episode as can be seen in Figure 3.1. This variety enables models trained on this corpus to generalise well across different viewpoints, speakers and domains.

Channel	**Series name**	**Description**
BBC 1 HD	News	Regular news
BBC 2 HD	World News	Regular news
BBC 1 HD	Breakfast	Regular news
BBC 1 HD	Newsnight	Current affairs debate
BBC 2 HD	Question Time	Current affairs debate
BBC 1 HD	The One Show	Magazine show

Table 3.1: List of programs that appear in our audio-visual corpus.

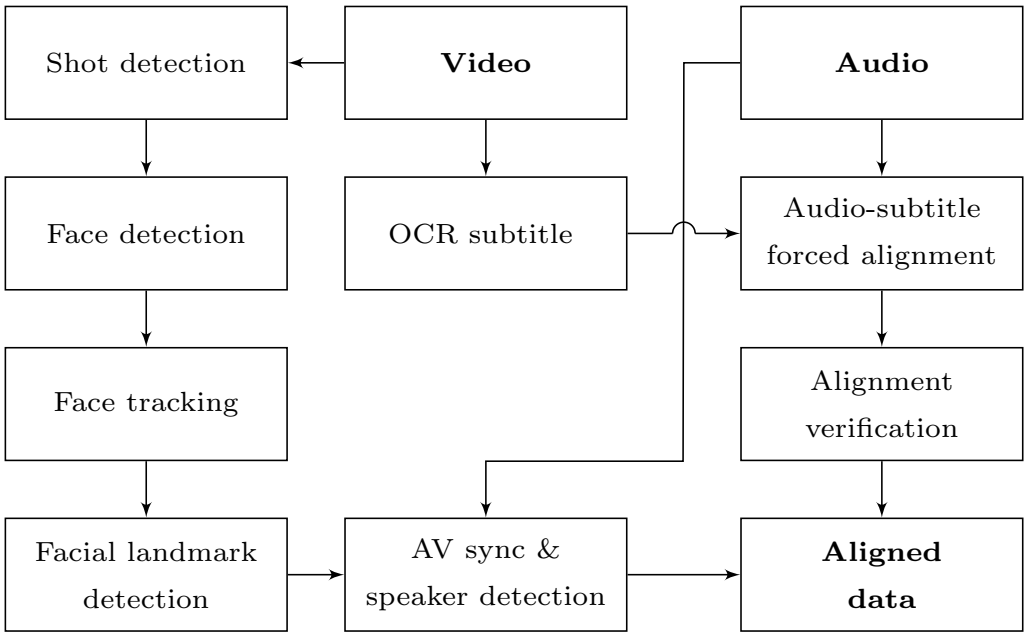

Figure 3.2: Pipeline to generate the text and visually aligned dataset. Timings are for a one-hour video.

3.2 Audio processing

Text alignment. We require the alignment between the audio and the subtitle in order to get a timestamp for every word that is being spoken in the videos. The BBC transmits subtitles as bitmaps rather than text, therefore subtitle text is extracted from the broadcast video using standard OCR methods (Buehler et al., 2009, Everingham et al., 2006). The subtitles are not time-aligned, and also not verbatim as they are generated live. The Penn Phonetics Lab Forced Aligner (Hermansky, 1990, Yuan and Liberman, 2008) (based on the open-source HTK toolbox (Woodland et al., 1995)) is used to force-align the subtitle to the audio signal. The aligner uses the Viterbi algorithm to compute the maximum likelihood alignment between the audio (modelled by PLP features (Rubin et al., 2013)) and the text. The audio is chunked at long pauses to increase efficiency. This method of obtaining the alignment has significant performance benefits over regular speech recognition methods that do not use prior knowledge of what is being said.

Figure 3.3: Programs in the training set. **Clockwise from left**: 'BBC News', 'BBC World News at One', 'The One Show', 'Newsnight', 'Question Time', 'Breakfast'.

Text alignment verification. The alignment result, however, is not perfect for a number of reasons, including: (1) the method often misses words that are spoken too quickly; (2) the subtitles are not verbatim; (3) the acoustic model is only trained to recognise American English. The noisy labels are filtered by double-checking against the commercial IBM Watson Speech to Text service. For generating the *training* set in Chapters 5 and 6, we do not filter the labels using this method because this leads to the rejection of many correct alignments (the agreement between the IBM Watson transcript and the forced alignment is only around 45%, whereas the accuracy of forced alignment is over 90%), so we trade-off some quality for quantity in training.

Figure 3.4: Subtitles on BBC TV. **Left**: 'Question Time', **Right**: 'BBC News at One'.

3.3 Video processing

Shot boundary detection. The shot boundaries are determined to find the within-shot frames for which face tracking is to be run. This is done by comparing color histograms across consecutive frames (Lienhart, 2001).

Face detection and tracking. The CNN face detector based on the Single Shot MultiBox Detector (SSD) (Liu et al., 2016) is used to detect face appearances on every frame of the video (Figure 3.5 left). As with most face detection methods, this results in many false positives and some missed detections. In a similar manner to (Everingham et al., 2006), all face detections of the same person are grouped across frames using a KLT tracker (Tomasi and Kanade, 1992) (Figure 3.5 middle). If the track overlaps with face detections on the majority of frames, it is assumed to be correctly tracking the face.

Facial landmark detection. Facial landmarks are needed to determine the mouth position for cropping. The landmarks are determined in every frame of the face track using the method of Kazemi and Sullivan (2014b) (Figure 3.5 right).

Facial pose estimation. In order to facilitate the testing of multi-view models, we divide the data into five pose categories based on the yaw-rotation of the face: (1) left profile; (2) left three-quarter; (3) frontal; (4) right three-quarter; (5) right profile. This is done using a ResNet-based pose regressor, trained on the CASIA-

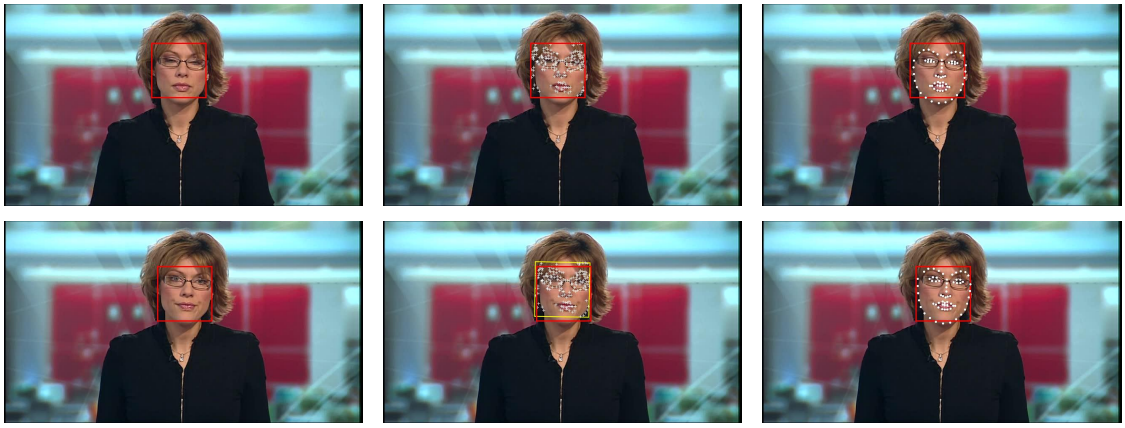

Figure 3.5: **Left:** Face detections; **Middle:** KLT features and the tracked bounding box (in yellow); **Right:** Facial landmarks.

WebFace dataset (Yi et al., 2014). The network has been trained to classify cropped face images into one of the above five categories. Examples belonging to each class are given in Figure 3.6.

Figure 3.6: Examples from each pose category. **Top row**: left profile; **2nd row**: left three-quarter; **3rd row**: frontal; **4th row**: right three-quarter; **Bottom row**: right profile.

Active speaker verification. The final step of the data processing pipeline is to identify who is speaking in the videos. Here, we use the ConvNet-based active speaker detection method described in Chapter 4. The network estimates the correlation between the audio track and the mouth motion of the video. This method is also able

to reject the clips that contain dubbing or voice-over.

3.4 Statistics and conclusion

Using this pipeline we have been able to generate a large audio-visual corpus aligned with spoken text covering an extensive vocabulary and many different speakers. The statistics of the data are given in Table 3.2.

Series name	**# vid.**	**# words**	**# facetracks**	**Total duration**	**Yield**
News	1,712	3.20M	70K	3.17M	33%
World News	361	0.57M	12K	0.65M	31%
Breakfast	308	3.45M	80K	3.41M	32%
Newsnight	605	2.16M	44K	1.68M	43%
Question Time	195	1.03M	35K	0.70M	47%
The One Show	259	0.25M	11K	0.54M	15%
Total	3,440	10.7M	253K	10.2M	34%

Table 3.2: Statistics of the processed corpus. The duration is in seconds. The yield is the proportion of useful face appearance relative to the total length of video. A useful face appearance is one that appears continuously for at least 5 seconds, with the face being that of the speaker.

This corpus is used for a range of experiments in this thesis – in Chapter 4 for audio-to-video synchronisation, Chapters 5 and 6 for lip reading, and Chapter 7 for visual speech synthesis. Moreover, the methods developed in this chapter forms the basis for the speaker recognition pipeline introduced in Chapter 9.

Chapter 4

Learning a synchronization network for lip motion and audio

Audio to video synchronisation (or lack of it) is a common problem in TV broadcasting. In television, a lip-sync error of up to several hundred milliseconds is not uncommon. The video usually lags the audio if the cause of the error is in the transmission. These errors are often noticeable – the threshold for detectability by an average viewer is around -125 ms (the audio lags the video) to +45 ms (the audio leads the video) (Peregudov et al., 2005).

In film production, audio to video synchronisation is a routine task, as the audio and the video are typically recorded using different equipment. Consequently, many solutions have been developed in this industry, the clapperboard being the most traditional one. Modern solutions use timecodes or sometimes time warping between the audio from the camera's built-in microphone and the external microphone, but it is not common to use the visual content as a guide to alignment.

Our objective in this chapter is to develop a *language independent* and *speaker independent* solution to the lip-sync problem, using only the video and the audio streams that are available to the TV viewer. The key contributions are the ConvNet architecture, and the data processing pipeline that enables a joint embedding between the sound and the mouth shapes to be learnt discriminatively from unlabelled videos, us-

ing cross-modal self-supervision. We first train this model on the frontal-view videos, and use a curriculum learning strategy to extend this method to profile views. To our knowledge, we are the first to end-to-end train a working AV synchronisation system.

This solution is of relevance to a number of different applications. In this chapter, we demonstrate that the method can be applied to two different tasks: (i) determining the *lip-sync error* in videos; (ii) detecting the speaker in a scene with multiple faces; both of which are crucial in building the dataset described in Chapter 3. In speaker detection, our results exceed the state-of-the-art on a public benchmark dataset (Chakravarty and Tuytelaars, 2016).

4.1 The SyncNet model

This section describes both the representations and network architectures for both the audio and the video inputs. The network ingests 0.2-second clips of each data type. In the dataset (Section 4.2), no explicit annotation (*e.g.* phonemes labels, or the precise time offset) is given for the audio-video data, however we make the assumption that in television broadcasts, the audio and the video are *usually* synced in single-speaker videos with a front-facing speaker.

The network consists of two asymmetric streams for audio and video, each of which is described below.

4.1.1 Audio stream

The input audio data are MFCC values. This is a representation of the short-term power spectrum of a sound on a non-linear mel scale of frequency. 13 mel frequency bands are used at each time step. The features are computed at a sampling rate of 100 Hz, giving 20 time steps for a 0.2-second input signal.

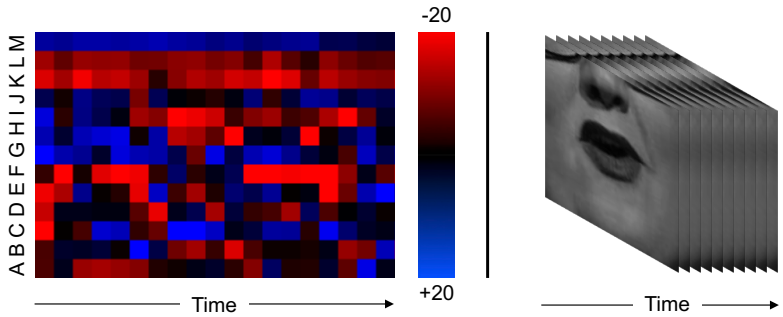

Figure 4.1: **Input representations**. **Left:** temporal representations as heatmaps for audio. The 13 rows (A to M) in the audio image encode each of the 13 MFCC features representing powers at different frequency bins from 0 Hz to 8 KHz. **Right:** Grayscale images of the mouth area.

Representation. The audio is encoded as a heatmap image representing MFCC values for each time step and each mel frequency band (see Figure 4.1). Previous work (Geras et al., 2015) has also attempted to train image-style ConvNet for similar inputs.

Architecture. We use a convolutional neural network inspired by those designed for image recognition. Our layer architecture (Figure 4.2) is based on VGG-M (Chatfield et al., 2014), but with modified filter sizes to ingest the inputs of unusual dimensions. VGG-M takes a square image of size 224×224 pixels, whereas our input size is 20 pixels (the number of time steps) in the time-direction, and only 13 pixels in the other direction (so the input image is 13×20 pixels).

4.1.2 Visual stream

Representation. The input format to the visual network is a sequence of mouth regions as grayscale images, as shown in Figure 4.1. The input dimensions are 111×111×5 (W×H×T) for 5 frames, which corresponds to 0.2-seconds at the 25 Hz frame rate.

Architecture. The visual stream is also based on the VGG-M model, but the *conv1* filter has been modified to ingest the 5-channel input instead of 1. This architecture

is closely related to the Early Fusion model in Chapter 5, which is compact and fast to train.

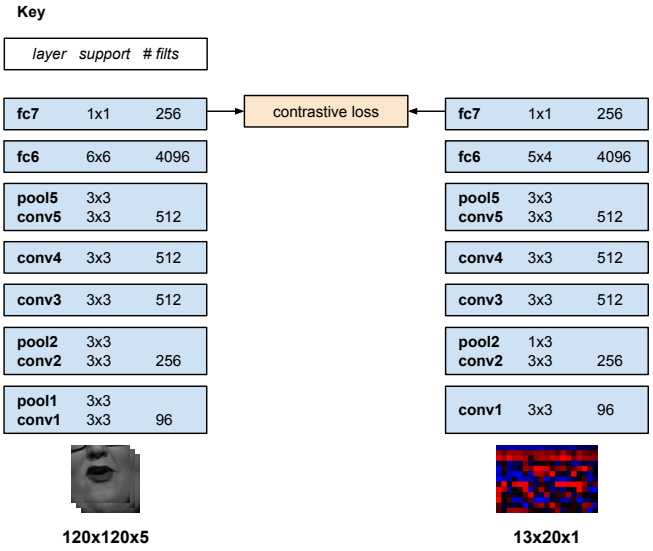

Figure 4.2: Two-stream ConvNet architecture. Both streams are trained simultaneously.

4.1.3 Loss function

The training objective is that the output of the audio and the video networks are similar for *positive* pairs, and different for *negative* pairs. Specifically, the Euclidean distance between the network outputs is minimised or maximised. We propose to use the contrastive loss (Equation 4.1), originally proposed for training Siamese networks (Chopra et al., 2005). v and a are fc_7 vectors for the video and the audio streams, respectively. $y \in [0, 1]$ is the binary similarity metric between the audio and the video inputs.

$$E = \frac{1}{2N} \sum_{n=1}^{N} (y_n)\, d_n^2 + (1 - y_n) \max\left(margin - d_n, 0\right)^2 \tag{4.1}$$

$$d_n = ||v_n - a_n||_2 \tag{4.2}$$

An alternative considered was to approach the problem as one of classification (on-sync/ off-sync, *or* into different offset bins using synthetic data), however we were

unable to achieve convergence.

4.1.4 Training

The training procedure is an adaptation of the usual procedure for a single-stream ConvNet (Krizhevsky et al., 2012, Simonyan and Zisserman, 2015) and inspired by (Chopra et al., 2005, Simonyan and Zisserman, 2014). However our network is different in that it consists of non-identical streams, two independent sets of parameters and inputs from two different domains. The network weights are learnt using stochastic gradient descent with momentum. The parameters for both streams of the network are learnt simultaneously.

Data augmentation. Applying data augmentation often improves validation performance and reduces overfitting in ConvNet image classification tasks (Krizhevsky et al., 2012). For the audio, the amplitude is randomly altered in the range of $\pm 10\%$. We do not make changes to the audio playback speed, as this could affect the important timing information. For *negative* examples only, we take random crops in time. For the video, we apply the standard augmentation methods used on the ImageNet classification task by (Krizhevsky et al., 2012, Simonyan and Zisserman, 2015) (*e.g.* random cropping, flipping, colour shift). A single transformation is applied to all video frames in a single clip.

Details. Our implementation is based on the MATLAB toolbox MatConvNet (Vedaldi et al., 2014). The network is trained with batch normalisation (Ioffe and Szegedy, 2015). A learning rate of 10^{-2} to 10^{-4} is used, which is slower than that typically used for training a ConvNet with batch normalisation. The training was stopped after 20 epochs, or when the validation error did not improve for 3 epochs, whichever was sooner.

4.2 Dataset

Figure 4.3: Still images of BBC News videos.

In this section, we describe the strategy to automatically generate data samples for training the lip synchronisation system.

We start from a subset of the audio-visual data described in Chapter 3, before the active speaker detection stage of the pipeline. The training, validation and test sets are divided in time, and the dates of videos corresponding to each set are shown in Table 4.1.

Set	**Dates**	**# pairs**	**# hours**
Train	01/07/2013 - 31/08/2015	3.70M	606
Val	01/09/2015 - 31/12/2015	0.31M	42
Test	01/01/2016 - 31/05/2016	0.35M	47

Table 4.1: Dataset statistics: recording dates, and number of lip-synched audio-video training pairs, number of hours of facetrack.

4.2.1 Sampling strategy

The training strategy relies on two initial assumptions: (i) the audio and the video streams in the majority of broadcast TV videos are *in-sync*; (ii) most of the front-facing faces in the single-person scenes are speaking.

Positive audio-video pairs are generated by taking a 5-frame video clip and the corresponding audio clip. Only the audio is randomly shifted by up to 2 seconds in order to generate synthetic *negative* audio-video pairs. This is illustrated in Figure 4.4. For

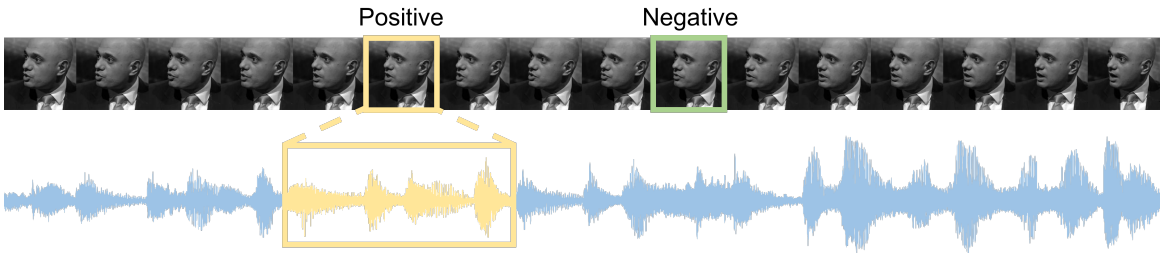

Figure 4.4: The process of obtaining *positive* and *negative* audio-video pairs.

the negative examples, We take the audio from the same clip, so that the network learns to recognise the alignment, rather than the correlation between the face and the voice.

Refining the training data. The training data generated using the proposed method is noisy in that it contains videos in which the voice and the mouth shapes do not correlate (*e.g.* dubbed videos) or are off-sync.

To overcome this problem, we train the network using the following strategy:

Stage 1: Train the network using this 'noisy' data.

Stage 2: Reject the false positives in the training set by rejecting positive pairs with a distance over a threshold.

Stage 3: Re-train the network on the remainder of the data.

4.3 Extending the model to profile views

In this section, we extend SyncNet from the frontal to the profile faces, using a curriculum learning strategy described in Section 4.3.1.

4.3.1 Curriculum learning

The training of the frontal-only SyncNet used the assumption that the majority of faces in the dataset are speaking. Whilst this may be the case for the news programmes it was trained on (Figure 4.5 left), this assumption cannot be used to bootstrap the multi-view model as there would be too much noise (in the form of non-speaking faces) for the network to learn the salient information. For example in a scene such as Figure 4.5 right, only one of these faces would be speaking at any one point. To circumvent this problem, we start with the frontal SyncNet trained on the news programmes, and adopt a curriculum learning approach (Figure 4.6) that gradually increases the working angle of the active speaker detection system.

Figure 4.5: **Left:** Still image from 'BBC News'; **Right:** Still image from 'The One Show'.

Stage 1. Frontal view. The first stage is to train the synchronisation network for the frontal faces (view 3 from Section 3.3) as described in Sections 4.1 and 4.2.

Stage 2. Three-quarter view. The network trained in Stage 1 is used to determine the active speaker on the three-quarter view (views 2 and 4) face tracks. The speaking tracks from these views are added to the training data; and the synchronisation network is re-trained.

Stage 3. Profile view. As before, the network in Stage 2 is used to perform speaker detection on the profile view (views 1 and 5) tracks. The speaking tracks are added to the training data and the network is re-trained.

Figure 4.6: Curriculum learning strategy. **ASD**: Active Speaker Detection.

4.4 Experiments

In this section we use the trained network to determine the lip-sync error in videos. The 256-dimensional fc_7 vectors for each stream are used as features representing the audio and the video. To obtain a (dis)similarity metric between the signals, the Euclidean distance between the features is taken. This is the same distance function that is used at training time (Equation 4.2). The histogram (Figure 4.7) shows the distribution of the metric.

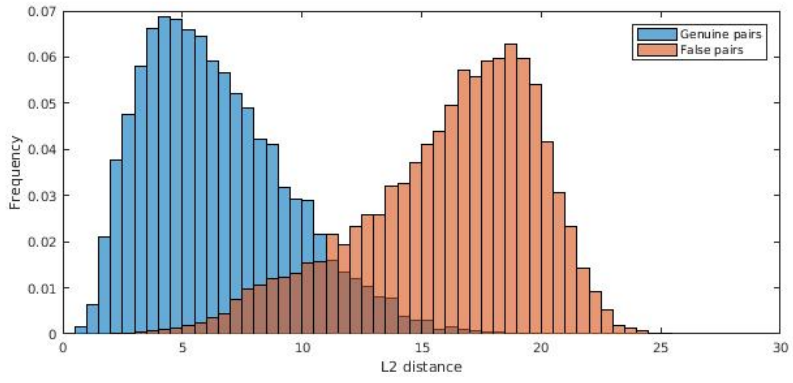

Figure 4.7: The distribution of Euclidean distances for *positive* and *negative* audio-video pairs, using a single 0.2-second sample. Note that this is on the noisy validation data that may include clips of non-speakers or dubbed videos.

Evaluation. We report Equal Error Rates on the labelled validation set in Table 4.2. The data is in the same format as used in training – the correct audio-video pairs for

positives, and artificially shifted audio for negatives.

View	Multi-view model	Frontal model
Frontal	13.6%	**13.2%**
Three-quarter	**14.8%**	17.1%
Profile	**16.2%**	21.7%

Table 4.2: **Equal Error Rates** (synched/ not synched) on the validation set, using **single 0.2-second samples**. Note that not every sample in the validation set contains discriminative information.

4.4.1 Determining the lip-sync error

To find the time offset between the audio and the video, we take a sliding-window approach. For each sample, the distance is computed between one 5-frame video feature and all audio features in the ± 1 second range. The correct offset is when this distance is at a minimum. Typical response plots are shown in Figure 4.9.

Since not every sample in a clip contain discriminative information (*e.g.* the person might be taking a breath at a particular moment), multiple samples are taken for each clip, and then averaged. Averaged over a 100-frame window, the method shows over 99% accuracy.

Experiments were also performed on a sample of foreign language videos (Figure 4.8), to show that our method works across different languages, and have found the qualitative results to be very strong.

Figure 4.8: Images of foreign language videos that were used for testing.

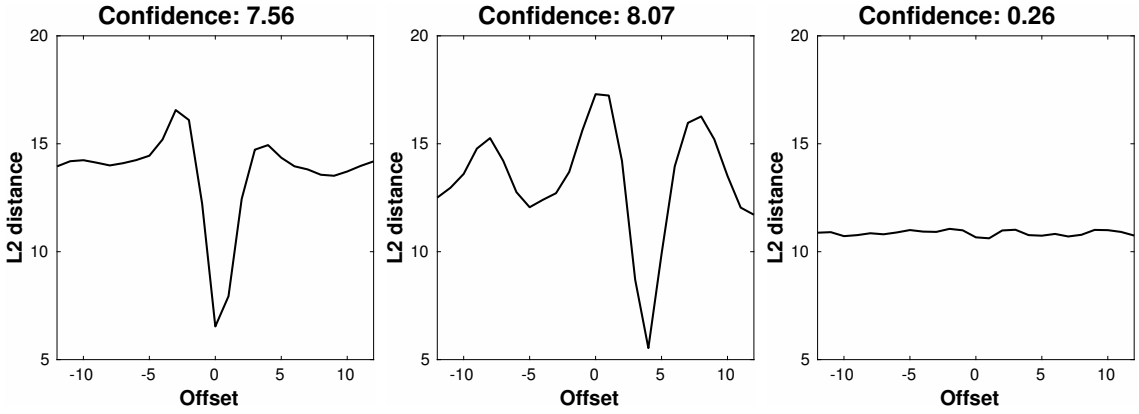

Figure 4.9: Mean distance between the audio and the video features for different offset values, averaged over a clip. The actual offset lies at the trough. The three example clips shown here are for different scenarios. **Left:** synchronised AV data; **Middle:** the audio leads the video; **Right:** the audio and the video are uncorrelated.

4.4.2 Application: active speaker detection

The problems of AV synchronisation and active speaker detection are closely related in that the correspondence between the video and the accompanying audio must be established. Therefore, the synchronisation method can be extended to determine the speaker in a scene where multiple faces are present. We define the confidence score of a time offset (synchronisation error) as the difference between the minimum and the median of the Euclidean distances (*e.g.* this value is around 6 to 7 for both plots in Figure 4.9). In a multi-subject scene, the speaker's face is naturally the one with the highest correspondence between the audio and the video. A non-speaker should have a correlation close to zero and therefore also a very low score.

Unlike the uni-modal methods for active speaker detection that rely on the lip motion only, our method also can detect cases where the person is speaking, but is uncorrelated to the audio (*e.g.* in dubbed videos).

Evaluation. We test our method using the dataset (Figure 4.10) and the evaluation protocol of Chakravarty and Tuytelaars (2016). The objective is to determine who the speaker is in a multi-subject scene.

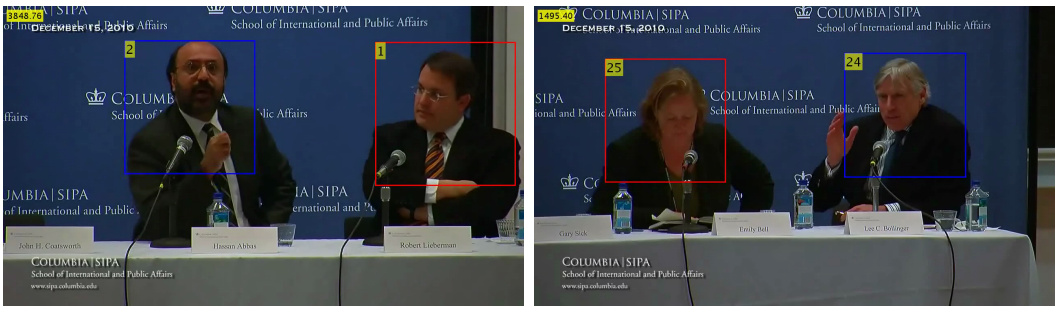

Figure 4.10: Still images from the Columbia dataset (Chakravarty and Tuytelaars, 2016).

The dataset contains 6 speakers, of which 5 (Bell, Bollinger, Lieberman, Long, Sick) are used for testing. A score threshold is set using the annotations on the remaining speaker (Abbas), at the point where the ROC curve intersects the diagonal (the equal error rate).

We report the F_1-scores in Table 4.3. The scores for each test sample are averaged over a 10-frame or 100-frame window. The performance is almost perfect for the 100-frame window. The disadvantage of increasing the size of the averaging window is that the method cannot detect examples in which the person speaks for a very short period; though that is not a problem in this case.

Method	Existing [a]		Ours	
Window	**10**	**100**	**10**	**100**
Bell	82.9%	90.3%	93.7%	**100%**
Bollinger	65.8%	69.0%	83.4%	**100%**
Lieberman	73.6%	82.4%	86.8%	**100%**
Long	86.9%	96.0%	97.7%	**99.8%**
Sick	81.8%	89.3%	86.1%	**99.8%**

Table 4.3: F_1-scores on the Columbia speaker detection dataset. [a] Chakravarty and Tuytelaars (2016): the results have been digitised from Figure 3b of their paper, and are accurate to around ±0.5%.

4.5 Conclusion

Using this method, we are able to train a two-stream network that learns an embedding of the audio and the lip motion, and provides a robust method of correcting the

lip-sync error, and determining the active speaker in multi-speaker scenes.

The method does not require annotation of the training data, unlike some previous works that are based on phoneme recognition. We train on audio-video pairs, and the advantage of this approach is that the amount of available data is virtually infinite, and the cost of obtaining it is minimal (almost any video of speech downloaded from the Internet can be used for training).

The key assumption is that the majority of the videos that we download are approximately synced, although some videos may have lip-sync errors. ConvNet loss functions and training are generally tolerant to the data being somewhat noisy.

As shown in Chapter 5, the visual stream of this network generates excellent features for the task of lip reading – on the LRW and OuluVS2 (Anina et al., 2015a) datasets, single-layer classifiers trained on the SyncNet features have outperformed networks trained end-to-end on the task. This is presumably because the SyncNet is trained on an extremely large amount of audio-visual data, whereas this is not feasible for lip reading.

Chapter 5

Deep learning for word-level lip reading

Lip-reading, the ability to understand speech using only visual information, is a very attractive skill. It has clear applications in speech transcription for cases where audio is not available, such as for archival silent films or (less ethically) off-mike exchanges between politicians and celebrities (the visual equivalent of open-mike mistakes). It is also complementary to the audio understanding of speech, and indeed can adversely affect perception if audio and lip motion are not consistent as evidenced by the McGurk effect (McGurk and MacDonald, 1976). For such reasons, lip-reading has been the subject of a vast research effort over the last few decades. It has also been the subject of comedy sketches, e.g. Seinfeld "The Lip Reader", and its ambiguity and challenge can be exploited to replace/overdub actual speech, *e.g.* in the YouTube channel "Bad Lip Reading".

Our objective in this chapter is a scalable approach to large lexicon *speaker independent* lip-reading. Furthermore, we aim to recognize words from *continuous speech*, where words are not segmented, and there may be co-articulation of the lips from preceding and subsequent words.

Apart from the problem of homophones (described in Chapter 1.4), lip-reading is a challenging problem in any case due to intra-class variations (such as accents, speed

of speaking, mumbling), and adversarial imaging conditions (such as poor lighting, strong shadows, motion, resolution, foreshortening, etc.).

The usual approach to inference for temporal sequences is to employ sequence models such as Hidden Markov Models or Recurrent Neural Networks (e.g. LSTMs). For lip-reading such models can be employed for predicting individual characters or phonemes. In contrast, we investigate using Convolutional Neural Networks (CNNs) for directly recognizing individual *words* from a sequence of lip movements.

Clearly, visual registration is an important element to consider in the design of the networks. Typically, the imaged head will move in the video, either due to actual movement of the head or due to camera motion. One approach would be to tightly register the mouth region (including lips, teeth and tongue, that all contribute to word recognition), but another is to develop networks that are tolerant to some degree of motion jitter. We take the latter approach, and do not enforce tight registration.

In this chapter, we develop CNN architectures for classifying multi-frame time series of lips. In particular we propose and compare different input and temporal fusion architectures, and discuss their pros and cons (Section 5.1). We analyse the performance and ambiguity of the resulting classifications in Section 5.3.

As discussed in the related work below, in these three aspects: speaker independence, learning from continuous speech, and lexicon (vocabulary) size, we go far beyond the current state of the art. We also exceed the state of the art in terms of performance, as is also shown in Section 5.3 by comparisons on the standard OuluVS benchmark dataset (Anina et al., 2015b).

5.1 Models

The task for the network is to predict which words are being spoken, given a video of a talking face. The input format to the network is a sequence of mouth regions, as shown in Figure 5.3. Previous attempts at visual speech recognition have relied on very precise localisation of the facial landmarks (the mouth in particular); our aim is learn from more noisy data, and tolerate some localisation irregularities both in position and in time.

5.1.1 Architectures

We cast the problem as one of multi-way classification, and so base our architecture on ones designed for image classification (Chatfield et al., 2014, Krizhevsky et al., 2012, Simonyan and Zisserman, 2015). In particular, we build on the VGG-M model (Chatfield et al., 2014) since this has a good classification performance, but is much faster to train and experiment on than deeper models, such as VGG-16 (Simonyan and Zisserman, 2015). We develop and compare models that differ principally in how they 'ingest' the T input frames (where here T= 25 for a 1 second interval). These variations take inspiration from previous work on human action classification (Ji et al., 2013, Karpathy et al., 2014, Ng et al., 2015, Tran et al., 2015). Apart from these differences, the architectures share the configuration of VGG-M, and this allows us to directly compare the performance across different input designs.

We next describe the three architectures, summarized in Figure 5.1, followed by a discussion of their differences. Their performance is compared in Section 5.3.

Early Fusion (EF). The network ingests a 25-channel image, where each of the channels encode an individual frame in *greyscale*. The layer structure for the subsequent layers is identical to that of the regular VGG-M network. This method is related to the Early Fusion model in (Karpathy et al., 2014), which takes *colour* im-

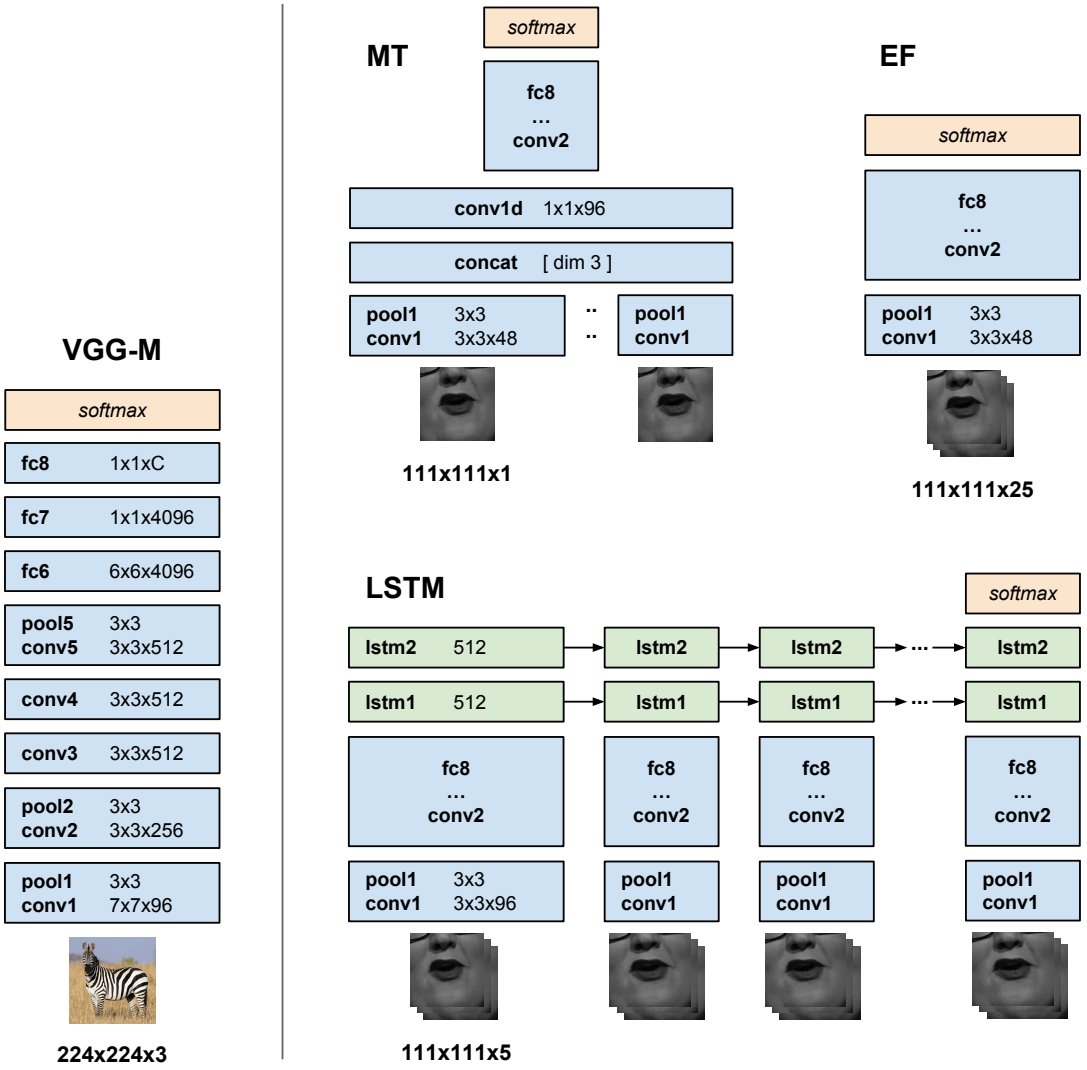

Figure 5.1: **Left:** VGG-M architecture that is used as a base. **Right:** Network architectures for lip reading.

ages and uses a T×3-channel convolutional filter at *conv1*. We did experiment with 25×3-channel colour input, but found that the increased number of parameters at *conv1* made training difficult due to overfitting (resulting in validation performance that is around 5% weaker; not quoted in Section 5.3).

Multiple Towers (MT). There are T= 25 towers with common *conv1* layers (with shared weights), each of which takes one input frame. The activations from the towers are concatenated channel-wise after *pool1*, producing an output activation with 1200 channels. The subsequent 1×1 convolution is performed to reduce this dimension, to keep the number of parameters at *conv2* at a managable level. The rest of the

network is the same as the regular VGG-M.

Long Short-Term Memory (LSTM). Here, each convolutional tower shares the configurations of SyncNet's (Chapter 4) visual stream, which is related to the **EF** architecture. The two-layer LSTM ingests the visual features (*fc6* activations) of the 5-frame sliding window, moving 1-frame at a time, and returns the classification result at the end of the sequence.

Discussion. The early fusion architecture **EF** shares similarities with previous work on human action recognition using CNNs (Ji et al., 2013, Karpathy et al., 2014, Ng et al., 2015) in the way that they assume registration between frames. The models perform time-domain operations beginning from the first layer to precisely capture local motion direction and speed (Karpathy et al., 2014). For these methods to capture useful information, good registration of details between frames is critical. However, we are not imposing strict registration, and in any case it goes slightly against the signal (lip motion and mouth region deformation) that we are trying to capture.

In contrast, the **MT** model delays all time-domain registrations (and operations) until after the first set of convolutional and pooling layers. This gives tolerance against minor registration errors (the receptive field size at *conv2* is 11 pixels). Note, the common *conv1* layers of the multiple towers ensures that the same filter weights are used for all frames, whereas in the early fusion architecture **EF** it is possible to learn different weights for each frame. The experimental results show that the registration-tolerant model gives a modest improvement over their counterparts, and the performance improvement is likely to be more significant where the tracking quality is less ideal.

Likewise, the **LSTM** delays all time-domain operations until after all of the convolutional layers, except within the 5 neighbouring frames, between which the movement would be negligible. In addition to registration-tolerance, this model benefits from

the ability to accept sequences of variable lengths, unlike the **EF** and **MT** models.

One other design choice is the size of the input images. This was chosen as 111×111 pixels, which is smaller than that typically used in image classification networks. The reason is that the size of the cropped mouth images are rarely larger than 111×111 pixels, and this smaller choice means that smaller filters can be used at *conv1* (than those used in VGG-M and ResNet-50) without sacrificing receptive fields, but at a gain in avoiding unnecessary parameters being learnt.

5.1.2 Training

Data augmentation. Data augmentation often helps to improve validation performance by reducing overfitting in CNN image classification tasks (Krizhevsky et al., 2012). We apply the augmentation techniques used on the ImageNet classification task by (Krizhevsky et al., 2012, Simonyan and Zisserman, 2015) (*e.g.* random cropping, flipping, colour shift), with a consistent transformation applied to all frames of a single clip. To further augment the training data, we make random shifts in time by up to 0.2 seconds, which improves the *top-1* validation error by 3.5% compared to the standard ImageNet augmentation methods. It was not feasible to scale in the time-domain as this results in artifacts being shown due to the relatively low video refresh rate of 25fps.

Details. Our implementation is based on the MATLAB toolbox MatConvNet (Vedaldi et al., 2014) and Caffe (Jia, 2013). The network is trained using SGD with momentum 0.9 and batch normalisation (Ioffe and Szegedy, 2015), but without dropout. The training was stopped after 20 epochs, or when the validation error did not improve for 3 epochs, whichever is sooner. The learning rate of 10^{-2} to 10^{-4} was used, decreasing on log scale.

ABOUT	ABSOLUTELY	ABUSE	ACCESS	ACCORDING	ACCUSED	ACROSS
ACTION	ACTUALLY	AFFAIRS	AFFECTED	AFRICA	AFTER	AFTERNOON
AGAIN	AGAINST	AGREE	AGREEMENT	AHEAD	ALLEGATIONS	ALLOW
ALLOWED	ALMOST	ALREADY	ALWAYS	AMERICA	AMERICAN	AMONG
AMOUNT	ANNOUNCED	ANOTHER	ANSWER	ANYTHING	AREAS	AROUND
ARRESTED	ASKED	ASKING	ATTACK	ATTACKS	AUTHORITIES	BANKS
BECAUSE	BECOME	BEFORE	BEHIND	BEING	BELIEVE	BENEFIT
BENEFITS	BETTER	BETWEEN	BIGGEST	BILLION	BLACK	BORDER
BRING	BRITAIN	BRITISH	BROUGHT	BUDGET	BUILD	BUILDING
BUSINESS	BUSINESSES	CALLED	CAMERON	CAMPAIGN	CANCER	CANNOT
CAPITAL	CASES	CENTRAL	CERTAINLY	CHALLENGE	CHANCE	CHANGE
CHANGES	CHARGE	CHARGES	CHIEF	CHILD	CHILDREN	CHINA
CLAIMS	CLEAR	CLOSE	CLOUD	COMES	COMING	COMMUNITY
COMPANIES	COMPANY	CONCERNS	CONFERENCE	CONFLICT	CONSERVATIVE	CONTINUE
CONTROL	COULD	COUNCIL	COUNTRIES	COUNTRY	COUPLE	COURSE
COURT	CRIME	CRISIS	CURRENT	CUSTOMERS	DAVID	DEATH
DEBATE	DECIDED	DECISION	DEFICIT	DEGREES	DESCRIBED	DESPITE
DETAILS	DIFFERENCE	DIFFERENT	DIFFICULT	DOING	DURING	EARLY
EASTERN	ECONOMIC	ECONOMY	EDITOR	EDUCATION	ELECTION	EMERGENCY
ENERGY	ENGLAND	ENOUGH	EUROPE	EUROPEAN	EVENING	EVENTS
EVERY	EVERYBODY	EVERYONE	EVERYTHING	EVIDENCE	EXACTLY	EXAMPLE
EXPECT	EXPECTED	EXTRA	FACING	FAMILIES	FAMILY	FIGHT
FIGHTING	FIGURES	FINAL	FINANCIAL	FIRST	FOCUS	FOLLOWING
FOOTBALL	FORCE	FORCES	FOREIGN	FORMER	FORWARD	FOUND
FRANCE	FRENCH	FRIDAY	FRONT	FURTHER	FUTURE	GAMES
GENERAL	GEORGE	GERMANY	GETTING	GIVEN	GIVING	GLOBAL
GOING	GOVERNMENT	GREAT	GREECE	GROUND	GROUP	GROWING
GROWTH	GUILTY	HAPPEN	HAPPENED	HAPPENING	HAVING	HEALTH
HEARD	HEART	HEAVY	HIGHER	HISTORY	HOMES	HOSPITAL
HOURS	HOUSE	HOUSING	HUMAN	HUNDREDS	IMMIGRATION	IMPACT
IMPORTANT	INCREASE	INDEPENDENT	INDUSTRY	INFLATION	INFORMATION	INQUIRY
INSIDE	INTEREST	INVESTMENT	INVOLVED	IRELAND	ISLAMIC	ISSUE
ISSUES	ITSELF	JAMES	JUDGE	JUSTICE	KILLED	KNOWN
LABOUR	LARGE	LATER	LATEST	LEADER	LEADERS	LEADERSHIP
LEAST	LEAVE	LEGAL	LEVEL	LEVELS	LIKELY	LITTLE
LIVES	LIVING	LOCAL	LONDON	LONGER	LOOKING	MAJOR
MAJORITY	MAKES	MAKING	MANCHESTER	MARKET	MASSIVE	MATTER
MAYBE	MEANS	MEASURES	MEDIA	MEDICAL	MEETING	MEMBER
MEMBERS	MESSAGE	MIDDLE	MIGHT	MIGRANTS	MILITARY	MILLION
MILLIONS	MINISTER	MINISTERS	MINUTES	MISSING	MOMENT	MONEY
MONTH	MONTHS	MORNING	MOVING	MURDER	NATIONAL	NEEDS
NEVER	NIGHT	NORTH	NORTHERN	NOTHING	NUMBER	NUMBERS
OBAMA	OFFICE	OFFICERS	OFFICIALS	OFTEN	OPERATION	OPPOSITION
ORDER	OTHER	OTHERS	OUTSIDE	PARENTS	PARLIAMENT	PARTIES
PARTS	PARTY	PATIENTS	PAYING	PEOPLE	PERHAPS	PERIOD
PERSON	PERSONAL	PHONE	PLACE	PLACES	PLANS	POINT
POLICE	POLICY	POLITICAL	POLITICIANS	POLITICS	POSITION	POSSIBLE
POTENTIAL	POWER	POWERS	PRESIDENT	PRESS	PRESSURE	PRETTY
PRICE	PRICES	PRIME	PRISON	PRIVATE	PROBABLY	PROBLEM
PROBLEMS	PROCESS	PROTECT	PROVIDE	PUBLIC	QUESTION	QUESTIONS
QUITE	RATES	RATHER	REALLY	REASON	RECENT	RECORD
REFERENDUM	REMEMBER	REPORT	REPORTS	RESPONSE	RESULT	RETURN
RIGHT	RIGHTS	RULES	RUNNING	RUSSIA	RUSSIAN	SAYING
SCHOOL	SCHOOLS	SCOTLAND	SCOTTISH	SECOND	SECRETARY	SECTOR
SECURITY	SEEMS	SENIOR	SENSE	SERIES	SERIOUS	SERVICE
SERVICES	SEVEN	SEVERAL	SHORT	SHOULD	SIDES	SIGNIFICANT
SIMPLY	SINCE	SINGLE	SITUATION	SMALL	SOCIAL	SOCIETY
SOMEONE	SOMETHING	SOUTH	SOUTHERN	SPEAKING	SPECIAL	SPEECH
SPEND	SPENDING	SPENT	STAFF	STAGE	STAND	START
STARTED	STATE	STATEMENT	STATES	STILL	STORY	STREET
STRONG	SUNDAY	SUNSHINE	SUPPORT	SYRIA	SYRIAN	SYSTEM
TAKEN	TAKING	TALKING	TALKS	TEMPERATURES	TERMS	THEIR
THEMSELVES	THERE	THESE	THING	THINGS	THINK	THIRD
THOSE	THOUGHT	THOUSANDS	THREAT	THREE	THROUGH	TIMES
TODAY	TOGETHER	TOMORROW	TONIGHT	TOWARDS	TRADE	TRIAL
TRUST	TRYING	UNDER	UNDERSTAND	UNION	UNITED	UNTIL
USING	VICTIMS	VIOLENCE	VOTERS	WAITING	WALES	WANTED
WANTS	WARNING	WATCHING	WATER	WEAPONS	WEATHER	WEEKEND
WEEKS	WELCOME	WELFARE	WESTERN	WESTMINSTER	WHERE	WHETHER
WHICH	WHILE	WHOLE	WINDS	WITHIN	WITHOUT	WOMEN
WORDS	WORKERS	WORKING	WORLD	WORST	WOULD	WRONG
YEARS	YESTERDAY	YOUNG				

Table 5.1: List of words in Lip Reading in the Wild (LRW) dataset.

5.2 Dataset

The 500-word **LRW** dataset is generated from the audio-visual corpus described in Chapter 3.

The training, validation and test sets are disjoint in time. The dates of videos corresponding to each set is shown in Table 5.2. Note that we leave a week's gap between the test set and the rest in case any news footage is repeated. The lexicon is obtained by selecting the 500 most frequently occurring words between 5 and 10 characters in length (Figure 5.2 gives the word duration statistics). The full list of words is given in

Table 5.1. This word length is chosen such that the speech duration does not exceed the fixed one-second bracket that is used in the recognition architecture (Figure 5.2), whilst shorter words are not included because there are too many ambiguities due to homophemes (*e.g.* 'bad', 'bat', 'pat', 'mat', etc. are all visually identical), and sentence-level context would be needed to disambiguate these.

Set	**Dates**	**# class**	**#/class**
Train	01/01/2010 - 28/02/2015	500	800+
Val	01/03/2015 - 25/07/2015	500	50
Test	01/08/2015 - 31/03/2016	500	50

Table 5.2: Dataset statistics.

These 500 words occur at least 800 times in the training set, and at least 40 times in each of the validation and test sets. For each of the occurrences, the one-second clip is taken, and the face is cropped with the mouth centred. The words are *not* isolated, as is the case in other lip-reading datasets; as a result, there may be co-articulation of the lips from preceding and subsequent words. The *test* set is manually checked for errors.

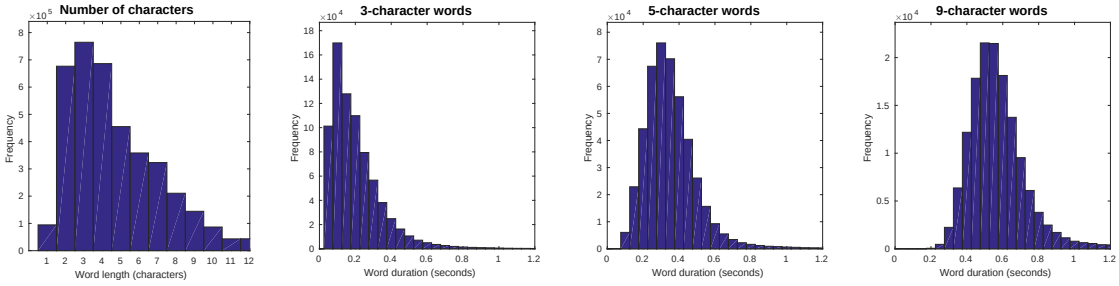

Figure 5.2: Word statistics. Regardless of the actual duration of the word, we take a 1-second clip for training and test.

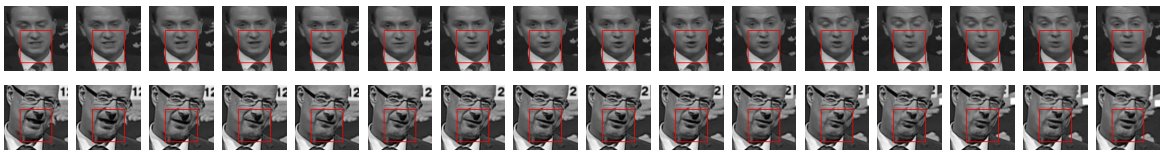

Figure 5.3: The mouth motions for 'afternoon' from two different speakers. The network sees the areas inside the red squares.

5.3 Experiments

In this section we evaluate and compare the several proposed architectures, and discuss the challenges arising from the visual ambiguities between words. We then compare to the state of the art on a public benchmark.

5.3.1 Comparison of architectures

Evaluation protocol. The models are evaluated on the independent test set (Section 5.2). We report *top-1* and *top-10* accuracies , as well as recall against rank curves. Here, the *'Recall@K'* is the proportion of times that the correct class is found in the top-K predictions for the word. We also report the character-level edit distance (Kondrak, 2000), which is the minimum number of character-level operations required to convert the predicted string to the ground truth. This edit distance (metric) is smaller where the predicted string is similar to the ground truth (*e.g.* 'concerned' and 'concerns' have an edit distance of 2) and larger penalties where the words are very different (*e.g.* 'concerned' and 'company' have an edit distance of 6).

Results. The results are shown in Table 5.3 and Figure 5.4. The *top-10* accuracy for the best model is over 95%, despite the relatively modest *top-1* figure of around 70%. This is a result of ambiguities in lip reading, which we will discuss next.

Net	**Top-1**	**Top-10**	**ED**
EF	57.0%	88.8%	2.32
MT	61.1%	90.4%	2.06
LSTM	66.0%	94.3%	1.73

Table 5.3: Word classification accuracy on the LRW dataset for the different architectures. **ED**: Edit Distance.

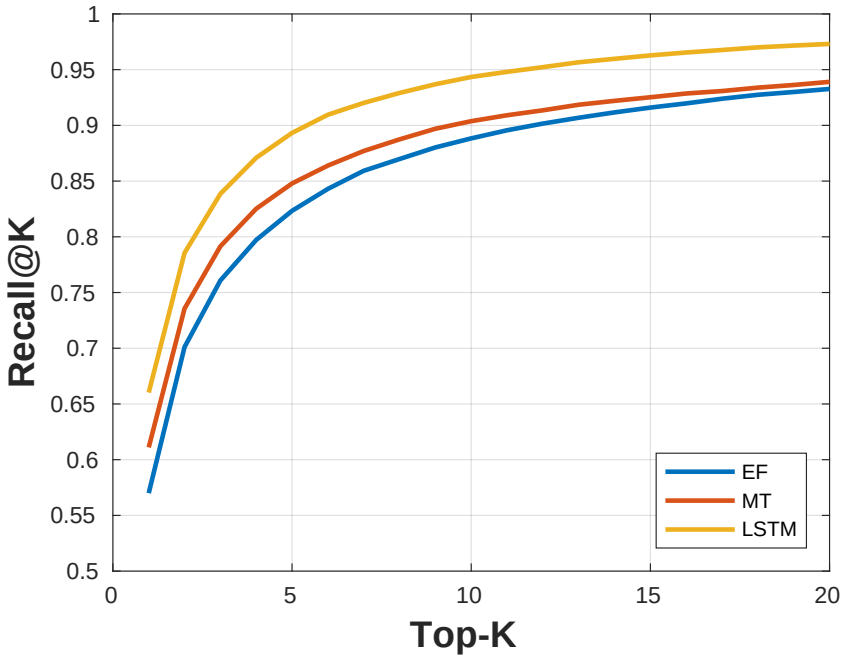

Figure 5.4: Recall vs Rank curves for the word classification.

0.32	BENEFITS	BENEFIT
0.31	QUESTIONS	QUESTION
0.31	REPORT	REPORTS
0.31	BORDER	IMPORTANT
0.31	AMERICA	AMERICAN
0.29	GROUND	AROUND
0.28	RUSSIAN	RUSSIA
0.28	FIGHT	FIGHTING
0.26	FAMILY	FAMILIES
0.26	AMERICAN	AMERICA
0.26	BENEFIT	BENEFITS
0.25	ELECTIONS	ELECTION
0.24	WANTS	WANTED

0.24	HAPPEN	HAPPENED
0.24	FORCE	FORCES
0.23	HAPPENED	HAPPEN
0.23	SERIOUS	SERIES
0.23	TROOPS	GROUPS
0.22	QUESTION	QUESTIONS
0.21	PROBLEM	PROBABLY
0.21	WANTED	WANTS
0.21	RUSSIA	RUSSIAN
0.20	TAKEN	TAKING
0.20	PROBLEM	PROBLEMS
0.20	MISSING	MEETING
0.20	PARTIES	PARTY

Table 5.4: Most frequently confused word pairs. The numbers refer to class confusions.

5.3.2 Analysis of confusions

Here, we examine the classification results, in particular, the scenarios in which the network fails to correctly classify the spoken word. Table 5.4 shows the most common confusions between words in the test set. This is generated by taking the largest off-diagonal values in the word confusion matrix. This result confirms our prior knowledge about the challenges in visual speech recognition – almost all of the top confusions are either (i) a plural of the original word (*e.g.* 'report' and 'reports') which is ambiguous because one word is a subset of the other, and the words are not isolated in our dataset

so this can be due to co-articulation; or (ii) a known homopheme visual ambiguity where the words cannot be distinguished using visual information alone (*e.g.* 'billion' and 'million', 'worse' and 'worst'). Such errors are phonetically understandable. For example, some of the most common confusions, e.g. 'groups' which is phonetically (G R UW P S) and 'troops' (T R UW P S), 'ground' (G R AW N D) and 'around' (ER AW N D), actually share most of the phonemes.

Apart from these difficulties, the failure cases are typically for extreme samples. For example, due to strong international accents, or poor quality/low bandwidth location reports and Skype interviews, where there are motion compression artifacts or frames dropped from the transmission.

5.3.3 Comparison to state of the art

It is worth noting that the *top-1* classification accuracy of 66%, shown in Table 5.3, is comparable to that of many of the recent works (Fu et al., 2008, Ngiam et al., 2011, Petridis and Pantic, 2016) performed on lexicon sizes that are orders of magnitude smaller.

Figure 5.5: Original video frames for 'hello' on OuluVS. Compare this to the our original input frames in Figure 3.4.

Method	**Top-1**
Zhou et al. (2014a)	73.5%
Saitoh et al. (2016)	85.6%
MT	93.2%
LSTM	94.1%

Table 5.5: Word classification accuracy on OuluVS2 (short phrases, frontal view). Note that **MT** and **LSTM** models have been pre-trained on the LRW dataset.

OuluVS2. We evaluate our method on the frontal-view subset of OuluVS2 dataset (An-

ina et al., 2015b). The dataset consists of 52 subjects uttering 10 phrases (*e.g.* 'thank you', 'hello', etc.), and has been widely used in previous works. Here, we assess on a speaker-independent experiment, where some of the subjects are reserved for testing.

To apply our method on this dataset, we pre-train the convolutional layers on the LRW dataset, and re-train the fully-connected and/or LSTM layers from scratch. Training from scratch on OuluVS2 underperforms as the size of this dataset is insufficient to train a deep network. For EF and MT models, we simply repeat the first and the last frames to fill the 1-second clip if the phrase is shorter than 25 frames. If the clip is longer, we take a random crop.

As can be seen in Table 5.5 the method achieves a strong performance, and sets the new state-of-the-art. Note that, without retraining the convolutional part of the network, we achieve these strong results on videos that are very different to ours in terms of lighting, background, camera perspective, etc. (Figure 5.5), which shows that the model generalises well across different formats.

5.4 Conclusion

We have shown that CNN and LSTM architectures can be used to classify temporal lip motion sequences of words with excellent results. We also demonstrated a recognition performance that exceeds the state of the art on a standard public benchmark dataset, OuluVS2. The next step is to extend this work to lip reading full sentences.

Chapter 6

Deep learning for sentence-level lip reading and multi-modal speech recognition

Lip reading is inherently ambiguous at the word level due to homophones – different characters that produce exactly the same lip sequence (*e.g.* 'p' and 'b'). However, such ambiguities can be resolved to an extent using the context of neighbouring words in a sentence.

In this chapter we go beyond the word-level recognition work of Chapter 5 and explore deep learning methods that can lip read natural sentences from 'in the wild' videos.

A machine that can lip read opens up a host of applications: 'dictating' instructions or messages to a phone in a noisy environment; transcribing and re-dubbing archival silent films; resolving multi-talker simultaneous speech; and, improving the performance of automated speech recogition in general.

In this case the model is based on the recent sequence-to-sequence (encoder-decoder with attention) translation architectures that have been developed for speech recognition and machine translation (Section 2.1.3).

We also investigate how lip reading can contribute to *audio* based speech recognition.

There is a large literature on this contribution, particularly in noisy environments, as well as the converse where some derived measure of audio can contribute to lip reading for the deaf or hard of hearing. To investigate this aspect we train a model to recognize characters from both audio and visual input, and then systematically disturb the audio channel or remove the visual channel.

Our model (Section 6.1) outputs at the character level, is able to learn a language model, and has a novel dual attention mechanism that can operate over visual input only, audio input only, or both. We show (Section 6.2) that training can be accelerated by a form of curriculum learning. We also describe (Section 6.3) the generation and statistics of a new large scale Lip Reading Sentences (LRS) dataset, based on the audio-visual corpus of Chapter 3 containing talking faces together with subtitles of what is said. The broadcasts contain faces 'in the wild' with a significant variety of pose, expressions, lighting, backgrounds, and ethnic origin.

The performance of the model is assessed on a test set of the LRS dataset, as well as on public benchmarks datasets for lip reading including LRW (Chapter 5) and GRID (Cooke et al., 2006). We demonstrate *open world* (unconstrained sentences) lip reading on the LRS dataset, and in all cases on public benchmarks the performance exceeds that of prior work.

6.1 The Watch, Listen, Attend and Spell model

In this section, we describe the *Watch, Listen, Attend and Spell* network that learns to predict characters in sentences being spoken from a video of a talking face, with or without audio.

We model each character y_i in the output character sequence $\mathbf{y} = (y_1, y_2, ..., y_l)$ as a conditional distribution of the previous characters $y_{<i}$, the input image sequence $\mathbf{x}^v = (x_1^v, x_2^v, ..., x_n^v)$ for lip reading, and the input audio sequence $\mathbf{x}^a = (x_1^a, x_2^a, ..., x_m^a)$.

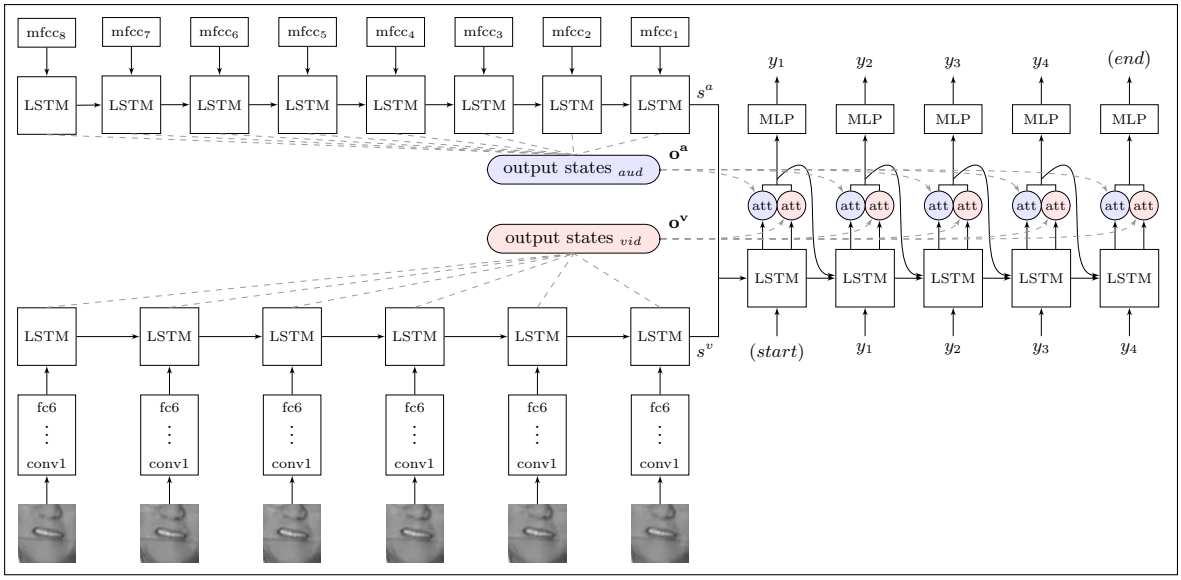

Figure 6.1: *Watch, Listen, Attend and Spell* architecture. At each time step, the decoder outputs a character y_i, as well as two attention vectors. The attention vectors are used to select the appropriate period of the input visual and audio sequences.

Hence, we model the output probability distribution as:

$$P(\mathbf{y}|\mathbf{x}^v, \mathbf{x}^a) = \prod_i P(y_i|\mathbf{x}^v, \mathbf{x}^a, y_{<i}) \tag{6.1}$$

Our model, which is summarised in Figure 6.1, consists of three key components: the image encoder `Watch` (Section 6.1.1), the audio encoder `Listen` (Section 6.1.2), and the character decoder `Spell` (Section 6.1.3). Each encoder transforms the respective input sequence into a fixed-dimensional state vector s, a hidden vector h, and sequences of encoder outputs $\mathbf{o} = (o_1, ..., o_p)$, $p \in (n, m)$; the decoder ingests the state and the attention vectors from both encoders and produces a probability distribution over the output character sequence.

$$s^v, \mathbf{o}^v = \texttt{Watch}(\mathbf{x}^v) \tag{6.2}$$

$$s^a, \mathbf{o}^a = \texttt{Listen}(\mathbf{x}^a) \tag{6.3}$$

$$P(\mathbf{y}|\mathbf{x}^v, \mathbf{x}^a) = \texttt{Spell}(s^v, s^a, \mathbf{o}^v, \mathbf{o}^a) \tag{6.4}$$

The three modules in the model are trained jointly. We describe the modules next, with implementation details given in Section 6.2.5.

6.1.1 Watch: Image encoder

The image encoder consists of the convolutional module that generates image features f_i^v for every input timestep x_i^v, and the recurrent module that produces the fixed-dimensional state vector s^v, the hidden vector h^v and a set of output vectors $\mathbf{o}^v$.

$$f_i^v = \texttt{CNN}(x_i^v) \tag{6.5}$$

$$h_i^v, o_i^v = \texttt{LSTM}(f_i^v, h_{i+1}^v) \tag{6.6}$$

$$s^v = h_1^v \tag{6.7}$$

The convolutional network is based on the VGG-M model (Chatfield et al., 2014), as it is memory-efficient, fast to train and has a decent classification performance on ImageNet (Russakovsky et al., 2015). The ConvNet layer configuration is shown in Table 6.1 and Figure 6.2. The convolutional layers are abbreviated as *conv1* $\cdots$ *fc6* in the main network diagram.

The encoder LSTM network consumes the output features f_i^v produced by the ConvNet at every input timestep, and generates a fixed-dimensional state vector s^v. In addition, it produces an output vector o_i^v at every timestep i. Note that the network

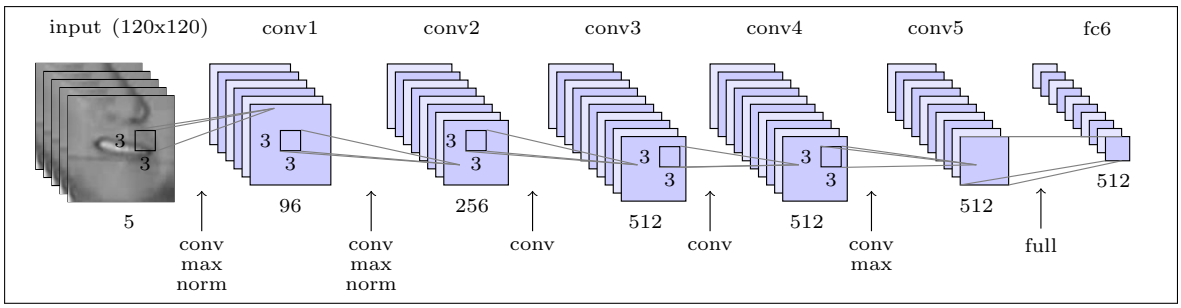

Figure 6.2: The ConvNet architecture. The input is five gray level frames centered on the mouth region. The 512-dimensional fc6 vector forms the input to the LSTM.

Layer	Support	Filt dim.	# filts.	Stride	Data size
conv1	3×3	5	96	1×1	109×109
pool1	3×3	-	-	2×2	54×54
conv2	3×3	96	256	2×2	27×27
pool2	3×3	-	-	2×2	13×13
conv3	3×3	256	512	1×1	13×13
conv4	3×3	512	512	1×1	13×13
conv5	3×3	512	512	1×1	13×13
pool5	3×3	-	-	2×2	6×6
fc6	6×6	512	512	-	1×1

Table 6.1: The ConvNet architecture.

ingests the inputs in reverse time order (as in Equation 6.6), which has been shown to improve results in Sutskever et al. (2014).

6.1.2 Listen: Audio encoder

The `Listen` module is an LSTM encoder similar to the `Watch` module, without the convolutional part. The LSTM directly ingests 13-dimensional MFCC features in reverse time order, and produces the state vector s^a, the hidden vector h^a and the output vectors $\mathbf{o}^a$.

$$h_j^a, o_j^a = \texttt{LSTM}(x_j^a, h_{j+1}^a) \tag{6.8}$$

$$s^a = h_1^a \tag{6.9}$$

6.1.3 Spell: Character decoder

The `Spell` module is based on a LSTM transducer (Bahdanau et al., 2015, Chan et al., 2015, Chorowski et al., 2015), but here we add a dual attention mechanism. At every output step k, the decoder LSTM produces the decoder states h_k^d and output vectors o_k^d from the previous step context vectors c_{k-1}^v and c_{k-1}^a, output y_{k-1} and decoder state h_{k-1}^d. The attention vectors are generated from the attention mechanisms $\texttt{Attention}^{\texttt{v}}$ and $\texttt{Attention}^{\texttt{a}}$.

The implementation of the mechanism is based on the work of Bahdanau *et al.* (Bahdanau et al., 2015). The attention vector α^v for the video stream, also often called the alignment, is computed as follows:

$$\alpha_{k,i}^v = \texttt{Attention}^{\texttt{v}}(h_k^d, \mathbf{o}^v) \tag{6.10}$$

$$= \frac{\texttt{exp}(e_{k,i})}{\sum_{i=1}^{n} \texttt{exp}(e_{k,i})} \tag{6.11}$$

$$e_{k,i} = w^T \texttt{tanh}(W h_k^d + V o_i^v + b) \tag{6.12}$$

where w, b, W and V are weights to be learnt, i is the index over the input video sequence, k is the index for the output time step.

We use two independent attention mechanisms for the lip and the audio input streams to refer to the asynchronous inputs with different sampling rates. The attention vectors are fused with the output states (Equations 6.14 and 6.15) to produce the context vectors c_k^v and c_k^a that encapsulate the information required to produce the next step output. The model generates the output from the list given in Table 6.2, and the probability distribution is generated by an MLP with softmax over the output.

$$h_k^d, o_k^d = \texttt{LSTM}(h_{k-1}^d, y_{k-1}, c_{k-1}^v, c_{k-1}^a) \tag{6.13}$$

$$c_k^v = \mathbf{o}^v \cdot \texttt{Attention}^{\texttt{v}}(h_k^d, \mathbf{o}^v) \tag{6.14}$$

$$c_k^a = \mathbf{o}^a \cdot \texttt{Attention}^{\texttt{a}}(h_k^d, \mathbf{o}^a) \tag{6.15}$$

$$P(y_i|\mathbf{x}^v, \mathbf{x}^a, y_{<i}) = \texttt{softmax}(\texttt{MLP}(o_k^d, c_k^v, c_k^a)) \tag{6.16}$$

At $k = 1$, the final encoder states s_l and s_a are used as the input instead of the previous decoder state – *i.e.* $h_0^d = \texttt{concat}(s^a, s^v)$ – to help produce the context vectors c_1^v and c_1^a in the absence of the previous state or context.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X	Y	Z	0	1	2	3	4	5
6	7	8	9	,	.	!	?	:	’	[sos]					
[eos]	[pad]														

Table 6.2: The output characters

Discussion. In our experiments, we have observed that the attention mechanism is absolutely critical for the audio-visual speech recognition system to work. Without attention, the model appears to ‘forget’ the input signal, and produces an output sequence that correlates very little to the input, beyond the first word or two (which the model gets correct, as these are the last words to be seen by the encoder). The attention-less model yields Word Error Rates over 100%, so we do not report these results.

The dual-attention mechanism allows the model to extract information from both audio and video inputs, even when one stream is absent, or the two streams are not time-aligned. The benefits are clear in the experiments with noisy or no audio (Section 6.4).

Bidirectional LSTMs have been used in many sequence learning tasks (Chan et al., 2015, Chorowski et al., 2015, Graves et al., 2013) for their ability to produce outputs

conditioned on future context as well as past context. We have tried replacing the unidirectional encoders in the `Watch` and `Listen` modules with bidirectional encoders, however these networks took significantly longer to train, whilst providing no obvious performance improvement. This is presumably because the `Decoder` module is anyway conditioned on the full input sequence, so bidirectional encoders are not necessary for providing context, and the attention mechanism suffices to provide the additional local focus.

6.2 Training strategy

In this section, we describe the strategy used to effectively train the *Watch, Listen, Attend and Spell* network, making best use of the limited amount of data available.

6.2.1 Curriculum learning

Our baseline strategy is to train the model from scratch, using the full sentences from the 'Lip Reading Sentences' dataset – previous works in speech recognition have taken this approach. However, as (Chan et al., 2015) reports, the LSTM network converges very slowly when the number of timesteps is large, because the decoder initially has a hard time extracting the relevant information from all the input steps.

We introduce a new strategy where we start training only on single word examples, and then let the sequence length grow as the network trains. These short sequences are parts of the longer sentences in the dataset. We observe that the rate of convergence on the training set is several times faster, and it also significantly reduces overfitting, presumably because it works as a natural way of augmenting the data. The test performance improves by a large margin, reported in Section 6.4.

6.2.2 Scheduled sampling

When training a recurrent neural network, one typically uses the previous time step ground truth as the next time step input, which helps the model learn a kind of language model over target tokens. However during inference, the previous step ground-truth is unavailable, resulting in poorer performance because the model was not trained to be tolerant to feeding in bad predictions at some time steps. We use the scheduled sampling method of Bengio et al. (2015) to bridge this discrepancy between how the model is used at training and inference. At train time, we randomly sample from the previous output, instead of always using the ground-truth. When training on shorter sub-sequences, ground-truth previous characters are used. When training on full sentences, the sampling probability from the previous output was increased in steps from 0 to 0.25 over time. We were not able to achieve stable learning at sampling probabilities of greater than 0.25.

6.2.3 Multi-modal training

Networks with multi-modal inputs can often be dominated by one of the modes (Feichtenhofer et al., 2016). In our case we observe that the audio signal dominates, because speech recognition is a significantly easier problem than lip reading. To help prevent this from happening, one of the following input types is selected with uniform probability at train time for each example: (1) audio only; (2) lips only; (3) audio and lips.

If mode (1) is selected, the audio-only data described in Section 6.3.1 is used. Otherwise, the standard audio-visual data is used.

We have over 300,000 sentences in the recorded data, but only around 100,000 have corresponding facetracks. In machine translation, it has been shown that monolingual dummy data can be used to help improve the performance of a translation model (Sen-

nrich et al., 2015). By similar rationale, we use the sentences without facetracks as supplementary training data to boost audio recognition performance and to build a richer language model to help improve generalisation.

6.2.4 Training with noisy audio

The WLAS model is initially trained with clean input audio for faster convergence. To improve the model's tolerance to audio noise, we apply additive white Gaussian noise with SNR of 10dB (10:1 ratio of the signal power to the noise power) and 0dB (1:1 ratio) later in training.

6.2.5 Implementation details

The input images are 120×120 in dimension, and are sampled at 25Hz. The image only covers the lip region of the face, as shown in Figure 6.6. The ConvNet ingests 5-frame sliding windows using the Early Fusion method of Chapter 5, moving 1-frame at a time. The MFCC features are calculated over 25ms windows and at 100Hz, with a time-stride of 1. For `Watch` and `Listen` modules, we use a three layer LSTM with cell size of 256. For the `Spell` module, we use a three layer LSTM with cell size of 512. The output size of the network is 45, for every character in the alphabet, numbers, common punctuations, and tokens for `[sos]`, `[eos]`, `[pad]`.

Our implementation is based on the TensorFlow library (Abadi et al., 2016). The network is trained using stochastic gradient descent with a batch size of 64 and with dropout and label smoothing. The layer weights of the convolutional layers are initialised from the visual stream of Chapter 4. All other weights are randomly initialised.

An initial learning rate of 0.1 was used, and decreased by 10% every time the training error did not improve for 2,000 iterations. Training on the full sentence data was

stopped when the validation error did not improve for 5,000 iterations.

6.3 Dataset

The dataset is built from the audio-visual corpus described in Chapter 3.

Sentence extraction. The videos are divided into invididual sentences/ phrases using the punctuations in the transcript. The sentences are separated by full stops, commas and question marks; and are clipped to 100 characters or 10 seconds, due to GPU memory constraints. We do not impose any restrictions on the vocabulary size.

Two overlapping datasets are generated from the corpus, representing different test conditions: the **LRS** dataset containing only the front-facing speakers; and the **MV-LRS** dataset containing videos taken from a balance of different viewpoints, from frontal to profile.

The training, validation and test sets are divided according to broadcast date, and the statistics of spoken sentences corresponding to each set are shown in Table 6.3 for the frontal-only data and Table 6.4 for the multi-view data. The dataset contains thousands of different speakers which enables the model to be speaker agnostic.

Set	**Dates**	**# Utter.**	**Vocab**
Train	01/2010 - 12/2015	101,195	16,501
Val	01/2016 - 02/2016	5,138	4,572
Test	03/2016 - 09/2016	11,783	6,882
All		118,116	17,428

Table 6.3: **The Lip Reading Sentences (LRS) audio-visual dataset.** Division of training, validation and test data; and the number of utterances and vocabulary size of each partition.

Set	Dates	# Utter.	Vocab
Train	01/2010 - 12/2015	67,793	14,440
Val	01/2016 - 02/2016	2,352	4,330
Test	03/2016 - 09/2016	4,429	4,375
All		74,574	14,960

Table 6.4: **The Multi-View Lip Reading Sentences (MV-LRS) dataset.** Division of training, validation and test data; and the number of utterances and vocabulary size of each partition.

6.3.1 Audio-only data

In addition to the audio-visual dataset, we prepare an auxiliary audio-only *training* dataset. These are the sentences in the BBC programs for which facetracks are not available. The use of this data is described in Section 6.2.3. It is only used for training, not for testing.

Set	Dates	# Utter.	Vocab
Train	01/2010 - 12/2015	342,644	25,684

Table 6.5: Statistics of the Audio-only training set.

6.4 Experiments on frontal view datasets

In this section we evaluate and compare the proposed architecture and training strategies. We also compare our method to the state of the art on public benchmark datasets.

To clarify which of the modalities are being used, we call the models in lips-only and audio-only experiments *Watch, Attend and Spell* (WAS), *Listen, Attend and Spell* (LAS) respectively. These are the same *Watch, Listen, Attend and Spell* model with either of the inputs disconnected and replaced with all-zeros.

6.4.1 LRS dataset.

The models are trained on the frontal-only LRS dataset (the train/val partition) and the Audio-only training dataset (Section 6.3). The inference and evaluation procedures are described below.

Beam search. Decoding is performed with beam search of width 4, in a similar manner to (Chan et al., 2015, Sutskever et al., 2014). At each timestep, the hypotheses in the beam are expanded with every possible character, and only the 4 most probable hypotheses are stored. Figure 6.3 shows the effect of increasing the beam width – there is no observed benefit for increasing the width beyond 4.

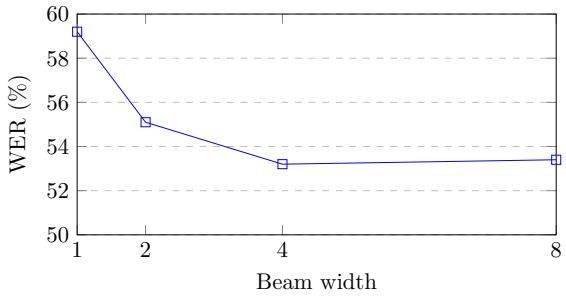

Figure 6.3: The effect of beam width on Word Error Rate.

Evaluation protocol. The models are evaluated on an independent test set (Section 6.3). For all experiments, we report the Character Error Rate (CER), the Word Error Rate (WER) and the BLEU metric. CER and WER are defined as $\texttt{ErrorRate} = (S + D + I)/N$, where S is the number of substitutions, D is the number of deletions, I is the number of insertions to get from the reference to the hypothesis, and N is the number of words in the reference. BLEU (Papineni et al., 2002) is a modified form of n-gram precision to compare a candidate sentence to one or more reference sentences. Here, we use the unigram BLEU.

Results. All of the training methods discussed in Section 6.2 contribute to improving the performance. A breakdown of this is given in Table 6.6 for the lips-only experiment. For all other experiments, we only report results obtained using the best

Method	SNR	CER	WER	BLEU†
Lips only				
Professional‡	-	58.7%	73.8%	23.8
WAS	-	59.9%	76.5%	35.6
WAS+CL	-	47.1%	61.1%	46.9
WAS+CL+SS	-	42.4%	58.1%	50.0
WAS+CL+SS+BS	-	39.5%	50.2%	54.9
Audio only				
Google Speech API	clean	17.6%	22.6%	78.4
Kaldi SGMM+MMI⋆	clean	9.7%	16.8%	83.6
LAS+CL+SS+BS	clean	10.4%	17.7%	84.0
LAS+CL+SS+BS	10dB	26.2%	37.6%	66.4
LAS+CL+SS+BS	0dB	50.3%	62.9%	44.6
Audio and lips				
WLAS+CL+SS+BS	clean	7.9%	13.9%	87.4
WLAS+CL+SS+BS	10dB	17.6%	27.6%	75.3
WLAS+CL+SS+BS	0dB	29.8%	42.0%	63.1

Table 6.6: Performance on the LRS test set. **WAS**: *Watch, Attend and Spell*; **LAS**: *Listen, Attend and Spell*; **WLAS**: *Watch, Listen, Attend and Spell*; **CL**: Curriculum Learning; **SS**: Scheduled Sampling; **BS**: Beam Search. †Unigram BLEU with brevity penalty. ‡Excluding samples that the lip reader declined to annotate. Including these, the CER rises to 78.9% and the WER to 87.6%. ⋆ The Kaldi SGMM+MMI model used here achieves a WER of 3.6% on the WSJ (eval92) test set, which is within 0.2% of the current state-of-the-art. The acoustic and language models have been re-trained on our dataset.

strategy.

Lips-only examples. The model learns to correctly predict extremely complex unseen sentences from a wide range of content – examples are shown in Table 6.7.

Audio-visual examples. As we hypothesised, the results in Table 6.6 demonstrate that the mouth movements provide important cues in speech recognition when the audio signal is noisy; and also give an improvement in performance even when the audio signal is clean – the character error rate is reduced from 16.2% for audio only to 13.3% for audio together lip reading. Table 6.8 shows some of the many examples where the WLAS model fails to predict the correct sentence from the lips or the audio alone, but successfully deciphers the words when both streams are present.

MANY MORE PEOPLE WHO WERE INVOLVED IN THE ATTACKS
CLOSE TO THE EUROPEAN COMMISSION'S MAIN BUILDING
WEST WALES AND THE SOUTH WEST AS WELL AS WESTERN SCOTLAND
WE KNOW THERE WILL BE HUNDREDS OF JOURNALISTS HERE AS WELL
THAT'S THE LOWEST FIGURE FOR EIGHT YEARS
MANCHESTER FOOTBALL CORRESPONDENT FOR THE DAILY MIRROR
LAYING THE GROUNDS FOR A POSSIBLE SECOND REFERENDUM
ACCORDING TO THE LATEST FIGURES FROM THE OFFICE FOR NATIONAL STATISTICS
IT COMES AFTER A DAMNING REPORT BY THE HEALTH WATCHDOG

Table 6.7: Examples of unseen sentences that WAS correctly predicts (lips only).

GT	IT WILL BE THE CONSUMERS
A	IN WILL BE THE CONSUMERS
L	IT WILL BE IN THE CONSUMERS
AV	IT WILL BE THE CONSUMERS
GT	CHILDREN IN EDINBURGH
A	CHILDREN AND EDINBURGH
L	CHILDREN AND HANDED BROKE
AV	CHILDREN IN EDINBURGH
GT	JUSTICE AND EVERYTHING ELSE
A	JUST GETTING EVERYTHING ELSE
L	CHINESES AND EVERYTHING ELSE
AV	JUSTICE AND EVERYTHING ELSE

Table 6.8: Examples of AVSR results. **GT:** Ground Truth; **A:** Audio only (10dB SNR); **L:** Lips only; **AV:** Audio-visual.

Attention visualisation. The attention mechanism generates explicit alignment between the input video frames (or the audio signal) and the hypothesised character output. Figure 6.4 visualises the alignment of the characters "Good afternoon and welcome to the BBC News at One" and the corresponding video frames.

Decoding speed. The decoding happens significantly faster than real-time. The model takes approximately 0.5 seconds to read and decode a 5-second sentence when using a beam width of 4.

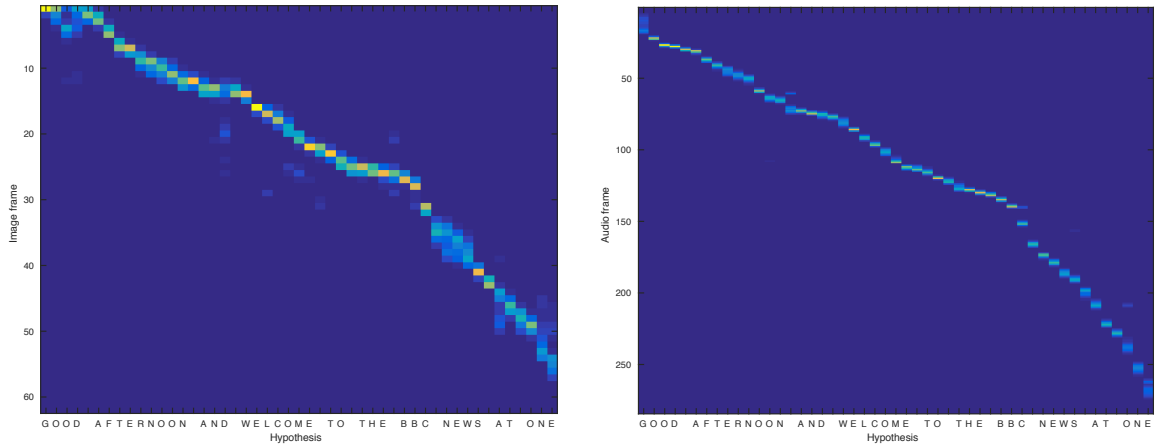

Figure 6.4: Alignment between the video frames and the character output (left); alignment between the audio and the character output (right)

6.4.2 Human experiment

In order to compare the performance of our model to what a human can achieve, we instructed a professional lip reading company to decipher a random sample of 200 videos from our test set. The lip reader has around 10 years of professional experience and deciphered videos in a range of settings, e.g. forensic lip reading for use in court, the royal wedding, etc.

The lip reader was allowed to see the full face (the whole picture in the bottom two rows of Figure 6.6), but not the background, in order to prevent them from reading subtitles or guessing the words from the video content. However, they were informed which program the video comes from, and were allowed to look at some videos from the training set with ground truth.

The lip reader was given 10 times the video duration to predict the words being spoken, and within that time, they were allowed to watch the video as many times as they wished. Each of the test sentences was up to 100 characters in length.

We observed that the professional lip reader is able to correctly decipher less than one-quarter of the spoken words (Table 6.6). This is consistent with previous studies on the accuracy of human lip reading (Marschark and Spencer, 2010). In contrast,

the WAS model (lips only) is able to decipher half of the spoken words. Thus, this is significantly better than professional lip readers can achieve.

6.4.3 LRW dataset

The 'Lip Reading in the Wild' (LRW) dataset (Chapter 5) consists of up to 1000 utterances of 500 isolated words from BBC television, spoken by over a thousand different speakers.

Evaluation protocol. The train, validation and test splits are provided with the dataset. We give word error rates.

Results. The network is fine-tuned for one epoch to classify only the 500 word classes of this dataset's lexicon. As shown in Table 6.9, our result exceeds the current state-of-the-art on this dataset by a large margin.

Methods	LRW	GRID
Lan et al. (2009)	-	35.0%
Wand et al. (2016)	-	20.4%
Assael et al. (2016)	-	4.8%
Chung and Zisserman (2016a)	38.9%	-
WAS (ours)	**23.8%**	**3.0%**

Table 6.9: **Word error rates** on external lip reading datasets.

6.4.4 GRID dataset

Figure 6.5: Still images from the GRID dataset.

The GRID dataset (Cooke et al., 2006) consists of 34 subjects, each uttering 1000

phrases. The utterances are single-syntax multi-word sequences of `verb` (4) + `color` (4) + `preposition` (4) + `alphabet` (25) + `digit` (10) + `adverb` (4) ; *e.g.* 'put blue at A 1 now'. The total vocabulary size is 51, but the number of possibilities at any given point in the output is effectively constrained to the numbers in the brackets above. The videos are recorded in a controlled lab environment, shown in Figure 6.5.

Evaluation protocol. The evaluation follows the standard protocol of (Wand et al., 2016) and (Assael et al., 2016) – the data is randomly divided into train, validation and test sets, where the latter contains 255 utterances for each speaker. We report the word error rates. Some of the previous works report word accuracies, which is defined as ($\texttt{WAcc} = 1 - \texttt{WER}$).

Results. The network is fine-tuned for one epoch on the GRID dataset training set. As can be seen in Table 6.9, our method achieves a strong performance of 3.0% (WER), that substantially exceeds the current state-of-the-art.

6.5 Experiments on multi view datasets

The experiments in Section 6.4 show a leap in the performance of automated lip reading, compared to traditional methods. However, this advance has only been demonstrated for frontal or near frontal faces, and so the question remains: can lips be read in profile to the same standard?

The objective of this section is to answer this question – the Multi-View Watch, Attend and Spell (MV-WAS) model is trained and evaluated on the multi-view dataset described in 6.3. We also test the model on the multi-view OuluVS2 dataset, and show that the performance far surpasses the current state of the art.

Figure 6.6: Example video frames from sentences in the MV-LRS dataset.

6.5.1 MV-LRS dataset

Training. The MV-WAS model is trained using the curriculum learning approach described in Section 6.2.1, where the model starts to learn from easier, single-word examples and gradually move to longer sentences. A single model is trained for all viewpoints (as opposed to separate models for every viewpoint) given that the viewpoint may change within a sentence (as shown in Figure 6.6), and the amount of data for each viewpoint would be insufficient for training in any case.

We compare performance to the frontal-only model. This model is pre-trained on the LRS dataset, and we fine-tune the LSTM layers on the multi-view dataset until the validation error stops improving. This is done so that the language model (implicitly learnt in the decoder) adapts to the new corpus that consists of videos from previously unseen genres (*e.g.* dramas).

Evaluation protocol. The performance measures used are consistent with that used for the frontal view experiment – we report the Character Error Rate (CER), the Word Error Rate (WER) and the unigram BLEU measure.

Decoding. The decoding is performed with a beam size of 4.

Viewpoint	**MV-WAS**			**WAS**		
	CER	WER	BLEU†	CER	WER	BLEU†
Frontal	46.5%	56.4%	49.3	**45.5%**	**56.1%**	**50.4**
Three-quarter	**50.4%**	**59.2%**	**46.1**	55.4%	65.2%	42.5
Profile	**54.4%**	**62.8%**	**42.5**	74.2%	82.6%	26.6

Table 6.10: Results on the MV-LRS dataset. Lower is better for CER and WER; higher is better for BLEU. †Unigram BLEU with brevity penalty.

Results. Performance measures for all viewpoints are given in Table 6.10. The profile performance of the **MV-WAS** model far exceeds the frontal-only WAS model fine-tuned on our dataset, and also shows a significant improvement for three-quarter faces. The performance of our model on frontal videos is comparable to that of the frontal-only WAS model. Table 6.7 gives examples of successfully read sentences.

6.5.2 OuluVS2 dataset

We evaluate the MV-WAS model on the OuluVS2 dataset (Anina et al., 2015a), described in Section 5.3.3. Here, we assess on a speaker-independent experiment, where 12 specified subjects are reserved for testing.

Training. We use the sequence-to-sequence model pre-trained on the MV-LRS dataset, and fine-tune the LSTM layers on the training portion of the OuluVS2 data. Unlike previous works (Lee et al., 2016, Saitoh et al., 2016, Zhou et al., 2014a) that use separate models trained for each viewpoint, we only train a single model to classify the phrases at all angles.

Decoding. The decoding is performed with a beam size of 1.

Results. As can be seen in Table 6.11 our method achieves a strong performance, and sets the new state-of-the-art for the multi-view task.

Method	**Frontal**	**30°**	**45°**	**60°**	**Profile**
Zhou et al. (2014a)	73.0%	75.0%	76.0%	75.0%	70.0%
Lee et al. (2016)	81.1%	80.0%	76.9%	69.2%	82.2%
Saitoh et al. (2016)	85.6%	79.7%	80.8%	83.3%	80.3%
MV-WAS (ours)	**91.1%**	**90.8%**	**90.0%**	**90.0%**	**88.9%**

Table 6.11: **Classification accuracy** on OuluVS2 Short Phrases. Higher is better.

6.6 Conclusion

In this chapter, we have introduced the *'Watch, Listen, Attend and Spell'* network model that can transcribe speech into characters. The model utilises a novel dual attention mechanism that can operate over visual input only, audio input only, or both. Using this architecture, we demonstrate lip reading performance that exceeds a professional lip reader on videos from BBC television. The model also surpasses the performance of all previous work on standard lip reading benchmark datasets, and we also demonstrate that visual information helps to improve speech recognition performance even when the audio is used.

Chapter 7

Generating talking faces from audio

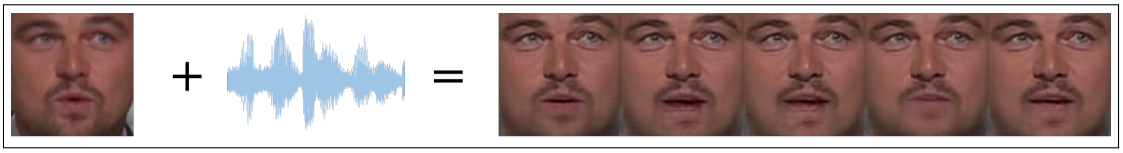

Figure 7.1: The Speech2Vid model generates a video of a talking face, given still images of the person and a speech segment. The model takes an image of the target face and an audio segment, and outputs a video of the target face lip synched with the audio. Note that the target face need not be in the training dataset *i.e.* the Speech2Vid is applicable to unseen images and speech.

In this chapter, we propose a method to generate videos of a talking face using only an audio speech segment and a face image of the target identity (audio and image to video). The speech segment need not be spoken originally by the target person (see Figure 7.1). We dub the approach *Speech2Vid*. Our method differs from previous approaches for this task (see Section 2.2.5) in that instead of learning phoneme to viseme mappings, we learn the correspondences between raw audio and video data directly. By focusing on the speech portion of audio and tight facial regions of speakers in videos, the Speech2Vid model is able to produce natural-looking videos of a talking face at test time even when using an image and audio outside of the training dataset.

The key idea of the approach is to learn a joint embedding of the target face and speech segment that can be used to generate a frame of that face saying (lip synched

with) the speech segment. Thus the inputs are still images of the face (that provides the identity, but is not speaking the target segment) and the target speech segment; and the generated output is the target face speaking the segment. The Speech2Vid model can be learnt from unlabelled videos, as shown in Figure 7.6.

7.1 The Speech2Vid model

Our main goal at test time is to generate a video of a talking face given two inputs: (i) an audio segment, and (ii) still images of the target identity (frontal headshot). The Speech2Vid model (summarised in Figure 7.2 at the block level), consists of four main components: an audio encoder, an identity image encoder, a talking face image decoder, and a deblurring module. For a given input sample, the model generates one frame of image output that best represents the audio sample at a specific time step. The model generates the video on a frame-by-frame basis by sliding a 0.35-second window over the audio sequence.

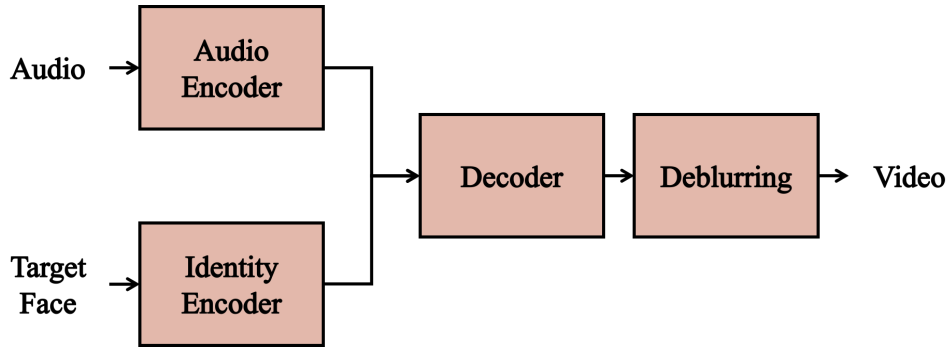

Figure 7.2: The overall Speech2Vid model is a combination of two encoders taking in two different streams of data, audio and image, a decoder that generates images corresponding to the audio, and a CNN deblurring module that refines the output frames.

7.1.1 Input Representations

This section describes the input representations for the audio and identity. These inputs are fed into separate modules in the network in the forms of 0.35-second audio

and a still image of the target identity.

Audio. The input to the audio encoder are Mel-frequency cepstral coefficients (MFCC) values extracted from the raw audio data. 13 coefficients are calculated per sample but only the last 12 are used in our case. Each sample fed into the audio encoder is made up of 0.35-second input audio data with a sampling rate of 100Hz resulting in 35 time steps. Each encoded sample can be viewed as a 12×35 heatmap, like the audio input in Chapter 4 (see Figure 7.3).

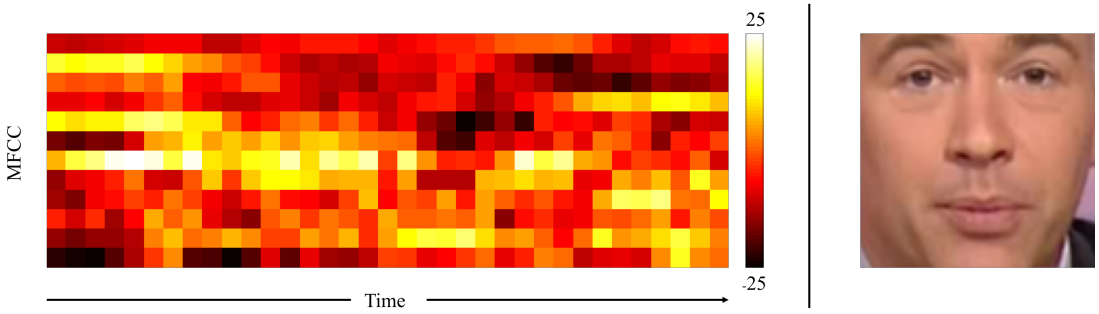

Figure 7.3: Inputs to the Vid2Speech model. **Left:** MFCC heatmap for the 0.35-second time period. The 12 rows in the matrix represent the power of the audio at different frequencies. **Right:** Still image of the speaker.

Identity. The input to the identity encoder is a single still image with dimensions $112 \times 112 \times 3$. In Section 7.3.2, we also experiment with having multiple still images as the input to the identity encoder instead of one, which significantly improves the output video quality.

7.1.2 The Architecture

The Speech2Vid architecture is given in Figure 7.4. We describe the three modules (audio encoder, the identity encoder, and the image decoder) in the following paragraphs. Note, these three modules are trained together. The deblurring module (described below in Section 7.1.3) is trained separately.

Audio encoder. We use a convolutional neural network originally designed for image recognition. The layer configurations is based on AlexNet (Krizhevsky et al., 2012)

and VGG-M (Chatfield et al., 2014), but filter sizes are adapted for the unusual input dimensions. This is similar to the configuration used to learn audio embedding in Chapter 4.

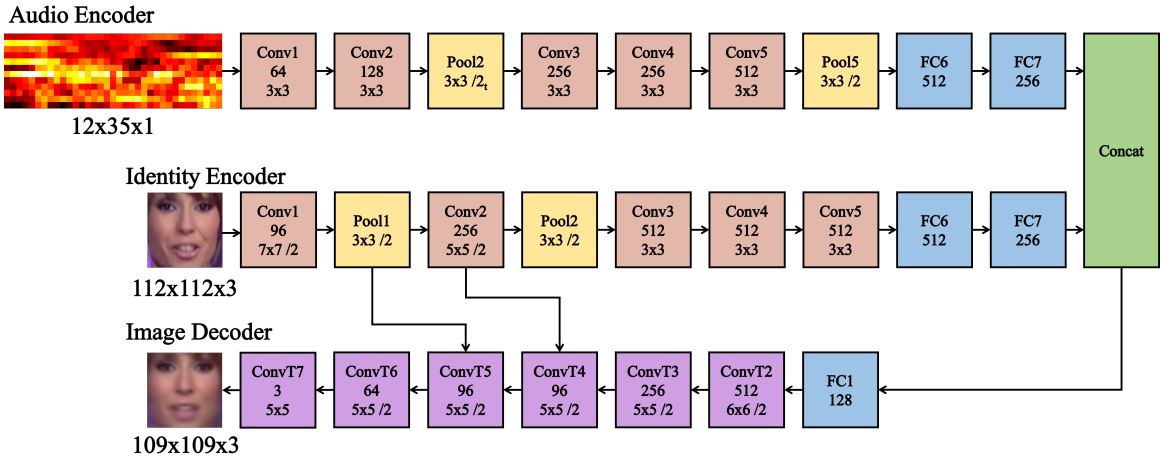

Figure 7.4: The three modules in the Speech2Vid model. From top to bottom: (i) audio encoder, (ii) identity encoder with a single still image input, and (iii) image decoder. /2 refers to the stride of each kernel in a specific layer which is normally of equal stride in both spatial dimensions except for the Pool2 layer in which we use stride 2 in the timestep dimension (denoted by $/2_t$). The network includes two skip connections between the identity encoder and the image decoder.

Identity encoder. Ideally, the identity vector produced by the encoder should have features unique for facial recognition and as such we use a VGG-M network pre-trained on the VGG Face dataset (Parkhi et al., 2015). The dataset includes 2.6M images of 2.6K unique identities. Only the weights of the convolutional layers are used in the encoder, while the weights of the fully-connected layers are reinitialized.

Image decoder. The decoder takes as input the concatenated feature vectors of the FC7 layers of the audio and identity encoders (both 256-dimensional). The features vector is gradually upsampled, layer-by-layer via transposed convolutions. See details in Figure 7.4. The network features two skip connections to help preserve the defining features of the target identity – this is done by concatenating the encoder activations with the decoder activations (as suggested in (Ronneberger et al., 2015)) at the locations shown in the network diagram.

Loss function. An L_1 loss is used (Equation 7.1), rather than L_2 that is more commonly used for image generation and in auto-encoders, as L_1 tends to encourage less blurring (Isola et al., 2017).

$$\mathcal{L}_{=} \sum_{n=1}^{N} ||\hat{y}_n - y_n|| \tag{7.1}$$

However, an image-space L_1 or L_2 loss between the prediction and the ground truth images has been known to severely penalise realistic outputs, for instance, slightly darker or lighter images that still look realistic. To mitigate this problem, we use the 'content representation' loss proposed by (Chen and Koltun, 2017, Gatys et al., 2016), where we take L_1 losses between layer activations from a pre-trained CNN, as well as between the images. Here, a pre-trained VGG Face (Parkhi et al., 2015) network is used, and the activations from 5 convolution layers (*conv1* to *conv5*) are matched (Figure 7.5), so that both fine details and global arrangements can be captured. The benefit of using this content loss can be seen in Figure 7.11.

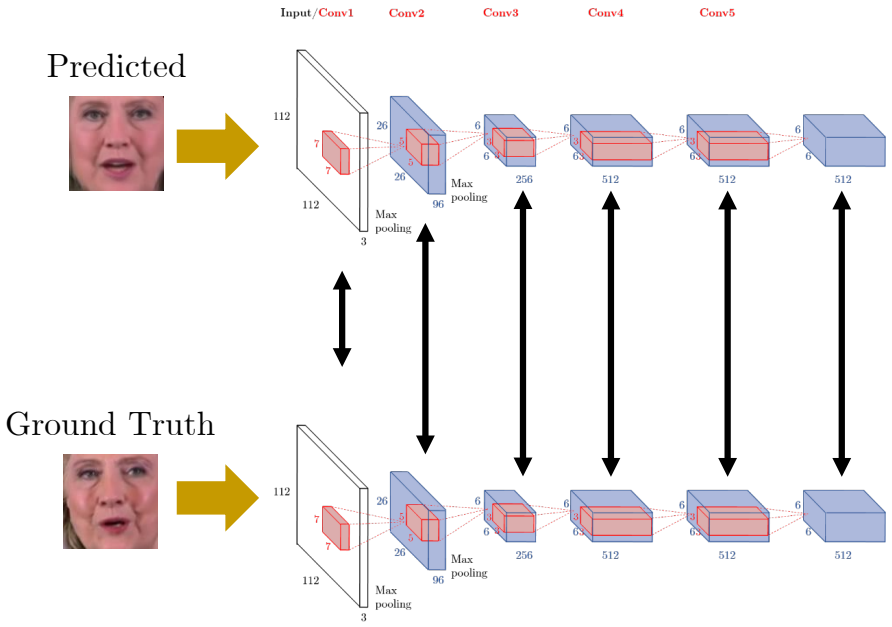

Figure 7.5: Content loss.

Training protocol. The network is trained on the video dataset described in Sec-

tion 7.2. During training, the ground truth output image of the target identity speaking the audio segment is used as supervision. The image is taken from the middle frame of the video in the 0.35-second sampling window. The image for the input identity of the speaker is randomly sampled from a different point in time, as shown in Figure 7.6). When multiple still images are used for the input identity (Section 7.3.2) we randomly sample multiple images from the same video stream.

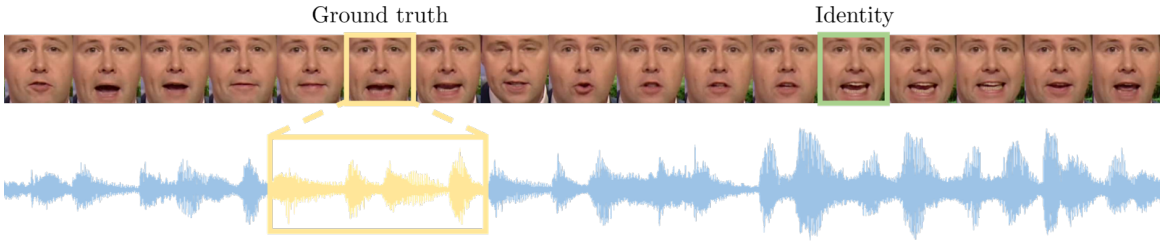

Figure 7.6: Sampling strategy for identity images during training. Identities are randomly sampled from future frames far from actual audio/output image samples.

Discussion. The network architecture is based purely on ConvNets, as opposed to the recurrent architectures often used for tasks relating to time sequences. The rationale is that the mouth shape of the speaker does not depend on anything other than the phoneme that is being said at the exact moment, and the long term context is unimportant. We find that the 0.35-second window is more than enough to capture this information. At test time, the video is generated frame-by-frame by sliding a temporal window across the entire audio segment while using the same single identity image.

Implementation details. Our implementation is based on the MATLAB toolbox MatConvNet (Vedaldi and Lenc, 2015) and trained on a NVIDIA Titan X GPU with 12GB memory. The network is trained with batch normalisation and a fixed learning rate of 10^{-5} using stochastic gradient descent with momentum. The training was stopped after 20 epochs, or when the performance on the validation set stops improving, whichever is sooner.

At test time, the network (including the deblurring layers) runs faster than twice

real-time on a GPU. This can be further accelerated by pre-computing and saving the features from the identity encoder module, rather than running this for every frame. In the case of redubbing video, the output video is generated at the same frame rate as the original video.

7.1.3 Deblurring module

CNNs trained to generate images with L_1 and L_2 losses tend to produce blurry images (Pathak et al., 2016, Zhang et al., 2016). To mitigate this problem, we train a separate deblurring CNN to sharpen the images produced by the Speech2Vid model. The model is inspired by VDSR (Kim et al., 2016), which uses a residual connection between the input and output, so that the network only has to learn the image difference. Our implementation has 10 convolutional and ReLU layers, and the layer configuration is shown in Figure 7.7.

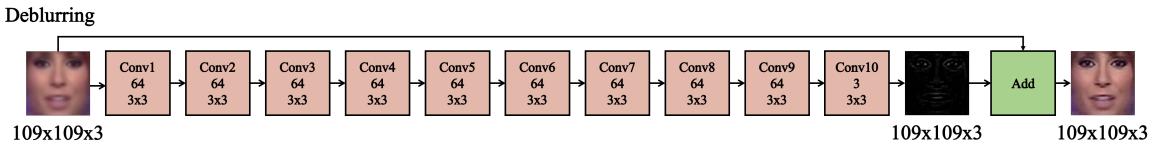

Figure 7.7: Deblurring CNN module

We train the network on artificially blurred face images (Figure 7.8), as opposed to training the network end-to-end together with the generator network. This is because the alignments between the input image, the target (ground truth) image and the generated image are not perfect even after the spatial registration (of Section 7.2), and thus avoid the deblurring network having to learn the residual coming from the misalignment.

The images that we ask the CNN to deblur are relatively homogeneous in content (they are all face images), and we find that the CNN performs very well in sharpening the images under this constraint.

Figure 7.8: Deblurring CNN input and output. **Left:** Original face image (ground truth); **Middle:** Input to the deblurring CNN; **Right:** Restored face image using the deblurring CNN.

7.2 Dataset

This section describes the strategy to prepare a large-scale dataset to train the generation network.

Video description. We start from the videos from the audio-visual corpus described in Chapter 3. This data mostly consists of broadcast news, which provide ideal training data for this task, given that a large proportion of the face tracks are front-facing, and of high quality. Moreover, the words are generally clearly spoken without too much background noise, and hence provide an easier learning environment for the network.

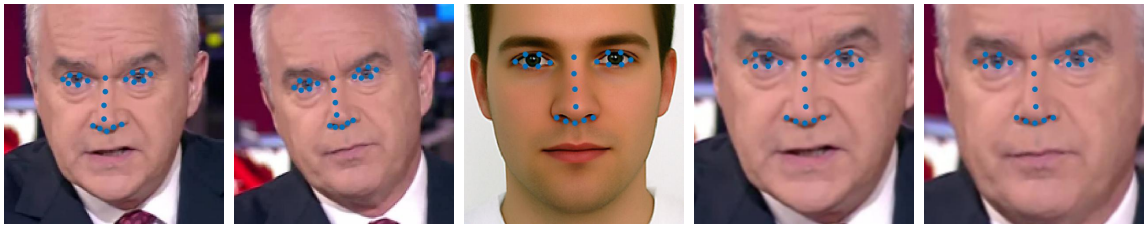

Figure 7.9: **Left pair:** Face images before registration; **Middle:** Canonical face; **Right pair:** Face images after registration with the canonical face.

Spatial registration. In order to establish spatial correspondance between the input

face (that provides the identity to the encoder) and the output face (from the decoder) in training from the ground truth frames, we register the facial landmarks between the two images. This is done by performing a similarity transformation (scale, rotation and translation) between the faces and an exemplar face with canonical position (Figure 7.9 middle). Only the landmarks on the eyes and the nose, not the mouth, are used to align the face image, as the mouth movements contain the information that we wish to capture.

Data statistics. The train, validation and test sets are divided by broadcast date. We use the splits given in Chapters 5 and 6. For every valid face track, we extract every 5th frame and the corresponding audio as samples for training and validation. Statistics on the dataset is given in Table 7.1.

Set	# Hours	# Samples
Train	37.7	678,389
Val	0.5	9,287

Table 7.1: Dataset statistics

7.3 Experiments

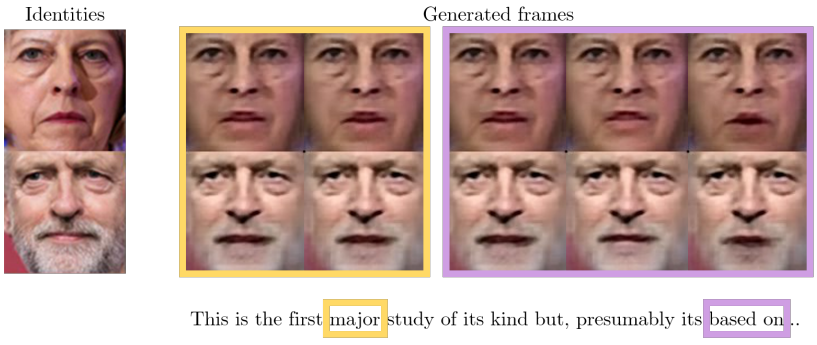

Figure 7.10: **Top row:** Identity 1 and the corresponding generated frames; **Middle row:** Identity 2 and the corresponding generated frames; **Bottom row:** Captions of the audio segment. **Best seen in video form.**

Figure 7.10 shows a visualization of the output of the model (the frames of the two segments highlighted in the captions "major" and "based on"). Note, the movement

of the mouths of the two examples reflect the sound of each word not unlike phoneme-to-viseme correspondences.

7.3.1 Preserving Identity with Skip Connections

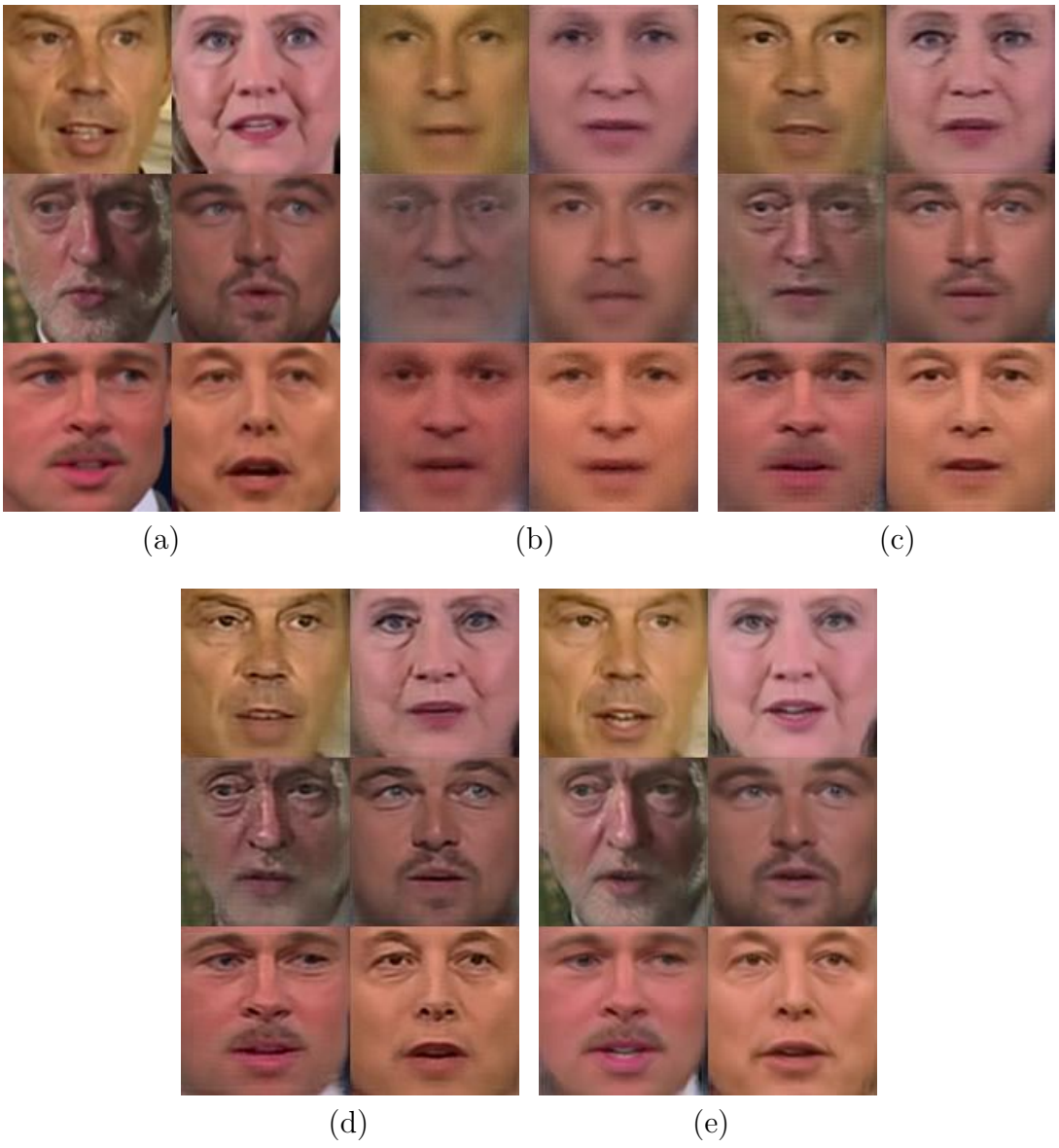

Figure 7.11: **(a)** Original still image to animate (input to the identity encoder); **(b)** Output frames without skip connection; **(c)** Output frames with skip connection and one input image; **(d)** Output frames with skip connection and five input images; **(e)** Output frames with skip connection and five input images, trained with VGG Face content loss.

Figure 7.11 shows a set of generated faces and various target identities (original stills). We observe that the skip connections are crucial to carry facial features from the input

image to the generated output – without these, the generated images lose defining facial features of target identities, as shown in the middle column. The skip connections at earlier layers (*e.g.* after conv1) were not used as it encouraged the output image to be too similar to the still input, often restricting the mouth shapes that we want to animate.

7.3.2 Preserving Identity with Multiple Still Images

Instead of a single image specifying the unique identity, five distinct images are concatenated channel-wise resulting in an input dimension of $112 \times 112 \times 15$, and the *conv1* layer of the identity encoder are modified to ingest inputs of these dimensions. In training these examples are sampled with a similar strategy to that of Figure 7.6, but multiple 'identity' images are sampled, instead of just one.

As can be seen in Figure 7.11, having multiple image examples for the unique identity enhances the quality of the generated faces. There are two reasons for this: first, with multiple example images as input, it is likely that the network now has access to images of the person showing the mouth open as well as closed. Thus, it has to hallucinate less in generation as, in principle, more can be sourced directly from the input images; Second, although the faces are aligned prior to the identity encoder, there are minor variations in the movement of the face other than the lips that are not relevant to the speech, from blinking and microexpression. The impact of these minor variations when extracting unique identity features is reduced by having multiple still images of the same person.

7.3.3 Application: Lip Transplant/Re-dubbing Videos

The Speech2Vid model can be applied to visually re-dub a source video with a different segment of spoken audio. The key stages are as follows: (i) obtain still images from

the source video for identity; (ii) generate the face video for the given audio and identity using the Speech2Vid model; (iii) re-align the landmarks of the generated video to the source video frames, and (iv) visually blend the aligned face with the source video frame.

Figure 7.12: **Top left:** Original still image; **Top right:** Generated mouth region, aligned with the original (target) face; **Bottom left:** Generated mouth region, superimposed on the original face. **Bottom right:** Generated mouth region, blended with the original face.

Alignment. Facial landmarks in the target video is determined using the method of (Kazemi and Sullivan, 2014a). A similarity transformation is used to align the generated face with the original face in the target image. Figure 7.12 (right) shows the generated face in alignment with the original face.

Poisson editing. The Poisson blending algorithm (Perez et al., 2003) blends two images together by matching gradients with boundary conditions. We use this technique to match the generated face with the source video frame, as shown in Figure 7.12. This can be used to blend the face from the same, or different identity to the source video frame.

Discussion. This method can be used to blend the generated face as a whole, or to

match only the lower half of the face. We qualitatively find that we strike the best balance between image naturalness and movement naturalness by only blending the lower half of the face, from just below the eyes.

7.4 Conclusion

We have demonstrated that the Speech2Vid model is able to generate videos of any identity speaking from any source of input audio. This work shows that there is promise in generating video data straight from an audio source. We have also shown that re-dubbing videos from a different audio source (independent of the original speaker) is possible.

Chapter 8

Word spotting in audio and sign language

One of the remarkable properties of deep learning with ConvNets is their ability to learn to classify images on their content given only supervision at the class level, *i.e.* without having to provide stronger supervisory information such as bounding boxes or pixel-wise segmentation. In particular the position and size of objects is unknown in the training images. This ability is evident from the results of the ImageNet and PASCAL VOC classification challenges (Everingham et al., 2015, Russakovsky et al., 2015), and indeed we have witnessed an annual increase in performance as new architectures and training regimes have been developed (Deng et al., 2009, He et al., 2015, Hu et al., 2017, Krizhevsky et al., 2012, Simonyan and Zisserman, 2015, Szegedy et al., 2015). Furthermore, several recent works have also shown that despite this class-level weak-supervision, the trained networks can to some extent infer the localization of the objects that the image contains (Oquab et al., 2015, Papandreou et al., 2015, Simonyan et al., 2014).

In this chapter we investigate this ability in image time series. Our aim is to obtain ConvNets that can both *classify* a time series clip as to whether it contains a target sequence or not, and *localize* the target sequence in the clip, using only class level supervision of the clip. Why is this challenging? There are two reasons, first we

consider target sequences that are very short within clips that are long – for example a target lasting less than 10 frames in a clip of hundreds of frames (a target less than 0.5s in a 12s clip); second, the supervision can be not only weak, but also noisy.

The benefits of achieving this are immense: traditionally for time series the supervision is strong, either time aligned or, if not aligned, then complete (see below); such supervision can be expensive and difficult to obtain in the visual realm. However, by relaxing the supervisory requirements, so that only labels at the clip level are used as shown here, labelling is far easier to obtain. The outcome is that it is possible to learn and localize short temporal signals, such as a hand gesture, which are virtually invisible to a non-expert. This is a 'needle in a haystack' problem, where the needle is unknown.

Usually time series are the province of Recurrent Neural Networks (RNNs), for example LSTMs (Hochreiter and Schmidhuber, 1997). Here we instead investigate non-recurrent feedforward ConvNet architectures for the classify and localize tasks through two problems: (i) speech recognition, and (ii) sign language recognition. In the first time series example, the sound is represented as an image using the standard MFCC features against time encoding, and the supervision is weak. In the second time series example, the meaning of the sign can be conveyed by a combination of hand motion, hand shape and facial expressions, and the supervision is both weak and noisy. For this example, we also investigate the image encodings of the hand motion and hand shape that are suitable for input to the ConvNet. Note, for both these problems it is a *sequence* that must be recognized – the target cannot be spotted in a single frame or time instance (as is the case for some human actions, *e.g.* playing an instrument).

We make the following contributions: (i) we show that under weak and noisy supervision it is possible to learn to classify image time series clips as to whether they contain a short target sequence or not, using a suitable ConvNet architecture; (ii) we show that the location of the target sequence can be obtained by back-propagating

through the network in the manner of Simonyan et al. (2014); and (iii) to achieve this we propose and investigate image time series encodings and corresponding ConvNet architectures. We also investigate the effects of different loss functions for this task.

To the best of our knowledge, this is the first time ConvNets have been used to recognise and localise complex temporal sequences, such as the gestures in sign language, in image time series using such weak annotation in training. The outcome is that it is possible to learn to recognize and localize individual signs depending on both hand motion and hand shape in long temporal clips. The performance far exceeds previous work in this area in terms of supervisory requirements and generalization across signers.

More generally, the method is applicable to other situations that involve plucking sequences out of long clips. For example, keyword spotting in always on speech recognition, identifying pathologies in medical time series data, or localizing human actions in videos.

8.1 Representations and architectures for recognizing time sequences

Given an audio clip of speech or a video clip of sign language gestures, the objectives are to determine if a target sequence is present in the clip, and, if so, where it is. We set up two problems with different levels of difficulty: (i) finding words spoken in an audio clip using supervision that is *weak*; and (ii) finding words signed as gestures in sign language video using supervision that is both *weak and noisy*. For gestures, both the *motion* and *shape* of the hands are considered. In both problems there are two issues: (i) how to encode the image time series, and (ii) the design of the ConvNet architecture to recognize the target sequence. Of course, these two issues are coupled.

8.1.1 Audio recognition

The input audio data is Mel-frequency cepstral coefficient (MFCC) (Davis and Mermelstein, 1980) values. This is a representation of the short-term power spectrum of a sound on a non-linear mel scale of frequency. 12 mel frequency bands are used at each time step. The features are computed at a sampling rate of 100Hz, giving 600 time steps for a 6-second input signal.

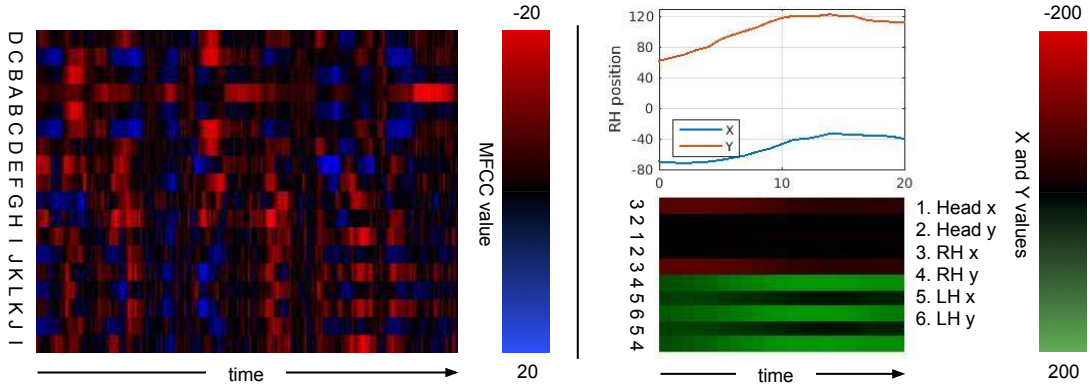

Figure 8.1: **Temporal representations as heatmaps for audio (left) and motion (right).** The audio image encodes the 12 MFCC features, and these are reflected vertically (at A and L) to mitigate boundary effects for the filters, resulting in an image of height 18 pixels. The represented audio is *'you've got a simple little dish'*. For the motion, the two channels representing position are shown, with red representing negative values and green positive. The represented motion is for the BSL sign for *'valley'* (a V-shape drawn with both hands moving downwards). Note that row 3 representing the right-hand (left in the image) x-values get dimmer with time as x increases and the values become less negative, whereas row 4 representing y-values get brighter with time as y increases and the values become more positive.

Representation. The audio is encoded as a heatmap image representing MFCC values for each time step and each mel frequency band (see Figure 8.1, left). The top and bottom three rows of the image are reflected to reduce boundary effects. Previous work (Geras et al., 2015) has also attempted to train image-style ConvNet for such inputs, but only with strong (frame-level) supervision.

Architecture. We use a convolutional neural network inspired by those designed for image recognition. Our layer architecture (Figure 8.2) is based on AlexNet (Krizhevsky et al., 2012), but with modifications. AlexNet takes a square image of size 224×224

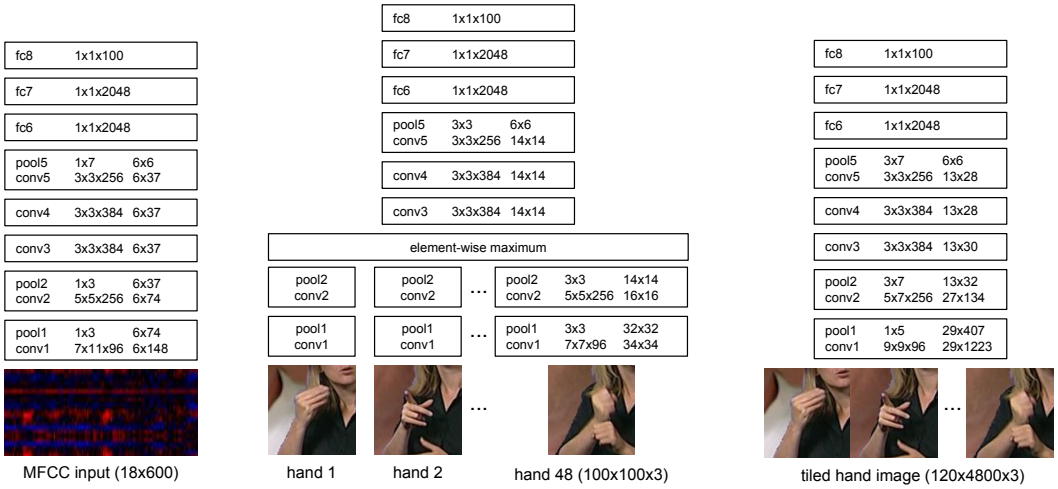

Figure 8.2: **ConvNet architectures for Audio (left) and Hand shape (middle/ right).** Max-pooling is used in all pooling layers. The five numbers following each convolutional layer specify: the size of the filters, the number of channels, and the resolution of the layer.

pixels, whereas our input size is at least 600 pixels (the number of time steps) in the time-direction, and only 18 pixels in the other direction (so the input image is 18×600 pixels). We introduce an aggressive max-pooling layer of size 7 in time at *pool5*. The design intention is as follows: the receptive field size of a pixel in *conv5* is 160, which is equivalent to 1.6 seconds. Given that the relevant information (the target word) is only 0.5 seconds in length, we conjecture that the target sequence is already encoded in a single activation in the time-domain. Max-pooling at this level should therefore collect the strong activations from the relevant parts and discard weak activations from the rest.

8.1.2 Motion recognition

The data to be represented consist of the pixel coordinates of the two hands and head in each frame (i.e. six values), hereafter referred to as keypoints. Details of how these points are obtained are given in Section 8.3.

Representation. Our choice of representation for the motion information is deliberately similar to that of audio. The position of keypoints on the body (*e.g.* head,

hands, etc.) are stored as intensity values in a heatmap (Figure 8.1, right). Two bytes (the first two channels) are used to store the position, which requires more precision, and one byte (the third channel) is used for the velocity (the frame difference in position).

Discussion. As well as its similarity to the audio encoding, this motion encoding was chosen as one that should be suited to convolutional filter learning. For example, horizontal temporal derivative filters on the brightness values can measure if the hand is moving upwards (negative output) or is stationary (zero output). Filters covering several rows can detect if the hands are moving together or not, etc. This encoding has the properties of being compact and minimal. We did consider several other representations, but rejected these as they resulted in much larger input images. For example, the motion could be encoded as optical flow in the manner of (Simonyan and Zisserman, 2014), but that would require two images per frame. A second possibility is to build Motion History Images (Ahad et al., 2012), and a third is to simply encode the entire person per frame with no keypoint detection – though in this case the background motion would be extremely distracting and challenging (see Figure 8.5).

Architecture. The architecture of this network follows that of the audio network described in Section 8.1.1, given that the input representation is of the same format.

8.1.3 Hand shape recognition

The data consist of a square image region centred on the right hand keypoint in each frame. A crop of 120×120 pixels is taken around the hand, at a sampling rate of 4 frames per second for 12 seconds, resulting in 48 RGB images. Since a sign typically lasts for less than half a second, only one of these 48 images will be relevant to the target sign.

Representation. We consider two architectures for 'ingesting' these 48 images.

The first uses multiple towers and is inspired by representations for human action recognition (Ng et al., 2015) with a tower for each of the 48 images. The second encodes the 48 images into a single time-ordered composite (similar to the spatial composites of (Papandreou et al., 2015)).

Architecture 1 – multiple stream. The architecture is shown in Figure 8.2 (middle) – there are 48 towers with common *conv1* and *conv2* layers, each of which takes an input frame. The activations from the towers are combined after *pool2* using an element-wise maxima $A_{x,y,c}$ taken across different towers: suppose that the activations of the ith tower at *pool2* are $a^i_{x,y,c}$ where c is the channel index, and x and y the pixel locations, then $A_{x,y,c} = \max_i(a^i_{x,y,c})$. The intention is that the combination layer collects the strongest activations from the most relevant images. Other positions for the combination layer were tried (e.g. for the *conv1* activations), but were found to be less effective than combining the *pool2* activations.

Architecture 2 – tiled input. Here, the 48 images are composed into a single image by tiling in the temporal direction. This results in a 120 $\times$ 4800 pixel image with three channels for RGB. Consequently, the output dimensions from the first layers are much larger in the horizontal (temporal) dimension than the vertical due to the large aspect ratio of the input image. The activations are pooled aggressively (horizontal size of 7 pixels) at *pool2* and *pool5* in the horizontal direction. *pool2* was chosen (rather than earlier) because the receptive field size at the input to this layer is large enough to cover a single input image, hence we can assume that the local details for the whole hand have been captured in the *pool2* activations. The rationale is similar to the pooling strategy in the audio case.

Discussion. The second representation is simple and surprisingly effective. Virtually no difference in validation performance is observed between the two networks, despite a considerable overhead in the former (*e.g.* the number of free parameters is similar, but they must be loaded 48 times; the memory required to store the activations

is around 50% greater; and many operations are faster on a single large image vs. multiple small images). Hence, the tiled-input network is the only one considered from this point on.

An obvious alternative might be to stack the multiple images as input channels to one tower, i.e. a 48 channel input ($\times$ 3 for RGB). This method was not chosen because the hands in each frame are not exactly spatially registered due to small hand movements and tracking noise. These small localization offsets vary between frames. This means that filters would not be able to learn anything useful in the time direction of the 48 channels. The two representations adopted avoid this lack of registration problem. For example, in the first architecture the *pool1* layer can compensate for the localization offsets.

8.1.4 Two-stream gesture recognition

In British Sign Language, the hand shape or hand motion or both may be necessary in order to define a sign. To enable both to be used for sign recognition the hand motion and hand shape networks are combined using late fusion, by concatenating the *pool5* activations, before the fully-connected layers. Late fusion is chosen since prior methods (Simonyan and Zisserman, 2014) have found that early fusion (e.g. at *conv2*) leads to one or the other network dominating. Also, as in (Simonyan and Zisserman, 2014), the two networks are initially trained separately, and then fused and trained together to give a single output.

8.1.5 Loss functions

We experiment with three loss functions:

(i) the *cross-entropy loss*, which is commonly used in multi-class classification problems such as ImageNet: $L(S, c) = -q(c) \log P(c) = -q(c) \log \frac{\exp S(c))}{\sum_q \exp S(q)}$ where $q(c)$ is

$1/n$ for all positive classes, n is the number of positive labels, S is the class score (*fc8* output), and c is the class index.

(ii) the *weighted hinge loss*, as a binary classification problem (present/ not present) for each class, which is weighted to deal with imbalances in the training data: $L(S, l) = w_l \max(0, 1 - lS)$ where l is the binary class label (present/ not present) and w_l is the ratio n_{neg}/n_{pos} for each class when $l = 1$, and 1 when $l = -1$.

(iii) the *weighted binary logistic loss*, again as a binary classification problem for each class: $L(S, l) = w_l \log(1 + \exp(-lS))$ where the notations are the same as above.

All three losses can be used in the case that multiple labels are positive for a clip. The cross-entropy loss is usually used in the exclusive multi-way classification case, such as ImageNet, where only one label is positive, but is not limited to this (van den Oord et al., 2016b).

Allowing multiple labels per clip is beneficial for two reasons: (i) we eliminate the need to discard training sequences that belong to multiple classes, hence increasing the amount of training data available. (ii) it improves the ratio of supervision per subtitle in the training data, in our case, by a factor of 1.22. We expect this benefit to be more significant if the vocabulary size is scaled up, given that the probability of two or more words appearing in a sentence is then higher. The performance under these different loss functions is evaluated and compared in Section 8.4.2.

8.1.6 Localisation via backprop

The objective here is, once the networks have been trained to classify the clip, localize the target sequence within (positive) clips. Simonyan et al. (2014) have shown that it is possible to infer the localization of visual objects in an image as a saliency map for a network trained to classify images. We adapt this method to time series to find the salient temporal intervals in the input signal that have high influence on the class

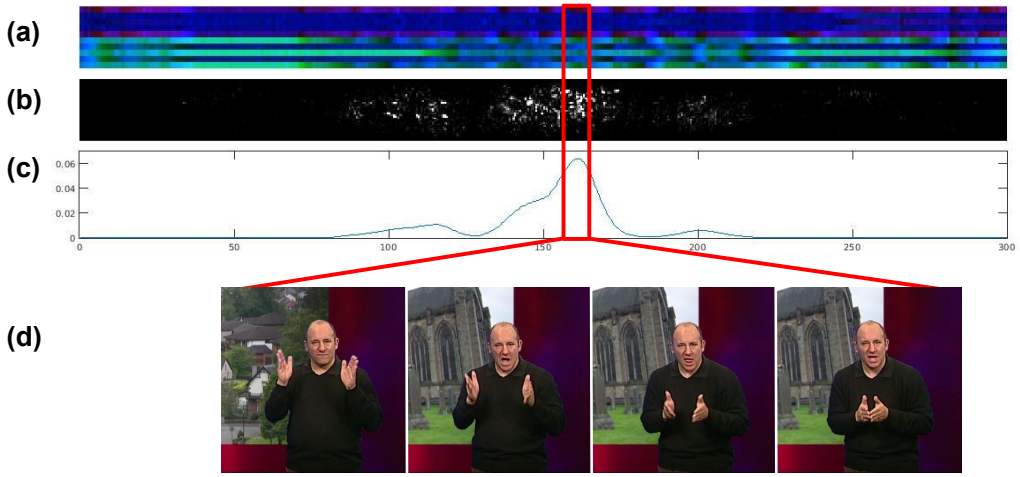

Figure 8.3: Salient region for '*valley*', localised only using motion information. The sign for '*valley*' is a large V drawn with both hands. (a) Motion heatmap; (b) Saliency map; (c) Saliency over time, after Gaussian filtering; (d) The corresponding target sequence.

score.

The method proceeds by approximating the relation between the class score S and the input image I (represented as a vector) as $S(I) = w^T I + b$. The vector w is the same size as the input image, and the magnitude of its elements signify the influence of the corresponding elements of the image on the class score. Hence the magnitude of w determines a saliency map on the image. The vector w can be obtained as $w = \frac{\partial S_c}{\partial I}\big|_{I_0}$ and this derivative is obtained by back-prop from the class score $S_0(I_0)$ to the image.

In our case, the derivative w shares the dimensions of the input time series. We compute saliency $M_{i,j}$ at position i, j as $M_{i,j} = \max_c |w_{i,j,c}|$ (*i.e.* the channel-wise maximum over every pixel, the result is shown in Figure 8.3b). A 2-dimensional Gaussian is used to smooth the signal, and then the column-sum is taken to obtain a score function against time (Figure 8.3c). The localisation method is common for all inputs types (*i.e.* audio, motion and hand).

Figure 8.4: Salient sequences for '*beef*' (left) and '*winter*' (right) are highlighted in green. Localised only using the hand shapes. The sign for '*beef*' is a thumb pointing towards the neck. The sign for '*winter*' is two clenched fists tremoring.

8.2 Dataset

We collect a novel dataset for this task that is used for recognition and localisation of both spoken words and sign gestures. The dataset consists of 890 high-definition TV broadcast videos aired between 2010 and 2016, each of which is between 30 to 60 minutes in length. The videos are '*sign-interpreted*' which means that the programmes have a signer in the corner of the screen. We use the video, audio and the corresponding subtitles as the weakly labelled training data for the tasks. We were able to collect a large amount of training data, given that many hours of sign-interpreted programmes are broadcast every day. The format of the dataset is similar to that of Pfister et al. (2013), but orders of magnitude larger in scale. Table 8.1 shows the key statistics of the dataset.

Label	**Single**	**Multiple**
Total # of programmes	890	890
Total video length (hours)	678	678
Vocabulary size	100	100
Total # of subtitles	662,165	662,165
Useful # of subtitles	50,000	104,247
Min./max. instances per class	500/500	500/2000
# of words per subtitle	9.96	10.11
In-vocab words per subtitle	1.00	1.22

Table 8.1: Dataset statistics

From a vocabulary of over 40K words, 100 target words are selected primarily based on their frequency of appearance in the programmes. Stop words such as '*a*', '*the*',

Figure 8.5: Sign-interpreted TV

and words with more than one meaning such as '*match*' and '*bank*' are excluded from selection. We also selected programmes primarily on the genres of 'wildlife' and 'cooking', in order to reduce this polysemy problem. For the audio recognition task, very short words are also excluded. All target words appear at least 500 times, and for single-label classification, the classes are balanced, with exactly 500 instances each. For multi-label classification, it is not possible to balance the classes, however the problem of unbalanced training data is overcome with per-class weighting in our loss functions.

A sequence is extracted for each occurrence of the target word in the subtitles. The alignment between the subtitle and the signs is imprecise, therefore the temporal window is padded by additional 8 seconds for video and 2 seconds for audio. The total length of each training sequence is 300 frames (12 seconds) for video, and 6 seconds for audio, whereas most subtitles are shorter than 4 seconds.

The dataset is divided into training, validation and test subsets (80:10:10) in chronological order, the test set being the oldest. This prevents the training and the testing

from happening on the same videos. For sign language recognition, the evaluation is performed on an independent dataset rather than our own.

Discussion. This dataset is particularly challenging for a number of reasons: (i) the word order in the subtitle is not the same as the order in which they are signed, hence we cannot estimate when the word might be signed; (ii) a word that appears in the subtitle may not be signed (the proportion of signed video which actually contains the target word is only 20-60%, depending on the word); (iii) the alignment between the sign and the subtitle is unknown and the offset can be more than 5 seconds; (iv) the contents are signed by 50 different signers and the audio has hundreds of speakers; (v) there is a large variation in content, from 'cooking' to 'wildlife', broadcast over a period of 6 years. (For the audio recognition problem, the first two do not apply.)

8.3 Implementation details

8.3.1 Data preparation

This section describes how the training data is generated. From a single broadcast (Figure 8.5), we extract information in three channels – the audio, the sign-interpreted video and the subtitle text.

Figure 8.6: Output from the upper-body tracker.

Text extraction and processing. British TV transmits subtitles as bitmaps rather than as text, therefore subtitle text is extracted from the broadcast video using stan-

dard OCR methods (Buehler et al., 2009, Everingham et al., 2006). Subtitles are stemmed (*e.g.* '*played*', '*played*', '*playing*' all become '*play*') and stop words (*e.g.* '*a*', '*the*') are removed.

Audio pre-processing. For each sequence, we extract MFCC features every 10ms where the features are computed over a 25ms interval using the same implementation as that in Chapters 4 and 7. The input to the network has 600 time steps, but 650 samples are taken over 6.5 seconds so that it can be jittered for training augmentation using random cropping. The 13-dimensional output vector represents the frequencies relating to the human voice.

Upper-body tracking. We use the ConvNet-based upper body pose estimator of (Pfister et al., 2015) to track the head, elbows and hands of the signer. The input to the tracker is a crop of the signer around 900 $\times$ 900 pixels, from a Full HD (1920 $\times$ 1080) frames. The pose estimator generates a confidence score for each keypoint, and one usually takes the maximum to estimate the location of the keypoint. However, the returned confidence heatmaps for some keypoints often have a multi-modal distribution (*e.g.* the left-hand detector gives high confidence for both hands), which can give incorrect estimates. Dynamic programming in time corrects many of these errors by optimising between the framewise confidence and the distance of the keypoints between neighbouring frames.

8.3.2 Training

Data augmentation. Applying data augmentation often improves validation performance and reduces overfitting in ConvNet image classification tasks (Krizhevsky et al., 2012). For the audio recognition task, we vary playback speeds and take random crops in the time domain. The audio track is played at three different speeds (0.9$\times$, 1.0$\times$, 1.1$\times$) in training. This affects the audio frequencies as well as the time of the track, hence the MFCC features and the heatmaps have to be re-computed.

Similar augmentation methods are used for the motion image. The video is played back at three different speeds, for which the velocities must be recomputed. The coordinates of the keypoints (the tracker output) are randomly shifted. The brightness of the heatmap image is also varied, which is equivalent to spatially scaling the input video.

The hand shape images are augmented in a similar manner to ordinary ImageNet training. Random crops are taken and the brightness is varied, but the images are not flipped as it can change the meaning of the sign.

Details. Our implementation is based on the MATLAB toolbox MatConvNet (Vedaldi et al., 2014). The network is trained with batch normalisation (Ioffe and Szegedy, 2015). Despite this, a slow learning rate of 10^{-3} to 10^{-4} is used to get a stable learning, due to the noisiness of the data. The training is stopped after 100 epochs, or when the validation error did not improve for 5 epochs, whichever is sooner.

8.4 Experiments

Experiments are conducted on each of the image series types described in Section 8.1. In each case there are two task to be performed: (i) to *classify* the clip (*e.g.* a 12s time series in the case of video) as to whether it contains the target sequence or not, and (ii) to *localize* (the temporal extent of) the target sequence within the clip. Both tasks are evaluated using Precision Recall curves. For the localization task it is necessary to use a criterion to decide if the target sequence has been correctly localized or not. Here we use a threshold on a temporal overlap score, in a similar manner to that of PASCAL VOC (Everingham et al., 2015).

In the following experiments the methods are trained and evaluated on the train/val and test partitions of the dataset of Section 8.2. However, the noise in the supervision (that only 20–60% of the words in the subtitle are actually signed) presents a

problem for evaluation as even a perfect classifier would not score well under such circumstances. We deal with this problem by giving results on both our own test set and also on an external test set (Pfister et al., 2014b) for which the labelling is not noisy. For audio there is no evaluation problem as the labelling on our test set is not noisy – words in the subtitles are spoken.

8.4.1 Audio recognition

Task. The task is to classify a 6-second audio clip as to whether it contains a target word or not, and provide a score of confidence that can be used for the Precision-Recall evaluation. The method is trained on the train/val partition of the Dataset, and evaluated on the test partition.

Results. Precision-recall curves for classification are shown in figure 8.7. For the classification task, strong performance is achieved on a variety of words, the longer multi-syllable words performing particularly well, whereas short words or words that rhyme (*e.g.* 'cat', 'rat', 'mat') are more difficult to learn because the MFCC features for these words are very similar. Ground truth labels are not available for the localisation task, but qualitative results are included in the supplementary material, and an indication of the strength of the saliency map is shown in Figure 8.8.

8.4.2 Gesture classification and localisation

External test dataset. The test dataset is based on the BBC sign language videos of Pfister et al. (2014b). This dataset is independent from our main dataset, and the format is the same as the data used by Pfister et al. (2014a), which makes it useful for comparisons. A number of words that appear frequently both in our training dataset and in the external test set (see Table 8.2) have been manually annotated at frame-level, which is used to evaluate both classification and localisation tasks. These

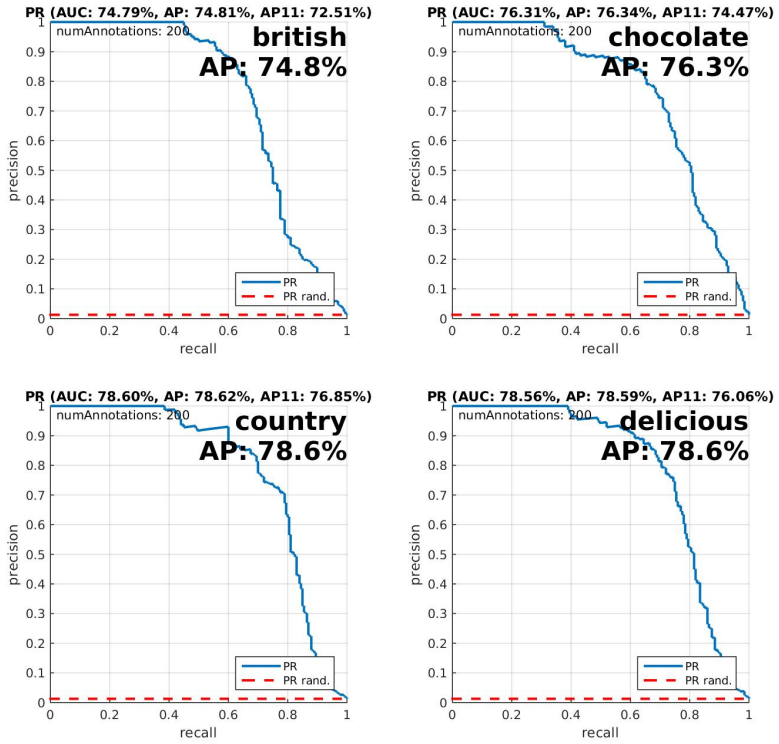

Figure 8.7: PR curves for **audio classification.**

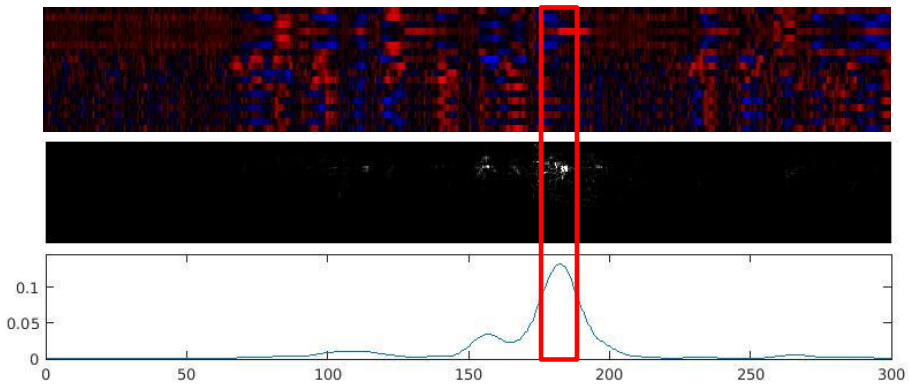

Figure 8.8: Localisation of the audio signal for '***dish***', from the sentence '*you've got a simple little* ***dish***.'

videos are from different domains to our training dataset, and they are only available in a lower resolution of 720×405 pixels. Given that our networks are trained using hand image crops from Full HD (1920×1080) videos, the test videos must be scaled up to a size suitable for the network.

Evaluation protocol for classification. The task here is to classify whether or not a gesture (corresponding to a word) is present in a 12-second temporal window. The

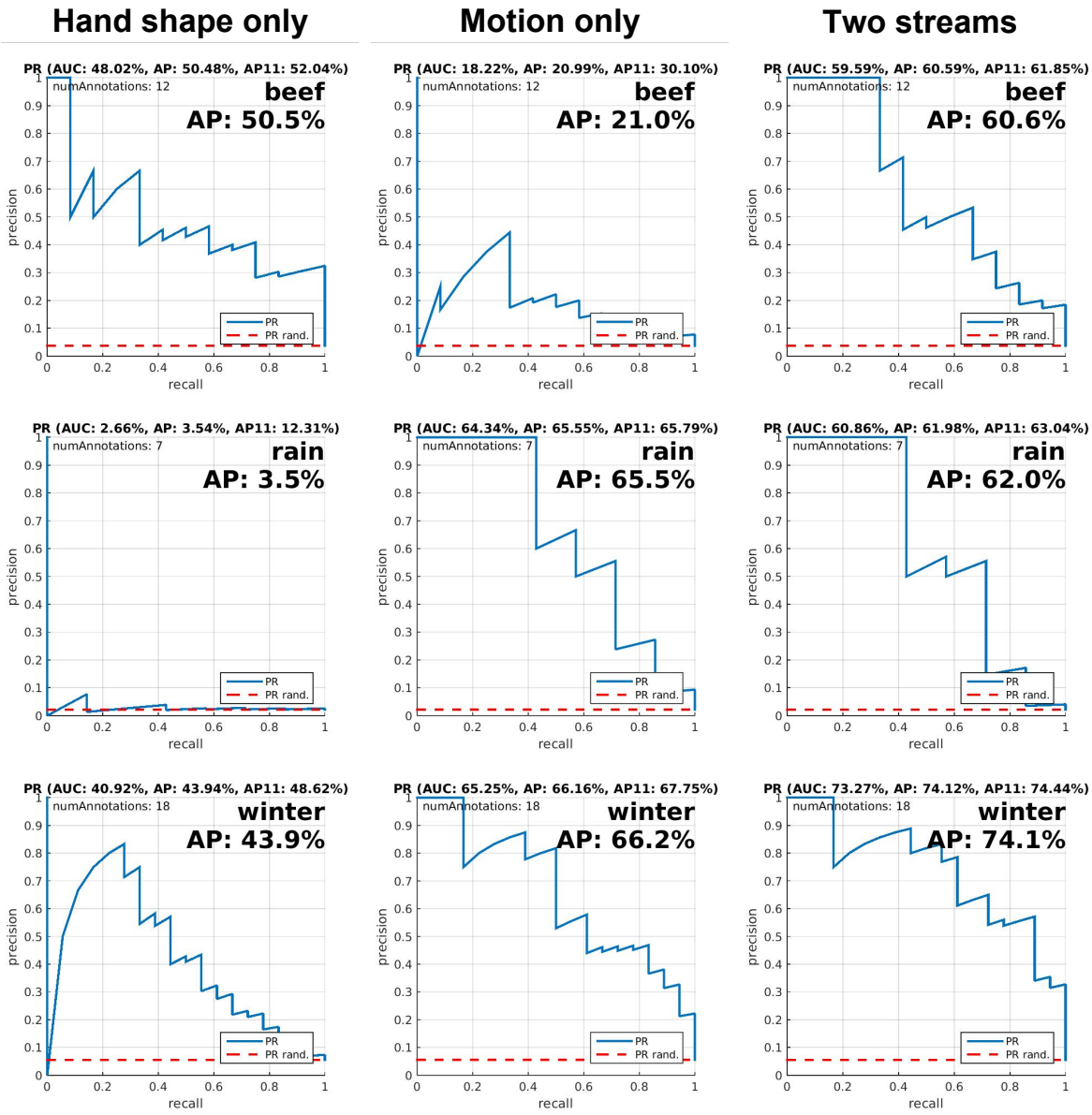

Figure 8.9: PR curves for **gesture classification** on the external BBC test set. **Left-most column:** hand shape network. **Middle column:** keypoint motion network. **Right-most column:** Combined two-stream network. The word *'beef'* is easily recognised from the hand shape, but more difficult from the motion. *'rain'* is recognisable only from the motion. Both motion and the hand shape give cues for *'winter'* but the two-stream network gives the best performance.

confidence of a gesture being present in the window (class score S_c) is given by the corresponding value of *fc8*.

Evaluation protocol for localisation. The task is to localise the temporal interval of the half-second target gesture within the 12-second window and provide a ranked

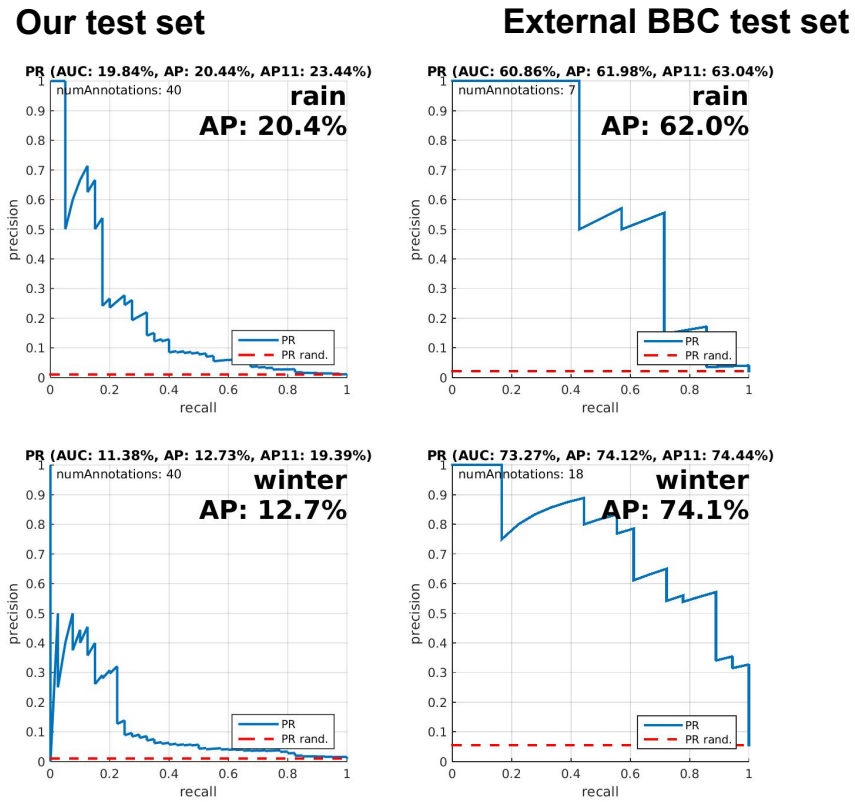

Figure 8.10: PR curves for **gesture classification.** The two plots on the **left** show the results on our own test dataset. The plots on the **right** show results on the external BBC test set. The performance on the external test dataset is considerably better, even though the domain and the format are different from our training dataset. This comparison demonstrates the extent of the supervision noise in our dataset (since only a fraction of the words appearing in the subtitles are actually signed).

list of temporal windows in the order of confidence. If the gesture overlaps at 50% with the ground truth, the localisation is deemed successful.

Classification results. Precision-recall curves for the gestures are given in Figure 8.9. The results demonstrate that some signs can only be recognised from keypoint motions, whilst some others are distinguished by hand shape, which provides the rationale for using a combined network. The model learns to recognise signs that either have discriminative motion and/or hand shapes. However, the method has difficulties learning some signs (*e.g. wet, south*) where neither are particularly discriminative from other general motions in a typical 12-second window.

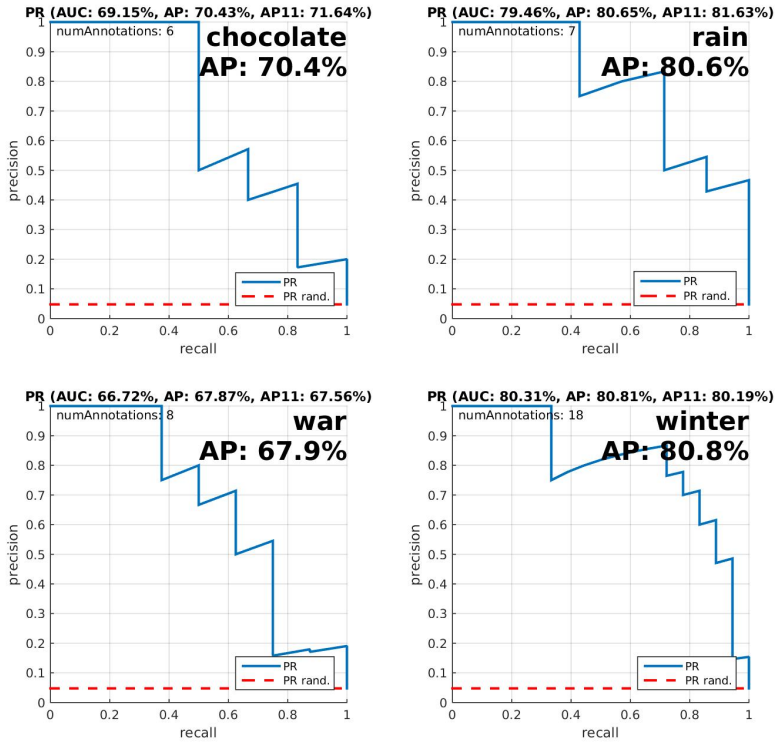

Figure 8.11: PR curves for **gesture localisation** on the external BBC test set. See Table 8.2 for further results.

Loss function	*Label*	beef	chocolate	milk	pig	rain	school	soup	valley	war	winter	***mAP***
Cross-entropy	Single	48.1	11.4	27.5	25.6	74.1	21.8	**82.7**	23.4	40.0	66.0	42.1
	Multiple	50.3	58.8	29.2	15.6	42.3	19.5	41.5	37.7	**75.2**	51.7	42.2
Weighted hinge	Single	22.6	11.1	47.4	32.3	69.1	26.9	35.9	15.4	35.8	33.1	33.0
	Multiple	48.9	34.5	21.8	**62.4**	65.3	12.5	26.8	12.8	54.7	61.2	40.0
Weighted logistic	Single	43.7	45.4	44.7	51.4	76.6	27.8	18.5	18.9	49.7	61.7	43.9
	Multiple	**59.0**	**70.4**	**58.8**	49.5	**80.6**	**48.9**	62.3	**49.1**	67.9	**80.8**	**62.7**

Table 8.2: Average Precision for **gesture localisation** on the external BBC test set.

Localisation results. The results in Table 8.2 show that the choice of loss function is very important, particularly when dealing with unbalanced classes and multiple positive labels per clip. For example, the cross-entropy loss works well for training with the single-label data, but fails to benefit from the more extensive multi-label data. Our best model trained using the weighted logistic loss compares well with the state-of-the-art.

The words that appear frequently in our dataset are different from those of Pfister et al. (2013) and Pfister et al. (2014a); therefore we must compare the performance

figures with caution. Our test set annotation methods and the evaluation protocol closely follow that of Pfister et al. (2014a). Comparing our PR curve (Figure 8.11) to Figure 7 of Pfister et al. (2014a), it is clear that our localisation performance is competitive with the strongly supervised method of Pfister et al. (2014a) (which uses a dictionary) and far exceeds the weakly supervised method of Pfister et al. (2013). For example, our average precision on *'winter'* (which appears in both our work and theirs) is 81%, Pfister et al. (2014a) is 50% and Pfister et al. (2013) is 18% (note, the performance figures of Pfister et al. (2013, 2014a) are not available, so values are estimated from the graphs). The other words in our evaluation do not appear in Pfister et al. (2013, 2014a), but the performance figures are competitive with those that do. It is notable that the performance of strongly supervised methods can be matched, particularly when the network has never been explicitly trained to localise these signals.

Computation time. The classification and localisation of the salient regions is extremely fast. Both tasks for a 12-second window can be done with a single forward-backward pass at run time, which takes approximately 100ms for the hand shape network and 5ms for the motion network. This is dependent on the video being pre-tracked, which takes approximately 10-seconds for a 12-second clip on a GPU. In comparison, the method of Pfister et al. (2013) takes over 100 seconds to process a clip of the same length.

8.5 Conclusion

This chapter makes two key contributions: (i) we show that with a suitable input encoding and ConvNet architecture, it is possible to recognise signals in image time series using only very weak and noisy annotation; (ii) we have obtained state-of-the-art performance in localisation of the target signal from a network that is trained only to classify the signals. Our approach is directly applicable to any scenario that

involves identifying short target sequences given only clip level annotation.

Chapter 9

Speaker identification from audio

In this chapter, we depart from the problem of 'what' is being said, and explore methods for recognising 'who' is saying it.

Speaker recognition under noisy and unconstrained conditions is an extremely challenging topic. Applications of speaker recognition are many and varied, ranging from authentication in high-security systems and forensic tests, to searching for persons in large corpora of speech data. All such tasks require high speaker recognition performance under 'real world' conditions. This is an extremely difficult task due to both extrinsic and intrinsic variations; extrinsic variations include background chatter and music, laughter, reverberation, channel and microphone effects; while intrinsic variations are factors inherent to the speaker themself such as age, accent, emotion, intonation and manner of speaking, amongst others (Stoll, 2011).

As noted throughout this thesis, Deep Convolutional Neural Networks (CNNs) have given rise to substantial improvements in speech recognition, computer vision and related fields due to their ability to deal with real world, noisy datasets without the need for handcrafted features (He et al., 2015, Krizhevsky et al., 2012, Simonyan and Zisserman, 2015). One of the most important ingredients for the success of such methods, however, is the availability of large training datasets, which is lacking in the field of speaker recognition (Section 2.2.4).

This chapter has two goals. The first is to propose a fully automated and scalable pipeline for creating a large-scale 'real world' speaker identification dataset. By using visual active speaker identification and face verification, our method circumvents the need for human annotation completely. We use this method to curate `VoxCeleb`, a large-scale dataset with hundreds of utterances for over a thousand speakers. The second goal is to investigate different architectures and techniques for training deep CNNs on spectrograms extracted directly from the raw audio files with very little pre-processing, and compare our results on this new dataset with more traditional state-of-the-art methods.

`VoxCeleb` can be used for both speaker identification and verification. Speaker identification involves determining which speaker has produced a given utterance, if this is performed for a closed set of speakers then the task is similar to that of multi-class classification. Speaker verification on the other hand involves determining whether there is a match between a given utterance and a target model. We provide baselines for both tasks.

9.1 Dataset

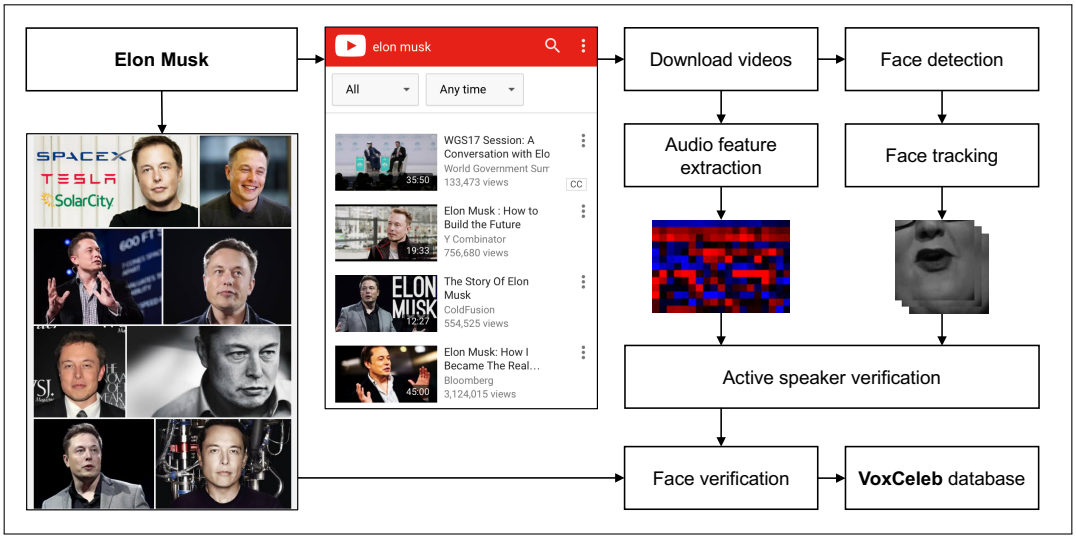

Figure 9.1: **Left:** Data processing pipeline

VoxCeleb contains over 100,000 utterances for 1,251 celebrities, extracted from videos uploaded to YouTube. The dataset is gender balanced, with 55% of the speakers male. The speakers span a wide range of different ethnicities, accents, professions and ages.

Videos included in the dataset are shot in a large number of challenging multi-speaker acoustic environments. These include red carpet, outdoor stadium, quiet studio interviews, speeches given to large audiences, excerpts from professionally shot multimedia, and videos shot on hand-held devices. Crucially, all are degraded with real world noise, consisting of background chatter, laughter, overlapping speech, room acoustics, and there is a range in the quality of recording equipment and channel noise. Unlike the SITW dataset, both audio and video for each speaker is released. Table 9.1 gives the dataset statistics.

# of POIs	1,251
# of male POIs	690
# of videos per POI	36 / 18 / 8
# of utterances per POI	250 / 116 / 45
Length of utterances (s)	145.0 / 8.2 / 4.0

Table 9.1: VoxCeleb dataset statistics. Where there are three entries in a field, numbers refer to the maximum / average / minimum.

We use multi-stage approach for collecting the large-scale speaker recognition dataset, starting from YouTube videos. Using this fully automated pipeline, we have obtained hundreds of utterances for over a thousand different Persons of Interest (POIs). The pipeline is summarised in Figure 9.1 left, and key stages are discussed in the following paragraphs:

Stage 1. Candidate list of POIs. The first stage is to obtain a list of POIs. We start from the list of people that appear in the VGG Face dataset (Parkhi et al., 2015) , which is based on an intersection of the most searched names in the Freebase knowledge graph, and the Internet Movie Data Base (IMDB). This list contains 2,622 identities, ranging from actors and sportspeople to entrepreneurs, of which approximately half are male and the other half female.

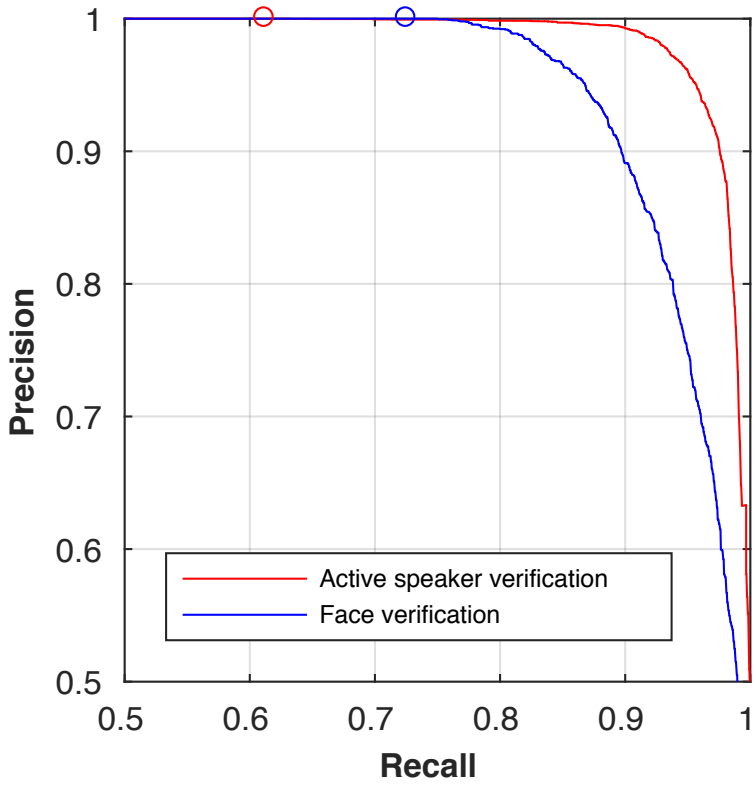

Figure 9.2: Precision-recall curves for the active speaker verification (using a 25-frame window) and the face verification steps, tested on standard benchmark datasets (Chakravarty and Tuytelaars, 2016, Parkhi et al., 2015). Operating points are shown in circles.

Stage 2. Downloading videos from YouTube. The top 50 videos for each of the 2,622 POIs are automatically downloaded using YouTube search. The word 'interview' is appended to the name of the POI in search queries to increase the likelihood that the videos contain an instance of the POI speaking, and to filter out sports or music videos. No other filtering is done at this stage.

Stage 3. Face tracking. This step is based on the video processing pipeline described in Chapter 3. The HOG-based face detector (King, 2009) is used to detect the faces in every frame of the video. Facial landmark positions are detected for each face detection using the regression tree based method of (Kazemi and Sullivan, 2014b). The shot boundaries are detected by comparing colour histograms across consecutive frames. Within each detected shot, face detections are grouped together into face tracks using a position-based tracker. This stage is closely related to the tracking

pipeline of Chapter 3, but optimised to reduce run-time given the very large number of videos to process.

Stage 4. Active speaker verification. The goal of this stage is to determine the audio-video synchronisation between mouth motion and speech in a video in order to determine which (if any) visible face is the speaker. This is done by using 'SyncNet', a two-stream CNN described in Chapter 4 which estimates the correlation between the audio track and the mouth motion of the video. This method is able to reject the clips that contain dubbing or voice-over.

Stage 5. Face verification. Active speaker face tracks are then classified into whether they are of the POI or not using the VGG Face CNN. This classification network is based on the VGG-16 CNN (Simonyan and Zisserman, 2015) trained on the VGG Face dataset (which is a filtered collection of Google Image Search results for the POI name). Verification is done by directly using this classification score with a high threshold.

Discussion. In order to ensure that our system is extremely confident that a person is speaking (Stage 4), and that they have been correctly identified (Stage 5) without any manual interference, we set very conservative thresholds in order to minimise the number of false positives. Precision-recall curves for both tasks on their respective benchmark datasets (Chakravarty and Tuytelaars, 2016, Parkhi et al., 2015) are shown in Figure 9.1 right, and the values at the operating point are given in Table 9.2. Employing these thresholds ensures that although we discard a lot of the downloaded videos, we can be reasonably certain that the dataset has few labelling errors.
This ensures a completely automatic pipeline that can be scaled up to any number of speakers and utterances (if available) as required.

Task	Dataset	Precision	Recall
Active speaker verification	Columbia[a]	1.000	0.613
Face verification	VGG Face[b]	1.000	0.726

Table 9.2: Precision-recall values at the chosen operating points. [a] Chakravarty and Tuytelaars (2016); [b] Parkhi et al. (2015);

9.2 Models

Our aim is to move from techniques that require traditional hand-crafted features, to a CNN architecture that can choose the features required for the task of speaker recognition. This allows us to minimise the pre-processing of the audio data and hence avoid losing valuable information in the process.

Input features. All audio is first converted to single-channel, 16-bit streams at a 16kHz sampling rate for consistency. Spectrograms are then generated in a sliding window fashion using a hamming window of width 25ms, step 10ms and 1024-point FFT. This gives spectrograms of size 512 x 300 for 3 seconds of speech. Mean and variance normalisation is performed on every frequency bin of the spectrum. This normalisation is crucial, leading to an almost 10% increase in classification accuracy, as shown in Table 9.6. No other speech-specific preprocessing (e.g. silence removal, voice activity detection, or removal of unvoiced speech) is used. These short time magnitude spectrograms are then used as input to the CNN.

Architecture. Since speaker identification under a closed set can be treated as a multiple-class classification problem, we base our architecture on the VGG-M (Chatfield et al., 2014) CNN, known for good classification performance on image data, with modifications to adapt to the spectrogram input. The fully connected *fc6* layer of dimension 9×8 (support in both dimensions) is replaced by two layers – a fully connected layer of 9×1 (support in the frequency domain) and an average pool layer with support $1 \times n$, where n depends on the length of the input speech segment (for example for a 3 second segment, $n = 8$). This makes the network invariant to tem-

poral position but *not* frequency, and at the same time keeps the output dimensions the same as those of the original fully connected layer. This also reduces the number of parameters from 319M in VGG-M to 67M in our network, which helps avoid overfitting. The complete CNN architecture is specified in Table 9.3.

Identification. Since identification is treated as a simple classification task, the output of the last layer is fed into a 1,251-way softmax in order to produce a distribution over the 1,251 different speakers.

Verification. For verification, feature vectors can be obtained from the classification network using the 1024 dimension fc7 vectors, and a cosine distance can be used to compare vectors. However, it is better to learn an *embedding* by training a Siamese network with a contrastive loss (Chopra et al., 2005). This is better suited to the verification task as the network learns to optimize similarity directly, rather than indirectly via a classification loss. For the embedding network, the last fully connected layer (*fc8*) is modified so that the output size is 256 instead of the number of classes. We compare both methods in the experiments.

Testing. A traditional approach to handling variable length utterances at test time is to break them up into fixed length segments (e.g. 3 seconds) and average the results on each segment to give a final class prediction. Average pooling, however allows the network to accommodate variable length inputs at test time, as the entire test utterance can be evaluated at once by changing the size of the *apool6* layer. Not only is this more elegant, it also leads to an increase in classification accuracy, as shown in Table 9.6.

Implementation details and training. Our implementation is based on the deep learning toolbox MatConvNet (Vedaldi et al., 2014). The network is trained using batch normalisation (Ioffe and Szegedy, 2015) and all hyper-parameters (e.g. weight decay, learning rates) use the default values provided with the toolbox. To reduce

Layer	Support	Filt dim.	# filts.	Stride	Data size
conv1	7×7	1	96	2×2	254×148
mpool1	3×3	-	-	2×2	126×73
conv2	5×5	96	256	2×2	62×36
mpool2	3×3	-	-	2×2	30×17
conv3	3×3	256	256	1×1	30×17
conv4	3×3	256	256	1×1	30×17
conv5	3×3	256	256	1×1	30×17
mpool5	5×3	-	-	3×2	9×8
fc6	9×1	256	4096	1×1	1×8
apool6	1×n	-	-	1×1	1×1
fc7	1×1	4096	1024	1×1	1×1
fc8	1×1	1024	1251	1×1	1×1

Table 9.3: CNN architecture. The data size up to *fc6* is for a 3-second input, but the network is able to accept inputs of variable lengths.

overfitting, we augment the data by taking random 3-second crops in the time domain during training. Using a fixed input length is also more efficient. For verification, the network is first trained for classification (excluding the test POIs for the verification task, see Section 9.3), and then all filter weights are frozen except for the modified last layer and the Siamese network trained with contrastive loss. Choosing good pairs for training is very important in metric learning. We randomly select half of the negative examples, and the other half using Hard Negative Mining, where we only sample from the hardest 10% of all negatives.

9.3 Experiments

This section describes the experimental setup for both speaker identification and verification, and compares the performance of our devised CNN baseline to a number of traditional state of the art methods on `VoxCeleb`.

9.3.1 Experimental setup

Speaker identification. For identification, the training and the testing are performed on the same POIs. On each POI, we reserve the speech segments from one video for test. The test video contains at least 5 non-overlapping segments of speech. For identification, we report *top-1* and *top-5* accuracies. The statistics are given in Table 9.4.

Speaker verification. For verification, all POIs whose name starts with an 'E' are reserved for testing, since this gives a good balance of male and female speakers. These POIs are not used for training the network, and are only used at test time. The statistics are given in Table 9.5.

Two key performance metrics are used to evaluate system performance for the verification task. The metrics are similar to those used by existing datasets and challenges, such as NIST SRE12 (Greenberg et al., 2013) and SITW (McLaren et al., 2016). The primary metric is based on the cost function C_{det}

$$C_{det} = C_{miss} \times P_{miss} \times P_{tar} + C_{fa} \times P_{fa} \times (1 - P_{tar}) \tag{9.1}$$

where we assume a prior target probability P_{tar} of 0.01 and equal weights of 1.0 between misses C_{miss} and false alarms C_{fa}. The primary metric, C_{det}^{min}, is the minimum value of C_{det} for the range of thresholds. The alternative performance measure used here is the Equal Error Rate (EER) which is the rate at which both acceptance and rejection errors are equal. This measure is commonly used for identity verification systems.

Set	# POIs	# Vid. / POI	# Utterances
Dev	1,251	17.0	139,124
Test	1,251	1.0	6,255

Table 9.4: Development and test set statistics for identification.

Set	# POIs	# Vid. / POI	# Utterances
Dev	1,211	18.0	140,664
Test	40	17.4	4,715

Table 9.5: Development and test set statistics for verification.

9.3.2 Baselines

GMM-UBM. The GMM-UBM system uses MFCCs of dimension 13 as input. Cepstral mean and variance normalisation (CMVN) is applied on the features. Using the conventional GMM-UBM framework, a single speaker-independent universal background model (UBM) of 1024 mixture components is trained for 10 iterations from the training data.

I-vectors/PLDA. Gender independent i-vector extractors (Dehak et al., 2011) are trained on the `VoxCeleb` dataset to produce 400-dimensional i-vectors. Probabilistic LDA (PLDA) (Ioffe, 2006) is then used to reduce the dimension of the i-vectors to 200.

Inference. For identification, a one-vs-rest binary SVM classifier is trained for each speaker m ($m \in 1...K$). All feature inputs to the SVM are L2 normalised and a held out validation set is used to determine the C parameter (determines trade off between maximising the margin and penalising training errors). Classification during test time is done by choosing the speaker corresponding to the highest SVM score. The PLDA scoring function (Ioffe, 2006) is used for verification.

9.3.3 Results

Results are given in Tables 9.6 and 9.7. For both speaker recognition tasks, the CNN provides superior performance to the traditional state-of-the-art baselines.

For identification we achieve an 80.5% *top-1* classification accuracy over 1,251 different classes, almost 20% higher than traditional state of the art baselines. The CNN

architecture uses the average pooling layer for variable length test data. We also compare to two variants: 'CNN-fc-3s', this architecture has a fully connected fc6 layer, and divides the test data into 3s segments and averages the scores. As is evident there is a considerable drop in performance compared to the average pooling original – partly due to the increased number of parameters that must be learnt; 'CNN-fc-3s no var. norm.', this is the CNN-fc-3s architecture without the variance normalization pre-processing of the input (the input is still mean normalized). The difference in performance between the two shows the importance of variance normalization for this data.

For verification, the margin over the baselines is narrower, but still a significant improvement, with the embedding being the crucial step.

Accuracy	Top-1 (%)	Top-5 (%)
I-vectors + SVM	49.0	56.6
I-vectors + PLDA + SVM	60.8	75.6
CNN-fc-3s no var. norm.	63.5	80.3
CNN-fc-3s	72.4	87.4
CNN	**80.5**	**92.1**

Table 9.6: Results for identification on `VoxCeleb` (higher is better). The different CNN architectures are described in Section 9.2.

Metrics	C_{det}^{min}	EER (%)
GMM-UBM	0.80	15.0
I-vectors + PLDA	0.73	8.8
CNN-1024D	0.75	10.2
CNN-256D Embedding	**0.71**	**7.8**

Table 9.7: Results for verification on `VoxCeleb` (lower is better).

9.4 Conclusions

We provide a fully automated and scalable pipeline for audio data collection and use it to create a large-scale speaker identification dataset called `VoxCeleb`, with 1,251 speakers and over 100,000 utterances. In order to establish benchmark performance,

we develop a novel CNN architecture with the ability to deal with variable length audio inputs, which outperforms traditional state-of-the-art methods for both speaker identification and verification on this dataset.

Chapter 10

Conclusion

We conclude this thesis by summarising the achievements of the work, as well as providing suggestions for future research.

10.1 Achievements

In this thesis, we have made significant advances in five areas: (i) lip synchronisation and active speaker detection; (ii) lip reading and audio-visual speech recognition; (iii) visual speech synthesis; (iv) sign language recognition; (v) speaker recognition. We have also contributed several datasets for further research in the field.

In Chapter 3, we described the fully automated pipeline to generate a large-scale audio-visual corpus from BBC television videos, aligned with the corresponding transcript. This data has formed the basis of our research in Chapters 4, 5, 6 and 7, and has led to the release of two datasets, LRW and MV-LRS, in 2017. Researchers from other universities have already published a number of papers based on these datasets in the first months since their release (Stafylakis and Tzimiropoulos, 2017a,b, Torfi et al., 2017).

Chapter 4 introduced a two-stream CNN model that can learn a joint embedding of the sound and the mouth images from unlabelled data. We applied the trained network

to the tasks of audio-to-video synchronisation and active speaker detection, and in the latter we set a new state-of-the-art on a standard benchmark dataset (Chakravarty and Tuytelaars, 2016). The embedding features learnt here also provides powerful representations for related applications such as lip reading.

We then moved on to investigate deep learning models for the task of automated lip reading. In Chapter 5, we explored a number of ConvNet architectures for recognising hundreds of individual words from the large-scale dataset generated in Chapter 3. In Chapter 6, we built on the work of the previous chapter and proposed an attention-based model that is able to lip read natural sentences from in the wild videos. The performance of this model beats a professional lip reader on videos from BBC television, and we also demonstrate that if audio is available, then visual information helps to improve automated speech recognition (ASR) performance. This advance is of significant practical applicability, since it can help to boost the performance of ASR systems in noisy environments which is the main weakness of the existing systems.

In Chapter 7, we developed a CNN model and training strategy for generating talking faces from audio. This model is trained using self-supervised training strategy similar to that used in Chapter 4. To our knowledge, this is the first work on generation of talking faces to use an end-to-end image generation method, rather than frame reselection or mesh-based methods.

Chapter 8 presented ConvNet-based architectures and representations that can be used for recognising short temporal signals in time series, and for localising the signal. This is demonstrated for two problems with different levels of difficulty: (i) speech recognition where the supervision is weak, and (ii) sign language recognition where the supervision is weak and noisy. We have shown that the performance far exceeds previous work in this area in terms of supervisory requirements and generalization across signers.

Finally, Chapter 9 introduced a new dataset and deep learning models for speaker recognition. Previous works for speaker identification have been limited to constrained conditions, due to the synthetic nature of existing datasets. Here, we collected a new large-scale speaker recognition dataset collected from 'in the wild' videos using an automated pipeline, and proposed ConvNet architectures that outperforms traditional baselines for speaker recognition. The dataset collected in this chapter has been released to the public, and has been downloaded by over 500 users.

10.2 Suggestion for future research

Next, we discuss possibilities for future research.

Lip reading and audio-visual speech recognition. There are several interesting extensions to consider: first, the attention mechanism that provides the alignment in Chapter 6 is unconstrained, but in practice should always move monotonically from left to right. This monotonicity could be incorporated as a soft or hard constraint; second, the sequence-to-sequence model is used in batch mode – decoding a sentence given the entire corresponding lip sequence. Instead, a more on-line architecture could be used, where the decoder does not have access to the part of the lip sequence in the future; third, whilst the current system works well for the domains that are seen during training (*e.g.* news), it may not generalise well to other settings for which it might be difficult to obtain audio-visual training data. Here, external language models could be introduced to facilitate domain transfers to scenarios in which only the text is available for training; finally, it is possible that research of this type could discern important discriminative cues that are beneficial for teaching lip reading to the hard of hearing.

Visual speech synthesis. The most beneficial extension to Chapter 7 would be improvements to the image re-blending, such that the synthesised talking face can be

blended back to the target image to generate a natural looking video. It would be interesting to explore modifications to the Speech2Vid model such that the full image is generated. Recent architectures introduced for inpainting (Pathak et al., 2016) or image style transfer works (Gatys et al., 2016) could form the basis of this extension. Another clear extension is to add a quantitative performance measure of our models. This is not a straightforward task as there is no definitive performance measure of generative models for a specific domain. In natural image generation, Salimans et al. (2016) proposed a scoring system dependent on an image's softmax output when fed into a network trained on a classification task *e.g.* the inception network trained on ImageNet (Szegedy et al., 2015). One possible option is to have a lip-specific inception score using networks trained on a lip-specific task (Chapter 6).

Gesture and sign language recognition. The ultimate goal here is continuous translation of sign languages, however there are many challenges as described in Chapter 1.2. In particular, the poor signal-to-noise ratio in the training data makes the learning very difficult. A possible extension would be to apply attention models to these problems. Attention has shown promising performance in many fields (Bahdanau et al., 2015, Chan et al., 2015, Chung et al., 2017b, Sharma et al., 2015, Xu et al., 2015), by focussing on only the parts of the input that are relevant. There is also potential to use the technique described in Chapter 8 for *localising* the salient signals, which could help produce word-level or sentence-level alignment, hence improving the quality of supervision.

Speaker recognition. There is a number of large-scale datasets for facial recognition that are available to the public, such as VGGFace2 (Cao et al., 2017) and LFW (Huang et al., 2007), however there is no equivalent in speaker recognition apart from the dataset introduced in Chapter 9. By reusing the pipeline introduced here, it is possible to build a much larger dataset for training. It would also be useful to explore further CNN architectures for speaker recognition, such as dilated convolutions (Oord et al.,

2016) which has proven effective for speech related applications.

Recognising unconstrained gestures and multi-modal speech recognition. This thesis addressed the problem of recognising visual speech, and formalised gestures in the form of British Sign Language. It would be interesting to explore the possibility of recognising non-formal gestures such as facial expressions or body language, and combine this visual information with ASR and lip reading to develop a machine that can recognise and understand truly multimodal human communications.

Bibliography

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

M. A. R. Ahad, J. K. Tan, H. Kim, and S. Ishikawa. Motion history image: its variants and applications. *Machine Vision and Applications*, 23(2):255–281, 2012.

I. Anina, Z. Zhou, G. Zhao, and M. Pietikäinen. Ouluvs2: a multi-view audiovisual database for non-rigid mouth motion analysis. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–5. IEEE, 2015a.

I. Anina, Z. Zhou, G. Zhao, and M. Pietikäinen. Ouluvs2: a multi-view audiovisual database for non-rigid mouth motion analysis. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–5. IEEE, 2015b.

Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas. Lipnet: Sentence-level lipreading. *arXiv:1611.01599*, 2016.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *Proc. ICLR*, 2015.

P. Bell, M. J. F. Gales, T. Hain, J. Kilgour, P. Lanchantin, X. Liu, A. McParland, S. Renals, O. Saz, M. Wester, and P. C. Woodland. The MGB challenge: Evaluating multi-genre broadcast media recognition. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 687–693. IEEE, 2015.

S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179, 2015.

H. Bredin and G. Chollet. Audiovisual speech synchrony measure: application to biometrics. *EURASIP Journal on Applied Signal Processing*, 2007(1):179–179, 2007.

P. Buehler, M. Everingham, and A. Zisserman. Learning sign language by watching tv (using weakly aligned subtitles). In *Proc. CVPR*, 2009.

P. Buehler, M. Everingham, and A. Zisserman. Employing signed TV broadcasts for automated learning of British sign language. In *Workshop on the Representation and Processing of Sign Languages*, 2010.

N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden. Subunets: End-to-end hand shape and continuous sign language recognition. In *International Conference on Computer Vision*, pages 22–27, Oct 2017.

Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. *arXiv preprint arXiv:1710.08092*, 2017.

J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. The AMI meeting corpus. In *International Conference on Methods and Techniques in Behavioral Research*, volume 88, 2005.

P. Chakravarty and T. Tuytelaars. Cross-modal supervision for learning active speaker detection in video. 2016.

W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals. Listen, attend and spell. *arXiv preprint arXiv:1508.01211*, 2015.

J. Charles, D. Magee, and D. Hogg. Virtual immortality: Reanimating characters from tv shows. In *Computer Vision–ECCV 2016 Workshops*, pages 879–886. Springer, 2016.

K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *Proc. BMVC.*, 2014.

K. Chen and A. Salman. Learning speaker-specific characteristics with a deep neural architecture. *IEEE Transactions on Neural Networks*, 22(11):1744–1756, 2011.

Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. *arXiv preprint arXiv:1707.09405*, 2017.

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proc. CVPR*, volume 1, pages 539–546. IEEE, 2005.

J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio. End-to-end continuous speech recognition using attention-based recurrent nn: first results. *arXiv preprint arXiv:1412.1602*, 2014.

J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems*, pages 577–585, 2015.

J. S. Chung and A. Zisserman. Lip reading in the wild. In *Proc. ACCV*, 2016a.

J. S. Chung and A. Zisserman. Out of time: automated lip sync in the wild. In *Workshop on Multi-view Lip-reading, ACCV*, 2016b.

J. S. Chung and A. Zisserman. Signs in time: Encoding human motion as a temporal image. In *Workshop on Brave New Ideas for Motion Representations, ECCV*, 2016c.

J. S. Chung and A. Zisserman. Lip reading in profile. In *British Machine Vision Conference*, 2017.

J. S. Chung, A. Jamaludin, and A. Zisserman. You said that? In *British Machine Vision Conference*, 2017a.

J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman. Lip reading sentences in the wild. In *Proc. CVPR*, 2017b.

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

H. Cooper and R. Bowden. Learning signs from subtitles: A weakly supervised approach to sign language recognition. In *Proc. CVPR*, 2009.

H. Cooper, E. Ong, N. Pugeault, and . Bowden. Sign language recognition using sub-units. *The Journal of Machine Learning Research*, 13(1):2205–2231, 2012.

M. Cuturi. Fast global alignment kernels. In *ICML*, 2011.

A. Czyzewski, B. Kostek, P. Bratoszewski, J. Kotus, and M. Szykulski. An audio-visual corpus for multimodal automatic speech recognition. *Journal of Intelligent Information Systems*, pages 1–26, 2017.

S. B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(4):357–366, 1980.

N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009.

S. El Hihi and Y. Bengio. Hierarchical recurrent neural networks for long-term dependencies. In *Advances in neural information processing systems*, pages 493–499, 1996.

M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is... Buffy" – automatic naming of characters in TV video. In *Proc. BMVC.*, 2006.

M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, Jan 2015. DOI: 10.1007/s11263-014-0733-5.

B. Fan, L. Wang, F. K. Soong, and L. Xie. Photo-real talking head with deep bidirectional lstm. In *Proc. ICASSP*, pages 4884–4888. IEEE, 2015.

C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proc. CVPR*, 2016.

L. Feng and L. K. Hansen. A new database for speaker recognition. Technical report, Technical University of Denmark, DTU, 2005.

T. Field. The importance of touch. *Karger Gazette*, 67:10–12, 2004.

W. M. Fisher, G. R. Doddington, and K. M. Goudie-Marshall. The DARPA speech recognition research database: specifications and status. In *Proc. DARPA Workshop on speech recognition*, pages 93–99, 1986.

J. Forster, C. Schmidt, T. Hoyoux, O. Koller, U. Zelle, J. H. Piater, and H. Ney. Rwth-phoenix-weather: A large vocabulary sign language recognition and translation corpus. In *LREC*, pages 3785–3789, 2012.

Y. Fu, S. Yan, and T. S. Huang. Classification and feature extraction by simplexization. *Information Forensics and Security, IEEE Transactions on*, 3(1):91–100, 2008.

G. Galatas, G. Potamianos, and F. Makedon. Audio-visual speech recognition incorporating facial depth information captured by the kinect. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 2714–2717. IEEE, 2012.

J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. *NASA STI/Recon technical report*, 93, 1993.

P. Garrido, L. Valgaerts, H. Sarmadi, I. Steiner, K. Varanasi, P. Pérez, and C. Theobalt. Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track. In *Computer Graphics Forum*, volume 34, pages 193–204. Wiley Online Library, 2015.

L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proc. CVPR*, pages 2414–2423, 2016.

K. J. Geras, A.-r. Mohamed, R. Caruana, G. Urban, S. Wang, O. Aslan, M. Philipose, M. Richardson, and C. Sutton. Compressing lstms into cnns. *arXiv preprint arXiv:1511.06433*, 2015.

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

J. J. Godfrey, E. C. Holliman, and J. McDaniel. Switchboard: Telephone speech corpus for research and development. In *Proc. ICASSP*, volume 1, pages 517–520. IEEE, 1992.

S. Goldin-Meadow. The role of gesture in communication and thinking. *Trends in cognitive sciences*, 3(11):419–429, 1999.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.

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*, pages 369–376. ACM, 2006a.

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*, pages 369–376. ACM, 2006b.

A. Graves, N. Jaitly, and A.-r. Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 273–278. IEEE, 2013.

C. S. Greenberg, V. M. Stanford, A. F. Martin, M. Yadagiri, G. R. Doddington, J. J. Godfrey, and J. Hernandez-Cordero. The 2012 NIST speaker recognition evaluation. In *INTERSPEECH*, pages 1971–1975, 2013.

J. H. Hansen, R. Sarikaya, U. H. Yapanel, and B. L. Pellom. Robust speech recognition in noise: an evaluation using the spine corpus. In *INTERSPEECH*, pages 905–908, 2001.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

G. Heigold, I. Moreno, S. Bengio, and N. Shazeer. End-to-end text-dependent speaker verification. In *Proc. ICASSP*, pages 5115–5119. IEEE, 2016.

J. Hennebert, H. Melin, D. Petrovska, and D. Genoud. POLYCOST: a telephone-speech database for speaker recognition. *Speech communication*, 31(2):265–270, 2000.

H. Hermansky. Perceptual linear predictive (plp) analysis of speech. *the Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.

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

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997.

J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017.

G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. 2007.

D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160:106–154, 1962.

S. Ioffe. Probabilistic linear discriminant analysis. In *Proc. ECCV*, pages 531–542. Springer, 2006.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. Learning visual groups from co-occurrences in space and time. *arXiv preprint arXiv:1511.06811*, 2015.

P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*, 2017.

A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. The ICSI meeting corpus. In *Proc. ICASSP*, volume 1. IEEE, 2003.

S. Ji, W. Xu, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. *IEEE PAMI*, 35(1):221–231, 2013.

Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. `http://caffe.berkeleyvision.org/`, 2013.

A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proc. CVPR*, pages 1725–1732, 2014.

V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014a.

V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proc. CVPR*, pages 1867–1874, 2014b.

P. Kenny. Joint factor analysis of speaker and session variability: Theory and algorithms. *CRIM, Montreal, CRIM-06/08-13*, 2005.

J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *Proc. CVPR*, June 2016.

D. E. King. Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10:1755–1758, 2009.

O. Koller, H. Ney, and R. Bowden. Deep learning of mouth shapes for sign language. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 85–91, 2015.

O. Koller, H. Ney, and R. Bowden. Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. In *Proc. CVPR*, pages 3793–3802, 2016a.

O. Koller, S. Zargaran, H. Ney, and R. Bowden. Deep sign: Hybrid cnn-hmm for continuous sign language recognition. In *British Machine Vision Conference*, York, UK, Sep 2016b.

G. Kondrak. A new algorithm for the alignment of phonetic sequences. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 288–295. Association for Computational Linguistics, 2000.

B. E. Koster, R. D. Rodman, and D. Bitzer. Automated lip-sync: Direct translation of speech-sound to mouth-shape. In *Signals, Systems and Computers, 1994. 1994 Conference Record of the Twenty-Eighth Asilomar Conference on*, volume 1, pages 583–586. IEEE, 1994.

R. Krishnan and S. Sarkar. Similarity measure between two gestures using triplets. In *CVPR Workshops*, 2013.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012.

Y. Lan, R. Harvey, B. Theobald, E.-J. Ong, and R. Bowden. Comparing visual features for lipreading. In *International Conference on Auditory-Visual Speech Processing 2009*, pages 102–106, 2009.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

B. Lee, M. Hasegawa-Johnson, C. Goudeseune, S. Kamdar, S. Borys, M. Liu, and T. S. Huang. Avicar: audio-visual speech corpus in a car environment. In *INTERSPEECH*. Citeseer, 2004.

D. Lee, J. Lee, and K.-E. Kim. Multi-view automatic lip-reading using neural network. In *ACCV 2016 Workshop on Multi-view Lip-reading Challenges*. Asian Conference on Computer Vision (ACCV), 2016.

J. Lewis. Automated lip-sync: Background and techniques. *The Journal of Visualization and Computer Animation*, 2(4):118–122, 1991.

R. Lienhart. Reliable transition detection in videos: A survey and practitioner's guide. *International Journal of Image and Graphics*, Aug 2001.

W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *Proc. ECCV*, pages 21–37. Springer, 2016.

Y. Lukic, C. Vogt, O. Dürr, and T. Stadelmann. Speaker identification and clustering using convolutional neural networks. In *IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2016.

E. Marcheret, G. Potamianos, J. Vopicka, and V. Goel. Detecting audio-visual synchrony using deep neural networks. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

M. Marschark and P. E. Spencer. *The Oxford handbook of deaf studies, language, and education*, volume 2. Oxford University Press, 2010.

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

D. F. McAllister, R. D. Rodman, D. L. Bitzer, and A. S. Freeman. Lip synchronization of speech. In *Audio-Visual Speech Processing: Computational & Cognitive Science Approaches*, 1997.

C. McCool and S. Marcel. Mobio database for the ICPR 2010 face and speech competition. Technical report, IDIAP Research Institute, 2009.

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

M. McLaren, L. Ferrer, D. Castan, and A. Lawson. The speakers in the wild (SITW) speaker recognition database. *INTERSPEECH*, 2016.

J. B. Millar, J. P. Vonwiller, J. M. Harrington, and P. J. Dermody. The Australian national database of spoken language. In *Proc. ICASSP*, volume 1, pages I–97. IEEE, 1994.

M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

S. Morishima, S. Ogata, K. Murai, and S. Nakamura. Audio-visual speech translation with automatic lip syncqronization and face tracking based on 3-d head model. In *Proc. ICASSP*, volume 2, pages II–2117. IEEE, 2002.

D. Morris. *The naked ape: A zoologist's study of the human animal*. Random House, 1994.

G. Morrison, C. Zhang, E. Enzinger, F. Ochoa, D. Bleach, M. Johnson, B. Folkes, S. De Souza, N. Cummins, and D. Chow. Forensic database of voice recordings of 500+ Australian English speakers. *URL: http://databases.forensic-voice-comparison.net*, 2015.

Y. Mroueh, E. Marcheret, and V. Goel. Deep multimodal learning for audio-visual speech recognition. In *Proc. ICASSP*, pages 2130–2134. IEEE, 2015.

A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017.

J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proc. CVPR*, pages 4694–4702, 2015.

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-11)*, pages 689–696, 2011.

K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata. Lipreading using convolutional neural network. In *INTERSPEECH*, pages 1149–1153, 2014.

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

A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free? weakly-supervised learning with convolutional neural networks. In *Proc. CVPR*, 2015.

D. O'Shaughnessy. Automatic speech recognition: History, methods and challenges. *Pattern Recognition*, 41(10):2965–2979, 2008.

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

G. Papandreou, I. Kokkinos, and P. Savalle. Untangling local and global deformations in deep convolutional networks for image classification and sliding window

detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *Proc. BMVC.*, 2015.

R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proc. CVPR*, pages 2536–2544, 2016.

E. K. Patterson, S. Gurbuz, Z. Tufekci, and J. N. Gowdy. Cuave: A new audio-visual database for multimodal human-computer interface research. In *Proc. ICASSP*, volume 2, pages II–2017. IEEE, 2002.

W. Paulus. Lips don't lie. `https://wolfpaulus.com/technology/lipsynchronization`, 2013. Accessed: 2017-12-06.

Y. Pei, T.-K. Kim, and H. Zha. Unsupervised random forest manifold alignment for lipreading. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 129–136, 2013.

A. Peregudov, K. Glasman, and A. Logunov. Relative timing of sound and vision: evaluation and correction. In *Consumer Electronics, 2005.(ISCE 2005). Proceedings of the Ninth International Symposium on*, pages 198–202. IEEE, 2005.

P. Perez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics*, 22(3):313–318, 2003.

S. Petridis and M. Pantic. Deep complementary bottleneck features for visual speech recognition. In *Proc. ICASSP*, pages 2304–2308, 2016.

T. Pfister, J. Charles, and A. Zisserman. Large-scale learning of sign language by watching TV (using co-occurrences). In *Proc. BMVC.*, 2013.

T. Pfister, J. Charles, and A. Zisserman. Domain-adaptive discriminative one-shot learning of gestures. In *Proc. ECCV*, 2014a.

T. Pfister, K. Simonyan, J. Charles, and A. Zisserman. Deep convolutional neural networks for efficient pose estimation in gesture videos. In *Proc. ACCV*, 2014b.

T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *Proc. ICCV*, 2015.

D. A. Reynolds and R. C. Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE transactions on speech and audio processing*, 3(1):72–83, 1995.

D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000.

F. Ronchetti, F. Quiroga, C. Estrebou, L. Lanzarini, and A. Rosete. Lsa64: A dataset of argentinian sign language. *XX II Congreso Argentino de Ciencias de la Computacin (CACIC)*, 2016.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

E. A. Rúa, H. Bredin, C. G. Mateo, G. Chollet, and D. G. Jiménez. Audio-visual speech asynchrony detection using co-inertia analysis and coupled hidden markov models. *Pattern Analysis and Applications*, 12(3):271–284, 2009.

S. Rubin, F. Berthouzoz, G. J. Mysore, W. Li, and M. Agrawala. Content-based tools for editing audio stories. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 113–122. ACM, 2013.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, S. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and F. Li. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

T. Saitoh, Z. Zhou, G. Zhao, and M. Pietikäinen. Concatenated frame image based cnn for visual speech recognition. In *Asian Conference on Computer Vision*, pages 277–289. Springer, 2016.

T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016. URL `http://arxiv.org/abs/1606.03498`.

M. E. Sargin, Y. Yemez, E. Erzin, and A. M. Tekalp. Audiovisual synchronization and fusion using canonical correlation analysis. *IEEE Transactions on Multimedia*, 9(7):1396–1403, 2007.

F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. CVPR*, 2015.

R. Sennrich, B. Haddow, and A. Birch. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*, 2015.

S. Sharma, R. Kiros, and R. Salakhutdinov. Action recognition using visual attention. *arXiv preprint arXiv:1511.04119*, 2015.

K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*, 2014.

T. Stafylakis and G. Tzimiropoulos. Combining residual networks with lstms for lipreading. *arXiv preprint arXiv:1703.04105*, 2017a.

T. Stafylakis and G. Tzimiropoulos. Deep word embeddings for visual speech recognition. *arXiv preprint arXiv:1710.11201*, 2017b.

L. L. Stoll. Finding difficult speakers in automatic speaker recognition. *Technical Report No. UCB/EECS-2011-152*, 2011.

I. Sutskever. Training recurrent neural networks. *University of Toronto, Toronto, Ont., Canada*, 2013.

I. Sutskever, O. Vinyals, and Q. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):95, 2017.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, 2015.

S. Tamura, H. Ninomiya, N. Kitaoka, S. Osuga, Y. Iribe, K. Takeda, and S. Hayamizu. Audio-visual speech recognition using deep bottleneck features and high-performance lipreading. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 575–582. IEEE, 2015.

S. Taylor, T. Kim, Y. Yue, M. Mahler, J. Krahe, A. G. Rodriguez, J. Hodgins, and I. Matthews. A deep learning approach for generalized speech animation. *ACM Transactions on Graphics (TOG)*, 36(4):93, 2017.

A. Thangali, J. P. Nash, S. Sclaroff, and C. Neidle. Exploiting phonological constraints for handshape inference in asl video. In *Proc. CVPR*, pages 521–528. IEEE, 2011.

C. Tomasi and T. Kanade. Selecting and tracking features for image sequence analysis. *Robotics and Automation*, 1992.

A. Torfi, S. M. Iranmanesh, N. Nasrabadi, and J. Dawson. 3d convolutional neural networks for cross audio-visual matching recognition. *IEEE Access*, 2017.

D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. 2015.

N. Ukai, T. Seko, S. Tamura, and S. Hayamizu. Gif-lr: Ga-based informative feature for lipreading. In *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pages 1–4. IEEE, 2012.

A. van den Oord, N. Kalchbrenner, L. Espeholt, k. kavukcuoglu, O. Vinyals, and A. Graves. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016a.

A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016b.

D. van der Vloed, J. Bouten, and D. A. van Leeuwen. NFI-FRITS: a forensic speaker recognition database and some first experiments. In *The Speaker and Language Recognition Workshop*, 2014.

A. Vedaldi and K. Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proc. ACMM*, 2015.

A. Vedaldi, S. Mahendran, S. Tsogkas, S. Maji, R. Girshick, J. Kannala, E. Rahtu, I. Kokkinos, M. B. Blaschko, D. Weiss, B. Taskar, K. Simonyan, N. Saphra, and S. Mohamed. Understanding objects in detail with fine-grained attributes. In *Proc. CVPR*, 2014.

T. Vintsyuk. Speech discrimination by dynamic programming. *Cybernetics and Systems Analysis*, 4(1):52–57, 1968.

U. Von Agris, J. Zieren, U. Canzler, B. Bauer, and K.-F. Kraiss. Recent developments in visual sign language recognition. *Universal Access in the Information Society*, 6 (4):323–362, 2008.

J. Wan, Q. Ruan, W. Li, and S. Deng. One-shot learning gesture recognition from RGB-D data using bag of features. *JMLR*, 14(1):2549–2582, 2013.

J. Wan, Q. Ruan, W. Li, G. An, and R. Zhao. 3d smosift: three-dimensional sparse motion scale invariant feature transform for activity recognition from rgb-d videos. *Journal of Electronic Imaging*, 23(2):023017–023017, 2014.

M. Wand, J. Koutník, and J. Schmidhuber. Lipreading with long short-term memory. In *Proc. ICASSP*, pages 6115–6119. IEEE, 2016.

X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2015.

R. Woo, A. Park, and T. J. Hazen. The MIT Mobile Device Speaker Verification Corpus: Data collection and preliminary experiments. *The Speaker and Language Recognition Workshop*, 2006.

P. C. Woodland, C. Leggetter, J. Odell, V. Valtchev, and S. J. Young. The 1994 htk large vocabulary speech recognition system. In *Proc. ICASSP*, volume 1, pages 73–76. IEEE, 1995.

K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015.

U. H. Yapanel, X. Zhang, and J. H. Hansen. High performance digit recognition in real car environments. In *INTERSPEECH*, 2002.

S. H. Yella, A. Stolcke, and M. Slaney. Artificial neural network features for speaker diarization. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 402–406. IEEE, 2014.

D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

J. Yuan and M. Liberman. Speaker identification on the scotus corpus. *Journal of the Acoustical Society of America*, 123(5):3878, 2008.

Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti. American sign language recognition with the kinect. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 279–286. ACM, 2011.

R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *Proc. ECCV*, pages 649–666. Springer, 2016.

G. Zhao, M. Barnard, and M. Pietikäinen. Lipreading with local spatiotemporal descriptors. *Multimedia, IEEE Transactions on*, 11(7):1254–1265, 2009.

Z. Zhou, X. Hong, G. Zhao, and M. Pietikäinen. A compact representation of visual speech data using latent variables. *IEEE PAMI*, 36(1), 2014a.

Z. Zhou, G. Zhao, X. Hong, and M. Pietikäinen. A review of recent advances in visual speech decoding. *Image and vision computing*, 32(9):590–605, 2014b.

G. Zoric and I. S. Pandzic. A real-time lip sync system using a genetic algorithm for automatic neural network configuration. In *2005 IEEE International Conference on Multimedia and Expo*, pages 1366–1369. IEEE, 2005.