Out of time: automated lip sync in the wild

Joon Son Chung and Andrew Zisserman

Visual Geometry Group, Department of Engineering Science, University of Oxford

Abstract. The goal of this work is to determine the audio-video synchronisation between mouth motion and speech in a video.
We propose a two-stream ConvNet architecture that enables a joint embedding between 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 apply the network to two further tasks: active speaker detection and lip reading. On both tasks we set a new state-of-the-art on standard benchmark datasets.

1 Introduction

Audio to video synchronisation (or lack of it) is a problem in TV broadcasting for the producer and the viewer. 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 -125ms (the audio lags the video) to +45ms (the audio leads the video) [1].

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 work 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 TV broadcast, without labelled data. 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. We demonstrate that the method can be applied to three different tasks: (i) determining the *lip-sync error* in videos; (ii) detecting the speaker in a scene with multiple faces; and (iii) lip reading. The experimental performance on all of these tasks is extremely strong. In speaker detection and lip reading, our results exceed the state-of-the-art on public datasets, Columbia [4] and OuluVS2 [2].

1.1 Related works

There is a large body of work on the audio to video synchronisation problem. 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 on computer vision methods that only rely on the audio-visual data.

A number of papers have used *phoneme* recognition as a proxy task for solving the lip-sync problem. In Lewis *et al.* [15], 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.* [19] classifies the face parameters into *visemes*, and uses the *viseme* to *phoneme* mapping to obtain the synchronisation. Although [13] and [18] 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 [13] and [18] correlate face parameters such as jaw position to the FFT of the sound signal. Zoric and Pandzic [29] 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. We do not make an intermediate classification of sounds and mouth shapes into vowels or *phonemes*.

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) [3, 22] or co-inertia analysis (CoIA) [20] of audio and visual features (*e.g.* geometric parameters or 2D DCT features). The most related work to ours is that of Marcharet *et al.* [17] 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.), whereas we learn the visual features directly.

Of relevance to the architectures developed in this paper are Siamese networks [6], in which similarity metrics are learnt for face classification without explicit class labels. [23, 27] are also relevant in that they simultaneously train multi-stream networks in which the inputs are of different domains.

2 Representations and architecture

This section describes the representations and network architectures for SyncNet. The network ingests 0.2-second clips of both audio and video inputs. In the dataset (Section 3), 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.

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

2.1 Audio stream

The input audio data is 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 100Hz, giving 20 time steps for a 0.2-second input signal.

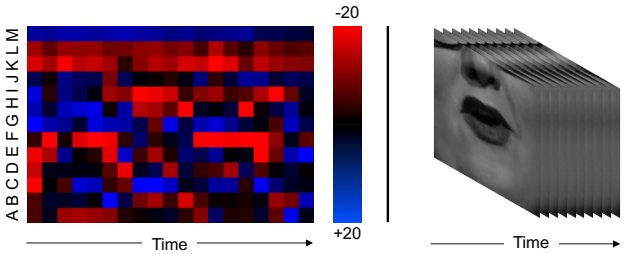

Fig. 1. Input representations. **Left:** temporal representations as heatmaps for audio. The 13 rows (A to M) in the audio image encode each of the 13 MFCC features representing powers at different frequency bins. **Right:** Grayscale images of the mouth area.

Representation. The audio is encoded as a heatmap image representing MFCC values for each time step and each mel frequency band (see Figure 1). The top and bottom three rows of the image are reflected to reduce boundary effects. Previous work [9] 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 2) is based on VGG-M [5], 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).

2.2 Visual stream

Representation. The input format to the visual network is a sequence of mouth regions as grayscale images, as shown in Figure 1. The input dimensions are 111×111×5 (W×H×T) for 5 frames, which corresponds to 0.2-seconds at the 25Hz frame rate.

Architecture. We base our architecture on that of [7], which is designed for the task of visual speech recognition. In particular, the architecture is based on the Early Fusion model, which is compact and fast to train. The *conv1* filter has been modified to ingest the 5-channel input.

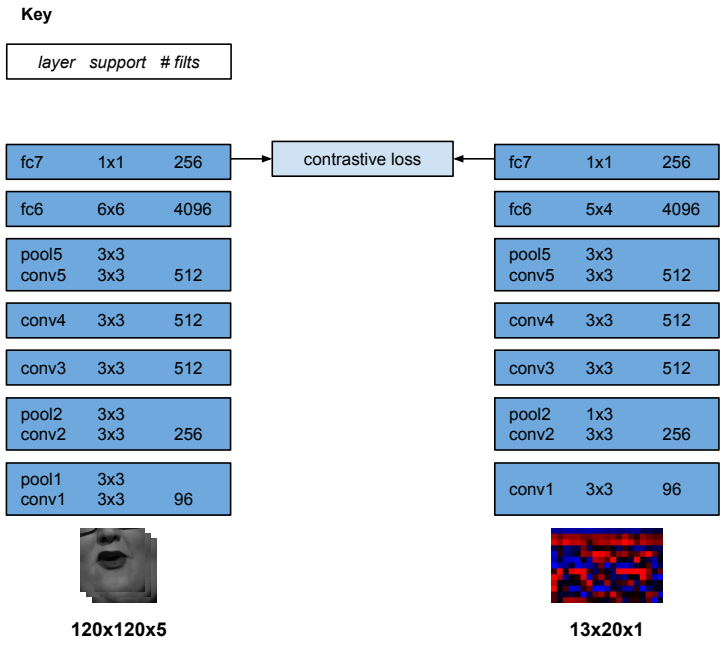

Fig. 2. SyncNet architecture. Both streams are trained simultaenously.

2.3 Loss function

The training objective is that the output of the audio and the video networks are similar for *genuine* pairs, and different for *false* pairs. Specifically, the Euclidean distance between the network outputs is minimised or maximised. We propose to use the contrastive loss (Equation 1), originally proposed for training Siamese networks [6]. 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.

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

$$d_n = ||v_n - a_n||_2 \tag{2}$$

An alternative to this would be 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 using this method.

2.4 Training

The training procedure is an adaptation of the usual procedure for a single-stream ConvNet [14, 24] and inspired by [6, 23]. 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 [14]. For the audio, the volume 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 *false* examples only, we take random crops in time. For the video, we apply the standard augmentation methods used on the ImageNet classification task by [14, 24] (*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 [26] and trained on a NVIDIA Titan X GPU with 12GB memory. The network is trained with batch normalisation [10]. 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 is sooner.

3 Dataset

Fig. 3. Still images of BBC News videos.

In this section, we describe the pipeline for automatically generating a large-scale audio-visual dataset for training the lip synchronisation system. Using the methods described, we collect several hundred hours of speech from BBC videos, covering hundreds of speakers. We start from BBC News programs recorded between 2013 and 2016 (Figure 3), given that a large number of different people appear in the news, in contrast to dramas with a fixed cast. The training, validation and test sets are divided in time, and the dates of videos corresponding to each set are shown in Table 1.

The processing pipeline is summarised in Figure 4. The visual part of the pipeline is based on the methods used by Chung and Zisserman [7], and we give a brief sketch of the method here. First, shot boundaries are determined

Set	Dates	# pairs	# hours
Train	01/07/2013 - 31/08/2015	3,707K	606
Val	01/09/2015 - 31/12/2015	316K	42
Test	01/01/2016 - 31/05/2016	350K	47

Table 1. Dataset statistics: recording dates, and number of genuine (positive) and false lip-sync audio-video training samples, number of hours of facetrack.

by comparing color histograms across consecutive frames [16]. The HOG-based face detection method of [12] is then performed on every frame, and the face detections are grouped across frames using a KLT tracker [25]. We discard any clips in which more than one face appears in the video, as the speaker is not known in this scenario.

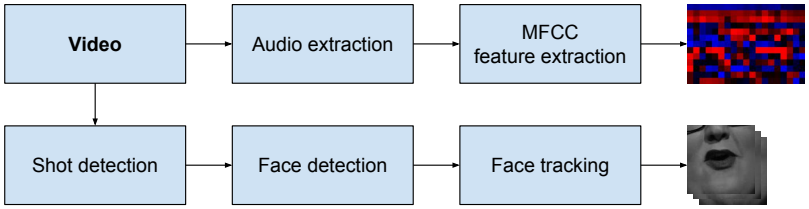

Fig. 4. Pipeline to generate the audio-visual dataset.

The audio part of the pipeline is straightforward. The Mel-frequency cepstral coefficient (MFCC) [8] features are used to describe the audio, which are commonly used in speech recognition systems. No other pre-processing is performed on the audio.

3.1 Compiling the training data

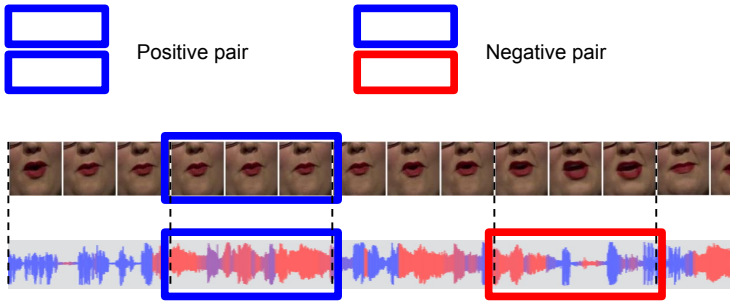

Fig. 5. The process of obtaining *genuine* and *false* audio-video pairs.

Genuine 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 *false* audio-video pairs. This is illustrated in Figure 5. We take the audio from the same clip, so that the network learns to recognise the alignment, rather than the speaker.

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.

A network is initially trained on this noisy data, and the trained network is used to discard the false positives in the training set by rejecting positive pairs with distance over a threshold. A network is then re-trained on this new data.

Discussion. 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.

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 of the features is taken. This is the same distance function that is used at training time. The histogram (Figure 6) shows the distribution of the metric.

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 $\pm$ 1 second range. The correct offset is when this distance is at a minimum. However as Table 2 suggests, not all samples in a clip are discriminative (for example, there may be samples in which nothing is being said at that particular time), therefore multiple samples are taken for each clip, and then averaged. Typical response plots are shown in Figure 8.

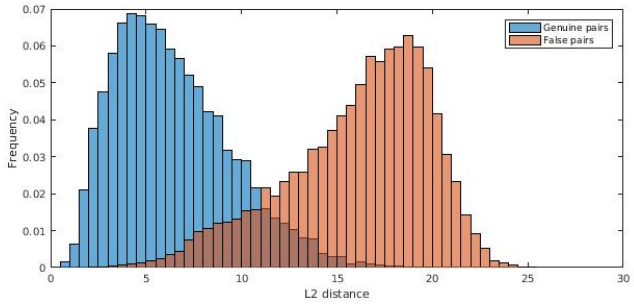

Fig. 6. The distribution of Euclidean distances for *genuine* and *false* audio-video pairs, using a single 0.2-second sample. Note that this is on the noisy validation data that may include clips of non-speakers or dubbed videos.

Evaluation. The precise time offset between the audio and the video is not known. Therefore, the evaluation is done manually, where the synchronisation is considered successful if the lip-sync error is not detectable to a human. We take a random sample of several hundred clips from the part of the dataset that has been reserved for testing, as described in Section 3. The success rates are reported in Table 2.

Method	**Accuracy**
Single sample (0.2s)	81%
Averaged over a clip	>99%

Table 2. Accuracy to within human-detectable range.

Experiments were also performed on a sample of foreign language videos (Figure 7), to show that our method works across different languages. Qualitative results are extremely good, and will be available from our research page.

Fig. 7. Images of foreign language videos that were used for testing.

Performance. The data preparation pipeline and the network runs significantly faster than real-time on a mid-range laptop (Apple MacBook Pro with NVIDIA GeForce GT 750M graphics), with the exception of the face detection step (external application), which runs at around $\times 0.3$ real-time.

4.2 Application: active speaker detection

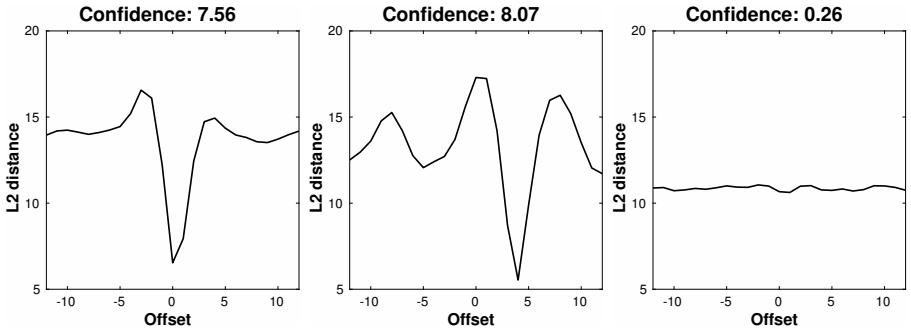

Fig. 8. Mean distance between the audio and the video features for different offset values, averaged over a clip. The actual offset lies at the trough. The three example clips shown here are for different scenarios. **Left:** synchronised AV data; **Middle:** the audio leads the video; **Right:** the audio and the video are uncorrelated.

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 8). 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 9) and the evaluation protocol of Chakravarty *et al.* [4]. The objective is to determine who the speaker is in a multi-subject scene.

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 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.

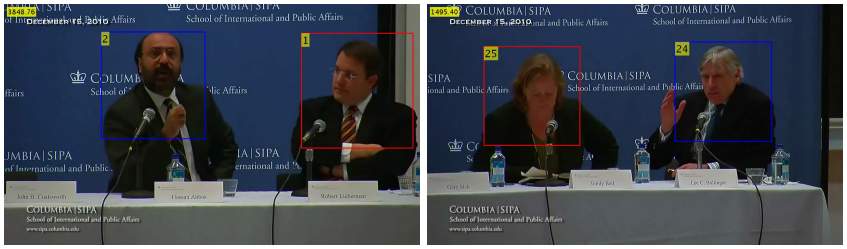

Fig. 9. Still images from the Columbia dataset [4].

Method	[4]		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 3. F_1-scores on the Columbia speaker detection dataset. The results of [4] have been digitised from Figure 3b of their paper, and are accurate to around ±0.5%.

4.3 Application: lip reading

Training a deep network for any task requires large quantities of data, but for problems such as lip reading, large-scale annotated data can be prohibitively expensive to collect. However, unlabelled spoken videos are copious and easy to obtain.

A useful by-product of the synchronisation network is that it enables very strong mouth descriptors to be learnt without any labelled data. We use this result to set the new state-of-the-art on the OuluVS2 [2] dataset. This consists of 52 subjects uttering the same 10 phrases (*e.g.* 'thank you', 'hello', etc.) or 10 predetermined digit sequences. It is assessed on a speaker-independent experiment, where 12 specified subjects are reserved for testing. Only the video stream is used for training and testing, *i.e.* this is a 'lip reading' experiment rather than one of audio-visual speech recognition.

Experimental setup. A simple uni-directional LSTM classifier with one layer and 250 hidden units is used for this experiment. The setup is shown in Figure 10. The LSTM network ingests the visual features (fc_7 activations from the ConvNet) of the 5-frame sliding window, moving 1-frame at a time, and returns the classification result at the end of the sequence.

Training details. Our implementation of the recurrent network is based on the Caffe [11] toolbox. The network is trained with stocastic gradient descent, with a learning rate of 10^{-3}. The gradients are back-propagated for the full length

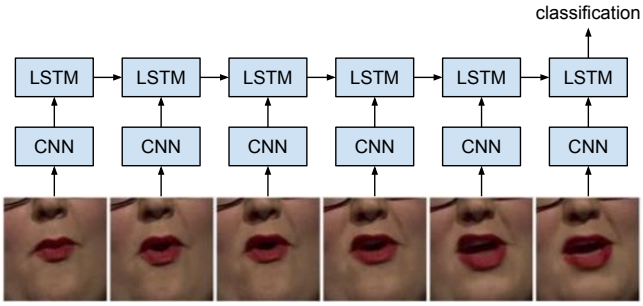

Fig. 10. Network configuration for the lip reading experiment. ConvNet weights are not updated at the time of LSTM training.

of the clip. Softmax log loss is used, which is typical for a n-way classification problem. Here $n = 10$ for the 10 phrases or digit sequences. The loss is computed only at the final timestep.

Method	**Short phrases**	**Fixed digits**
Zhou *et al.* [28]	73.5%	-
Chung and Zisserman [7]	93.2%	-
VGG-M + LSTM	31.9%	25.4%
SyncNet + LSTM	**94.1%**	**92.8%**

Table 4. Test set classification accuracy on OuluVS2, frontal view.

Evaluation. We compare our results to the previous state-of-the-art on this dataset; and also the same LSTM setup, but instead with a VGG-M [5] convolutional network pre-trained on ImageNet [21]. We report the results in Table 4. In particular, it is notable that our result beats that of [7], which is obtained using a network that has been pre-trained on a very large *labelled* dataset.

5 Conclusion

We have demonstrated that a two-stream ConvNet can be trained to synchronise audio to mouth motion, from natural videos of speech that are easy to obtain. A useful application of this method is in media players, where the lip-sync error can be corrected on a local machine at run-time. Furthermore, the approach can be extended to any problem where it is useful to learn a joint embedding of correlated data in different domains.

We have also shown that the trained network works effectively for the tasks of speaker detection in video, and lip reading.

Download. The trained model is available for download from:
`http://www.robots.ox.ac.uk/~vgg/software/lipsync`.

Acknowledgements. We are very grateful to Andrew Senior for suggesting this problem; to Rob Cooper and Matt Haynes at BBC Research for help in obtaining the lip synchronisation dataset; and to Punarjay Chakravarty and Tinne Tuytelaars for supplying the Columbia dataset. Funding for this research is provided by the EPSRC Programme Grant Seebibyte EP/M013774/1.

Bibliography

[1] Bt.1359 : Relative timing of sound and vision for broadcasting. ITU (1998)

[2] Anina, I., Zhou, Z., Zhao, G., Pietikäinen, M.: 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. vol. 1, pp. 1–5. IEEE (2015)

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

[4] Chakravarty, P., Tuytelaars, T.: Cross-modal supervision for learning active speaker detection in video. arXiv preprint arXiv:1603.08907 (2016)

[5] Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. In: Proc. BMVC. (2014)

[6] Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: Proc. CVPR. vol. 1, pp. 539–546. IEEE (2005)

[7] Chung, J.S., Zisserman, A.: Lip reading in the wild. In: Proc. ACCV (2016)

[8] Davis, S.B., Mermelstein, P.: 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)

[9] Geras, K.J., Mohamed, A.r., Caruana, R., Urban, G., Wang, S., Aslan, O., Philipose, M., Richardson, M., Sutton, C.: Compressing lstms into cnns. arXiv preprint arXiv:1511.06433 (2015)

[10] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)

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

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

[13] Koster, B.E., Rodman, R.D., Bitzer, D.: 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. vol. 1, pp. 583–586. IEEE (1994)

[14] Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS. pp. 1106–1114 (2012)

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

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

[17] Marcheret, E., Potamianos, G., Vopicka, J., Goel, V.: Detecting audio-visual synchrony using deep neural networks. In: Sixteenth Annual Conference of the International Speech Communication Association (2015)

[18] McAllister, D.F., Rodman, R.D., Bitzer, D.L., Freeman, A.S.: Lip synchronization of speech. In: Audio-Visual Speech Processing: Computational & Cognitive Science Approaches (1997)
[19] Morishima, S., Ogata, S., Murai, K., Nakamura, S.: Audio-visual speech translation with automatic lip syncqronization and face tracking based on 3-d head model. In: Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on. vol. 2, pp. II–2117. IEEE (2002)
[20] Rúa, E.A., Bredin, H., Mateo, C.G., Chollet, G., Jiménez, D.G.: Audio-visual speech asynchrony detection using co-inertia analysis and coupled hidden markov models. Pattern Analysis and Applications 12(3), 271–284 (2009)
[21] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, S., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., Li, F.: Imagenet large scale visual recognition challenge. IJCV (2015)
[22] Sargin, M.E., Yemez, Y., Erzin, E., Tekalp, A.M.: Audiovisual synchronization and fusion using canonical correlation analysis. IEEE Transactions on Multimedia 9(7), 1396–1403 (2007)
[23] Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS (2014)
[24] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
[25] Tomasi, C., Kanade, T.: Selecting and tracking features for image sequence analysis. Robotics and Automation (1992)
[26] Vedaldi, A., Lenc, K.: Matconvnet – convolutional neural networks for matlab. CoRR abs/1412.4564 (2014)
[27] Zhong, Y., Arandjelović, R., Zisserman, A.: Faces in places: Compound query retrieval. In: British Machine Vision Conference (2016)
[28] Zhou, Z., Hong, X., Zhao, G., Pietikäinen, M.: A compact representation of visual speech data using latent variables. IEEE transactions on pattern analysis and machine intelligence 36(1), 1–1 (2014)
[29] Zoric, G., Pandzic, I.S.: A real-time lip sync system using a genetic algorithm for automatic neural network configuration. In: 2005 IEEE International Conference on Multimedia and Expo. pp. 1366–1369. IEEE (2005)