Synchronising audio and ultrasound by learning cross-modal embeddings

Citation for published version:
Eshky, A, Ribeiro, M, Richmond, K & Renals, S 2019, Synchronising audio and ultrasound by learning cross-modal embeddings. in INTERSPEECH 2019: Proceedings of the 20th Annual Conference of the International Speech Communication Association (ISCA). International Speech Communication Association, Graz, Austria, pp. 4100-4104, Interspeech 2019, Graz, Austria, 15/09/19. https://doi.org/10.21437/Interspeech.2019-1804

Digital Object Identifier (DOI):
10.21437/Interspeech.2019-1804

Published In:
INTERSPEECH 2019: Proceedings of the 20th Annual Conference of the International Speech Communication Association (ISCA)

Synchronising audio and ultrasound by learning cross-modal embeddings

Aciel Eshky, Manuel Sam Ribeiro, Korin Richmond, Steve Renals

CSTR, School of Informatics, University of Edinburgh, UK

{aeshky, sam.ribeiro, korin.richmond, s.renals}@ed.ac.uk

Abstract

Audiovisual synchronisation is the task of determining the time offset between speech audio and a video recording of the articulators. In child speech therapy, audio and ultrasound videos of the tongue are captured using instruments which rely on hardware to synchronise the two modalities at recording time. Hardware synchronisation can fail in practice, and no mechanism exists to synchronise the signals post hoc. To address this problem, we employ a two-stream neural network which exploits the correlation between the two modalities to find the offset. We train our model on recordings from 69 speakers, and show that it correctly synchronises 82.9% of test utterances from unseen therapy sessions and unseen speakers, thus considerably reducing the number of utterances to be manually synchronised. An analysis of model performance on the test utterances shows that directed phone articulations are more difficult to automatically synchronise compared to utterances containing natural variation in speech such as words, sentences, or conversations.

Index Terms: Audiovisual synchronisation, speech audio & ultrasound, machine learning, neural-networks, self-supervision.

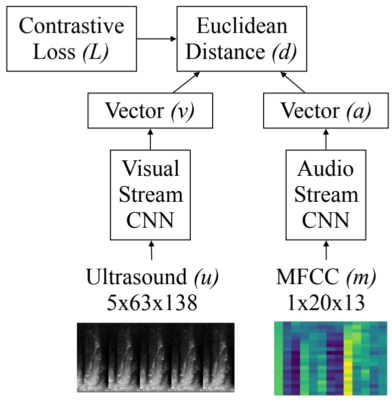

Figure 1: *UltraSync maps high dimensional inputs to low dimensional vectors using a contrastive loss function, such that the Euclidean distance is small between vectors from positive pairs and large otherwise. Inputs span ≃200ms: 5 consecutive raw ultrasound frames on one stream and 20 frames of the corresponding MFCC features on the other.*

1. Introduction

Ultrasound tongue imaging (UTI) is a non-invasive way of observing the vocal tract during speech production [1]. Instrumental speech therapy relies on capturing ultrasound videos of the patient's tongue simultaneously with their speech audio in order to provide a diagnosis, design treatments, and measure therapy progress [2]. The two modalities must be correctly synchronised, with a minimum shift of $+45$ms if the audio leads and -125ms if the audio lags, based on synchronisation standards for broadcast audiovisual signals [3]. Errors beyond this range can render the data unusable – indeed, synchronisation errors do occur, resulting in significant wasted effort if not corrected. No mechanism currently exists to automatically correct these errors, and although manual synchronisation is possible in the presence of certain audiovisual cues such as stop consonants [4], it is time consuming and tedious.

In this work, we exploit the correlation between the two modalities to synchronise them. We utilise a two-stream neural network architecture for the task [5], using as our only source of supervision pairs of ultrasound and audio segments which have been automatically generated and labelled as positive (correctly synchronised) or negative (randomly desynchronised); a process known as self-supervision [6]. We demonstrate how this approach enables us to correctly synchronise the majority of utterances in our test set, and in particular, those exhibiting natural variation in speech.

Section 2 reviews existing approaches for audiovisual synchronisation, and describes the challenges specifically associated with UTI data, compared with lip videos for which automatic synchronisation has been previously attempted. Section 3 describes our approach. Section 4 describes the data we use, including data preprocessing and positive and negative sample creation using a self-supervision strategy. Section 5 describes our experiments, followed by an analysis of the results. We conclude with a summary and future directions in Section 6[1].

2. Background

Ultrasound and audio are recorded using separate components, and hardware synchronisation is achieved by translating information from the visual signal into audio at recording time. Specifically, for every ultrasound frame recorded, the ultrasound beam-forming unit releases a pulse signal, which is translated by an external hardware synchroniser into an audio pulse signal and captured by the sound card [7, 8]. Synchronisation is achieved by aligning the ultrasound frames with the audio pulse signal, which is already time-aligned with the speech audio [9].

Hardware synchronisation can fail for a number of reasons. The synchroniser is an external device which needs to be correctly connected and operated by therapists. Incorrect use can lead to missing the pulse signal, which would cause synchronisation to fail for entire therapy sessions [10]. Furthermore, low-quality sound cards report an approximate, rather than the exact, sample rate which leads to errors in the offset calculation [9]. There is currently no recovery mechanism for when synchronisation fails, and to the best of our knowledge, there has been no prior work on automatically correcting the synchronisation error between ultrasound tongue videos and audio. There is, however, some prior work on synchronising lip movement with audio which we describe next.

[1] Code available at: https://github.com/aeshky/ultrasync

2.1. Audiovisual synchronisation for lip videos

Speech audio is generated by articulatory movement and is therefore fundamentally correlated with other manifestations of this movement, such as lip or tongue videos [11]. An alternative to the hardware approach is to exploit this correlation to find the offset. Previous approaches have investigated the effects of using different representations and feature extraction techniques on finding dimensions of high correlation [12, 13, 14]. More recently, neural networks, which learn features directly from input, have been employed for the task. SyncNet [5] uses a two-stream neural network and self-supervision to learn cross-modal embeddings, which are then used to synchronise audio with lip videos. It achieves near perfect accuracy (>99%) using manual evaluation where lip-sync error is not detectable to a human. It has since been extended to use different sample creation methods for self-supervision [6, 15] and different training objectives [15]. We adopt the original approach [5], as it is both simpler and significantly less expensive to train than the more recent variants.

2.2. Lip videos vs. ultrasound tongue imaging (UTI)

Videos of lip movement can be obtained from various sources including TV, films, and YouTube, and are often cropped to include only the lips [5]. UTI data, on the other hand, is recorded in clinics by trained therapists [16]. An ultrasound probe placed under the chin of the patient captures the midsaggital view of their oral cavity as they speak. UTI data consists of sequences of 2D matrices of raw ultrasound reflection data, which can be interpreted as greyscale images [16]. There are several challenges specifically associated with UTI data compared with lip videos, which can potentially lower the performance of models relative to results reported on lip video data. These include:

Poor image quality: Ultrasound data is noisy, containing arbitrary high-contrast edges, speckle noise, artefacts, and interruptions to the tongue's surface [1, 17, 18]. The oral cavity is not entirely visible, missing the lips, the palate, and the pharyngeal wall, and visually interpreting the data requires specialised training. In contrast, videos of lip movement are of much higher quality and suffer from none of these issues.

Probe placement variation: Surfaces that are orthogonal to the ultrasound beam image better than those at an angle. Small shifts in probe placement during recording lead to high variation between otherwise similar tongue shapes [1, 19, 18]. In contrast, while the scaling and rotations of lip videos lead to variation, they do not lead to a degradation in image quality.

Inter-speaker variation: Age and physiology affect the quality of ultrasound data, and subjects with smaller vocal tracts and less tissue fat image better [1, 18]. Dryness in the mouth, as a result of nervousness during speech therapy, leads to poor imaging. While inter-speaker variation is expected in lip videos, again, the variation does not lead to quality degradation.

Limited amount of data: Existing UTI datasets are considerably smaller than lip movement datasets. Consider for example VoxCeleb and VoxCeleb2 used to train SyncNet [5, 15], which together contain 1 million utterances from 7,363 identities [20, 21]. In contrast, the UltraSuite repository (used in this work) contains 13,815 spoken utterances from 86 identities.

Uncorrelated segments: Speech therapy data contains interactions between the therapist and patient. The audio therefore contains speech from both speakers, while the ultrasound captures only the patient's tongue [16]. As a result, parts of the recordings will consist of completely uncorrelated audio and ultrasound. This issue is similar to that of dubbed voices in lip videos [5], but is more prevalent in speech therapy data.

3. Model

We adopt the approach in [5], modifying it to synchronise audio with UTI data. Our model, UltraSync, consists of two streams: the first takes as input a short segment of ultrasound and the second takes as input the corresponding audio. Both inputs are high-dimensional and are of different sizes. The objective is to learn a mapping from the inputs to a pair of low-dimensional vectors of the same length, such that the Euclidean distance between the two vectors is small when they correlate and large otherwise [22, 23]. This model can be viewed as an extension of a siamese neural network [24] but with two asymmetrical streams and no shared parameters. Figure 1 illustrates the main architecture. The visual data u (ultrasound) and audio data m (MFCC), which have different shapes, are mapped to low dimensional embeddings v (visual) and a (audio) of the same size:

$$\psi(u;\theta) \rightarrow v,\ \phi(m;\eta) \rightarrow a \quad (1)$$

The model is trained using a contrastive loss function [22, 23], L, which minimises the Euclidean distance $d = ||v - a||_2$ between v and a for positive pairs ($y = 1$), and maximises it for negative pairs ($y = 0$), for a number of training samples N:

$$L(\theta,\eta) = \frac{1}{N}\sum_{n=1}^{N} y_n d_n^2 + (1-y_n)\{max(1-d_n,0)\}^2 \quad (2)$$

Given a pair of ultrasound and audio segments we can calculate the distance between them using our model. To predict the synchronisation offset for an utterance, we consider a discretised set of candidate offsets, calculate the average distance for each across utterance segments, and select the one with the minimum average distance. The candidate set is independent of the model, and is chosen based on task knowledge (Section 5).

4. Data

For our experiments, we select a dataset whose utterances have been correctly synchronised at recording time. This allows us to control how the model is trained and verify its performance using ground truth synchronisation offsets. We use UltraSuite[2]: a repository of ultrasound and acoustic data gathered from child speech therapy sessions [16]. We used all three datasets from the repository: UXTD (recorded with typically developing children), and UXSSD and UPX (recorded with children with speech sound disorders). In total, the dataset contains 13,815 spoken utterances from 86 speakers, corresponding to 35.9 hours of recordings. The utterances have been categorised by the type of task the child was given, and are labelled as: Words (A), Non-words (B), Sentence (C), Articulatory (D), Non-speech (E), or Conversations (F). See [16] for details.

Each utterance consists of 3 files: audio, ultrasound, and parameter. **The audio file** is a RIFF wave file, sampled at 22.05 KHz, containing the speech of the child and therapist. **The ultrasound file** consists of a sequence of ultrasound frames capturing the midsagittal view of the child's tongue. A single ultrasound frame is recorded as a 2D matrix where each column represents the ultrasound reflection intensities along a single scan line. Each ultrasound frame consists of 63 scan lines of 412 data points each, and is sampled at a rate of $\simeq$121.5 fps. Raw ultrasound frames can be visualised as greyscale images and can

[2]http://www.ultrax-speech.org/ultrasuite

thus be interpreted as videos. **The parameter file** contains the synchronisation offset value (in milliseconds), determined using hardware synchronisation at recording time and confirmed by the therapists to be correct for this dataset.

4.1. Preparing the data

First, we exclude utterances of type "Non-speech" (E) from our training data (and statistics). These are coughs recorded to obtain additional tongue shapes, or swallowing motions recorded to capture a trace of the hard palate. Both of these rarely contain audible content and are therefore not relevant to our task. Next, we apply the offset, which should be positive if the audio leads and negative if the audio lags. In this dataset, the offset is always positive. We apply it by cropping the leading audio and trimming the end of the longer signal to match the duration.

To process the ultrasound more efficiently, we first reduce the frame rate from $\simeq$121.5 fps to $\simeq$24.3 fps by retaining 1 out of every 5 frames. We then downsample by a factor of (1, 3), shrinking the frame size from 63x412 to 63x138 using max pixel value. This retains the number of ultrasound vectors (63), but reduces the number of pixels per vector (from 412 to 138).

The final pre-preprocessing step is to remove empty regions. UltraSuite was previously anonymised by zero-ing segments of audio which contained personally identifiable information. As a preprocessing step, we remove the zero regions from audio and corresponding ultrasound. We additionally experimented with removing regions of silence using voice activity detection, but obtained a higher performance by retaining them.

4.2. Creating samples using a self-supervision strategy

To train our model we need positive and negative training pairs. The model ingests short clips from each modality of $\simeq$200ms long, calculated as $t = l/r$, where t is the time window, l is the number of ultrasound frames per window (5 in our case), and r is the ultrasound frame rate of the utterance ($\simeq$24.3 fps). For each recording, we split the ultrasound into non-overlapping windows of 5 frames each. We extract MFCC features (13 cepstral coefficients) from the audio using a window length of $\simeq$20ms, calculated as $t/(l\times 2)$, and a step size of $\simeq$10ms, calculated as $t/(l\times 4)$. This give us the input sizes shown in Figure 1.

Positive samples are pairs of ultrasound windows and the corresponding MFCC frames. To create negative samples, we randomise pairings of ultrasound windows to MFCC frames within the same utterance, generating as many negative as positive samples to achieve a balanced dataset. We obtain 243,764 samples for UXTD (13.5hrs), 333,526 for UXSSD (18.5hrs), and 572,078 for UPX (31.8 hrs), or a total 1,149,368 samples (63.9hrs) which we divide into training, validation and test sets.

4.3. Dividing samples for training, validation and testing

We aim to test whether our model generalises to data from new speakers, and to data from new sessions recorded with known speakers. To simulate this, we select a group of speakers from each dataset, and hold out all of their data either for validation or for testing. Additionally, we hold out one entire session from each of the remaining speakers, and use the rest of their data for training. We aim to reserve approximately 80% of the created samples for training, 10% for validation, and 10% for testing, and select speakers and sessions on this basis.

Each speaker in UXTD recorded 1 session, but sessions are of different durations. We reserve 45 speakers for training, 5 for validation, and 8 for testing. UXSSD and UPX contain fewer speakers, but each recorded multiple sessions. We hold out 1 speaker for validation and 1 for testing from each of the two datasets. We also hold out a session from the first half of the remaining speakers for validation, and a session from the second half of the remaining speakers for testing[3]. This selection process results in 909,858 (pooled) samples for training (50.5hrs), 128,414 for validation (7.1hrs) and 111,096 for testing (6.2hrs). From the training set, we create shuffled batches which are balanced in the number of positive and negative samples.

Table 1: *Each stream has 3 convolutional layers followed by 2 fully-connected layers. Fully connected layers have 64 units each. For convolutional layers, we specify the number of filters and their receptive field size as "num×size×size" followed by the max-pooling downsampling factor. Each layer is followed by batch-normalisation then ReLU activation. Max-pooling is applied after the activation function.*

Stream	Conv1	Conv2	Conv3	Full4	Full5
Visual	23x5x5 x2 pool	64x5x5 x2 pool	128x5x5 x2 pool	64	64
Audio	23x3x3	64x3x3 x2 pool	128x3x3 x2 pool	64	64

5. Experiments

We select the hyper-parameters of our model empirically by tuning on the validation set (Table 1). Hyper-parameter exploration is guided by [25]. We train our model using the Adam optimiser [26] with a learning rate of 0.001, a batch size of 64 samples, and for 20 epochs. We implement learning rate scheduling which reduces the learning rate by a factor of 0.1 when the validation loss plateaus for 2 epochs.

Upon convergence, the model achieves 0.193 training loss, 0.215 validation loss, and 0.213 test loss. By placing a threshold of 0.5 on predicted distances, the model achieves 69.9% binary classification accuracy on training samples, 64.7% on validation samples, and 65.3% on test samples.

Synchronisation offset prediction: Section 3 described briefly how to use our model to predict the synchronisation offset for test utterances. To obtain a discretised set of offset candidates, we retrieve the true offsets of the training utterances, and find that they fall in the range [0, 179] ms. We discretise this range taking 45ms steps and rendering 40 candidate values (45ms is the smaller of the absolute values of the detectability boundaries, -125 and $+45$ ms). We bin the true offsets in the candidate set and discard empty bins, reducing the set from 40 to 24 values. We consider all 24 candidates for each test utterance. We do this by aligning the two signals according to the given candidate, then producing the non-overlapping windows of ultrasound and MFCC pairs, as we did when preparing the data. We then use our model to predict the Euclidean distance for each pair, and average the distances. Finally, we select the offset with the smallest average distance as our prediction.

Evaluation: Because the true offsets are known, we evaluate the performance of our model by computing the discrepancy

[3]Held out subsets: UXTD speakers 07, 08, 12, 13, 26 for validation and speakers 30, 38, 43, 45, 47, 52, 53, 55 for testing. UXSSD speaker 01 and session 'Mid' (for speakers 02-04) for validation, and speaker 07 and session 'Mid' (for speakers 05, 06, 08) for testing. UPX speakers 01 and session 'BL3' (for speakers 02-10) for validation, and speaker 15 and session 'BL3' (for speakers 11-14 and 16-20) for testing.

Table 2: *Model accuracy per test set and utterance type. Performance is consistent across test sets for Words (A) where the sample sizes are large, and less consistent for types where the sample sizes are small. 71% of UXTD utterances are Articulatory (D), which explains the low performance on this test set (64.8% in Table 4). In contrast, performance on UXTD Words (A) is comparable to other test sets.*

	Words (A)		**Non-words (B)**		**Sentence (C)**		**Articulatory (D)**		**Conversation (F)**	
Test Set	**N**	**Acc**	**N**	**Acc**	**N**	**Acc**	**N**	**Acc**	**N**	**Acc**
UXTD	108	88.9%	22	86.4%	0	-	325	55.4%	0	-
UXSSD	307	88.6%	20	65.0%	58	94.8%	11	54.5%	0	-
UPX	499	92.4%	16	100.0%	128	93.8%	4	100.0%	4	75.0%
All	914	90.7%	58	82.8%	186	94.1%	340	55.9%	4	75.0%

Table 3: *Model accuracy per utterance type, where N is the number of utterances. Performance is best on utterances containing natural variation in speech, such as Words (A) and Sentences (C). Non-words (B) and Conversations (F) also exhibit this variation, but due to smaller sample sizes the lower percentages are not representative. Performance is lowest on Articulatory utterances (D), which contain isolated phones. The mean and standard deviation of the discrepancy between the prediction and the true offset are also shown in milliseconds.*

Utterance Type	**N**	**Acc**	**Discrepancy**
Words (A)	914	90.7%	1 ± 102 ms
Non-words (B)	58	82.8%	-2 ± 39 ms
Sentence (C)	186	94.1%	16 ± 150 ms
Articulatory (D)	340	55.9%	129 ± 408 ms
Conversation (F)	4	75.0%	-87 ± 141 ms
All	1502	82.9%	32 ± 223 ms
A, B, C and F	1162	91.2%	3 ± 112 ms

between the predicted and the true offset for each utterance. If the discrepancy falls within the minimum detectability range ($-125 < x < +45$) then the prediction is correct. Random prediction (averaged over 1000 runs) yields 14.6% accuracy with a mean and standard deviation discrepancy of 328 ± 518ms. We achieve 82.9% accuracy with a mean and standard deviation discrepancy of 32 ± 223ms. SyncNet reports >99% accuracy on lip video synchronisation using a manual evaluation where the lip error is not detectable to a human observer [5]. However, we argue that our data is more challenging (Section 2.2).

Analysis: We analyse the performance of our model across different conditions. Table 3 shows the model accuracy broken down by utterance type. The model achieves 91.2% accuracy on utterances containing words, sentences, and conversations, all of which exhibit natural variation in speech. The model is less successful with Articulatory utterances, which contain isolated phones occurring once or repeated (e.g., "sh sh sh"). Such utterances contain subtle tongue movement, making it more challenging to correlate the visual signal with the audio. And indeed, the model finds the correct offset for only 55.9% of Articulatory utterances. A further analysis shows that 84.4% (N=90) of stop consonants (e.g., "t"), which are relied upon by therapists as the most salient audiovisual synchronisation cues [4], are correctly synchronised by our model, compared to 48.6% (N=140) of vowels, which contain less distinct movement and are also more challenging for therapists to synchronise.

Table 4 shows accuracy broken down by test set. The model performs better on test sets containing entirely new speakers compared with test sets containing new sessions from previously seen speakers. This is contrary to expectation but could be due to the UTI challenges (described in Section 2.2) affecting different subsets to different degrees. Table 4 shows that the model performs considerably worse on UXTD compared to other test sets (64.8% accuracy). However, a further breakdown of the results in Table 2 by test set and utterance type explains this poor performance; the majority of UXTD utterances (71%) are Articulatory utterances which the model struggles to correctly synchronise. In fact, for other utterance types (where there is a large enough sample, such as Words) performance on UXTD is on par with other test sets.

Table 4: *Model accuracy per test set. Contrary to expectation, performance is better on test sets containing new speakers than on test sets containing new sessions from known speakers. The performance on UXTD is considerably lower than other test sets, due to it containing a large number of Articulatory utterances, which are difficult to synchronise (see Tables 3 and 2).*

Test Set	**N**	**Acc**	**Discrepancy**
UXTD, new speakers	455	64.8%	97 ± 357 ms
UXSSD, new sessions	126	82.5%	19 ± 160 ms
UXSSD, new speaker	270	89.6%	9 ± 135 ms
UPX, new sessions	306	91.2%	-3 ± 40 ms
UPX, new speaker	345	94.2%	-2 ± 123 ms
All	1502	82.9%	32 ± 223 ms

6. Conclusion

We have shown how a two-stream neural network originally designed to synchronise lip videos with audio can be used to synchronise UTI data with audio. Our model exploits the correlation between the modalities to learn cross-model embeddings which are used to find the synchronisation offset. It generalises well to held-out data, allowing us to correctly synchronise the majority of test utterances. The model is best-suited to utterances which contain natural variation in speech and least suited to those containing isolated phones, with the exception of stop consonants. Future directions include integrating the model and synchronisation offset prediction process into speech therapy software [7, 8], and using the learned embeddings for other tasks such as active speaker detection [5].

7. Acknowledgements

Supported by EPSRC Healthcare Partnerships Programme grant number EP/P02338X/1 (Ultrax2020).

8. References

[1] M. Stone, "A guide to analysing tongue motion from ultrasound images," *Clinical Linguistics and Phonetics*, vol. 19, no. 6-7, pp. 455–501, 2005.

[2] J. Cleland, J. M. Scobbie, and A. A. Wrench, "Using ultrasound visual biofeedback to treat persistent primary speech sound disorders," *Clinical Linguistics and Phonetics*, vol. 29, no. 8-10, pp. 575–597, 2015.

[3] ITU-R, "Recommendation ITU-R BT.1359: Relative timing of sound and vision for broadcasting," January 1998.

[4] M. Alm and D. Behne, "Audio-visual speech experience with age influences perceived audio-visual asynchrony in speech," *The Journal of the Acoustical Society of America*, vol. 134, no. 4, pp. 3001–3010, 2013.

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

[6] B. Korbar, D. Tran, and L. Torresani, "Cooperative learning of audio and video models from self-supervised synchronization," in *Advances in Neural Information Processing Systems*, 2018, pp. 7763–7774.

[7] A. Wrench, "SonoSpeech: Ultrasound application for recording, client assessment and visual feedback," Software v217.10, 2018.

[8] ——, "Articulate Assistant Advanced (AAA): Research application for recording and analysing ultrasound, EPG, EMA and other instrumental data," Software v217.10, 2018.

[9] ——, "Discussion regarding the hardware synchronisation method used in SonoSpeech and Articulate Assistant Advanced," Personal Communication, 2018.

[10] J. Cleland, "Discussion regarding the use of hardware synchronisation by speech and language therapists," Personal Communication, 2018.

[11] H. Yehia, P. Rubin, and E. Vatikiotis-Bateson, "Quantitative association of vocal-tract and facial behavior," *Speech Communication*, vol. 26, no. 1-2, pp. 23–43, 1998.

[12] M. E. Sargin, Y. Yemez, E. Erzin, and A. M. Tekalp, "Audiovisual synchronization and fusion using canonical correlation analysis," *IEEE Transactions on Multimedia*, vol. 9, no. 7, pp. 1396–1403, 2007.

[13] H. Bredin and G. Chollet, "Audiovisual speech synchrony measure: application to biometrics," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 179–179, 2007.

[14] G. Garau, A. Dielmann, and H. Bourlard, "Audio-visual synchronisation for speaker diarisation," in *11th Annual Conference of the International Speech Communication Association (Interspeech)*, 2010, pp. 2654–2657.

[15] S. Chung, J. S. Chung, and H. Kang, "Perfect match: Improved cross-modal embeddings for audio-visual synchronisation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019.

[16] A. Eshky, M. S. Ribeiro, J. Cleland, K. Richmond, Z. Roxburgh, J. M. Scobbie, and A. A. Wrench, "Ultrasuite: a repository of ultrasound and acoustic data from child speech therapy sessions," in *19th Annual Conference of the International Speech Communication Association (Interspeech)*, 2018, pp. 1888–1892.

[17] M. Li, C. Kambhamettu, and M. Stone, "Automatic contour tracking in ultrasound images," *Clinical linguistics & phonetics*, vol. 19, no. 6-7, pp. 545–554, 2005.

[18] M. S. Ribeiro, A. Eshky, K. Richmond, and S. Renals, "Speaker-independent classification of phonetic segments from raw ultrasound in child speech," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019.

[19] J. Cleland, A. Wrench, S. Lloyd, and E. Sugden, "Ultrax2020: Ultrasound technology for optimising the treatment of speech disorders: Clinicians' resource manual," University of Strathclyde, Tech. Rep., 2018.

[20] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *18th Annual Conference of the International Speech Communication Association (Interspeech)*, 2017, pp. 2616–2620.

[21] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *19th Annual Conference of the International Speech Communication Association (Interspeech)*, 2018, pp. 1086–1090.

[22] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, pp. 539–546.

[23] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2006, pp. 1735–1742.

[24] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *Advances in neural information processing systems*, 1994, pp. 737–744.

[25] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *25th British Machine Vision Conference (BMVC)*, 2014.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.