A Simple and Efficient method for Dubbed Audio Sync Detection using Compressive Sensing

Avijit Vajpayee
Amazon Prime Video
avivaj@amazon.com

Zhikang Zhang*
Arizona State University
zhikang.zhang@asu.edu

Abhinav Jain
Amazon Prime Video
jaabhin@amazon.com

Vimal Bhat
Amazon Prime Video
vimalb@amazon.com

Abstract

Lack of temporal synchronization between audio and video streams represents one of the major quality defects in videos. The defect is more prominent in dubbed media due to errors in post-production such as improper audio overlay. Prior works in Audio-Video sync detection rely on either lip synchronization methods, which cannot be applied to dubbed media, or on self-supervised embeddings for general sound events, which are not accurate. In this paper, we present a novel, accurate and efficient method for temporal sync detection between dubbed audio tracks and corresponding non-dubbed original-language audio tracks. Using the availability of non-dubbed audio tracks and existing lip sync methods, we can simplify the problem of "Dubbed Audio-to-Video" sync detection to that of "Dubbed Audio-to-Original Audio" sync detection. Our method finds and compares matching frames in compressed audio signatures, achieving near perfect classification with 99.4 F1 score in less than 1 minute of processing time per hour of audio, along with ≈ 99.6% relative reduction in memory footprint compared to an uncompressed full audio spectrogram. We believe this is the first work to tackle temporal sync detection in dubbed media.

1. Introduction

Ensuring synchronization in the time domain between multiple modalities such as visual and audio is an important aspect of video quality. Lack of such synchronization causes degradation of speech comprehension and difficulties in audio-visual integration [27]. Studies conducted by the International Telecommunications Union [18] suggest that viewers are able to perceive desynchronizations as low as $-125ms$ (audio lag) and $+45ms$ (audio lead).

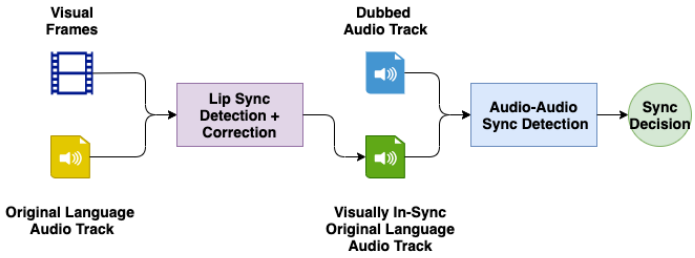

Figure 1: Overall design of Dubbed A/V Sync utilizing Lip Sync and Audio-to-Audio Sync detection

Online streaming services have grown exponentially in recent years leading to a significant growth in cinematic media. This has also lead to an increase in number of dubbed movies. A/V sync issues are more common in dubbed content due to additional errors introduced during post-production such as :

- Improper overlay of dubbed audio track on the visual stream.
- Poor quality dubbing.

Despite its important need in real-world applications, audio-video sync detection in dubbed content has been neglected in research so far. Prior works for audio-video sync detection task have relied on lip synchronization methods [7, 8, 9] that learn how to correlate lip movements to speech signals. Lip synchronization is not applicable on dubbed movies where the lips movements correspond to the original filmed language whereas speech signals correspond to the dubbing language.

Given the original language audio is typically available with dubbed cinematic content, we can utilize existing lip synchronization methods to verify that the original language

*Work done as part of internship at Amazon

audio track is in sync with visuals. With this, the problem of solving Audio-to-Video synchronization for dubbed audio resolves to that of Dubbed Audio-to-Original Audio synchronization as shown in figure 1. The key contributions of this paper can be summarized as follows :

1. Present a novel approach for A/V Sync Detection in dubbed videos by using Lip Sync and Audio-to-Audio synchronization. We believe this is the first paper to tackle A/V sync quality issues in dubbed videos.

2. Present a novel, fast and memory efficient method for Audio-to-Audio synchronization by finding matching segments between two audio signatures. The audio signatures are generated by applying compressive sensing techniques on frequency-vs-time spectrograms. Compressive sensing is a signal processing technique in which the signal is sparsely sampled allowing for efficient storage.

3. Demonstrate effectiveness of compressive sensing to efficiently generate audio signatures achieving 99.6% relative reduction in dimensionality without loss in sync detection performance against a baseline of full-spectrogram based matching.

Section 2 gives an overview of prior research in related areas. Section 3 provides the technical details of our approaches. Section 4 and 5 describe the dataset used and our experimental results respectively. Finally the conclusions as well as directions for future research are outlined in section 6.

2. Related Works

Lip Synchronization : Focus on synchronizing lip movements with speech in "talking heads" segments of video clips. Earlier works [22, 24] utilized explicit phoneme (fundamental unit of language) to viseme (lip configuration corresponding to phoneme) matching. Lewis (1991) [22] utilized phoneme recognition on audio, whereas Morishima *et al*. (2002) [24] classified face parameters to visemes. SyncNet V1 by Chung and Zisserman (2016) [7] was a 2-branch Siamese style CNN architecture that learned to correlate lip movements and speech in short segments ($\approx$ 0.2 sec) without doing explicit phoneme to viseme matching. SyncNet V2 by Chung and Zisserman (2017) [8] improved V1 to provide accurate predictions for non-frontal speaking videos through a curriculum learning strategy of training on increasingly harder samples of in-profile face video clips. Perfect Match by Chung *et al*. (2019) [9] incorporated a multi-class matching loss over SyncNet architecture, providing minor improvements. SyncNet and its variants have been shown to be very accurate at lip sync detection in non-dubbed original language videos achieving $>$ 98% accuracy. They however are not applicable on dubbed content as the basic premise of correlation between lip movements and speech signals does not hold. Beyond audio-video sync detection, automated correction of lip sync errors has also been explored by Halperin *et al*. (2019) [15], who applied Dynamic Time Warping (DTW) on features extracted from SyncNet to align video to speech for Automatic Dialogue Replacement (ADR).

General A/V Synchronization : Super-set of lip sync where general sound events (such as telephone ringing, ball bouncing) are synchronized with corresponding visual segments. Foley-style detection [12] has been previously applied on Tennis videos by synchronising the timestamps via audio and video event detectors for a ball bouncing. This approach is however not scalable for other general sound events commonly encountered in movie/episodic content. Other approaches [6, 19, 21, 23, 29, 38] focussed on using audio-visual synchrony as a pre-text self-supervised task for learning generic video representations that are transferrable to other downstream tasks such as Action Recognition and Sound Source Separation. As such these models cover a more generic use-case but have limited accuracy (60-70%) at a higher degree of desynchronization (2-6 seconds) than desired ($\approx$ 0.2 seconds). Owens and Efros (2018) [29] utilize a 2-branch late fusion CNN architecture to learn audio-visual embeddings. Korbar *et al*. (2018) [21] utilized a similar 2-branch CNN but improved the training regime by incorporating a curriculum learning strategy to sample negative examples (progressing from large to medium desynchronizations). Recent works have included attention mechanisms for late or early fusion of multi-modal signals. Khosravan *et al*. (2019) [19] utilized late spatial and temporal attention over video block representations (same stem architecture as Owens and Efros). Hierarchical fusion of audio-video features has been explored, with Cheng *et al*. (2020) [6] utilizing a transformer style architecture [34] and Xiao *et al*. (2020) [38] augmenting Slow-Fast networks [13] with an additional audio modality branch.

Acoustic Fingerprinting : Cano *et al*. (2002) [3] described an acoustic fingerprint as a content based compact signature that summarizes an audio recording. Acoustic fingerprinting has been utilized in several use-cases such as music identification and search [35], audio copy detection [28], music genre identification [16], advertisement tracking [4] etc. Shazam algorithm by Wang (2003) [35] is one of the most popular methods of acoustic fingerprinting and is based on amplitude peak pairs in the spectrogram representation of an audio signal. Panako by Six and Leman (2014) [33] improved on basic amplitude landmarks

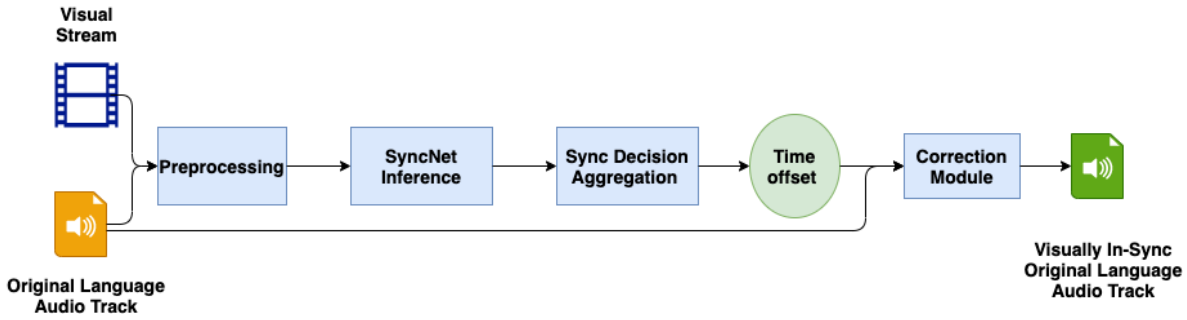

Figure 2: Overall pipeline to perform lip sync detection via SyncNet.

by adding robust handling measures for time-scale and pitch modifications. Most recently, SAMAF [2] by Suárez *et al*. (2020) utilized a sequence-to-sequence autoencoder model for audio fingerprinting to achieve state-of-art results in audio identification on VoxCeleb 1 dataset [26]. Audio fingerprinting techniques have been previously explored by Shrstha *et al*. (2007) for synchronization of multi-camera video recordings [32].

Compressive Sensing : Is a signal processing method to acquire and reconstruct a signal that is sparse in some domain from far fewer samples than required by Nyquist–Shannon sampling theorem [25]. Compressive sensing has been applied in a wide range of domains such as video [30, 17], images [40, 39] and audio [37, 36, 1]. Utilizing the fact that audio signals are typically sparse in frequency domain (i.e., energy is present in only some frequencies), compressive sensing methods can be applied on audio signals to represent them efficiently. Acoustic fingerprinting via compressive sensing has been previously explored by Saravanos *et al*. (2020) [31] who utilized K-SVD based dictionary learning for song identification.

As highlighted earlier, lip synchronization methods are very accurate in original language A/V sync detection but cannot be applied on dubbed media where there is no correlation between lip movements and spoken speech. In theory, general A/V Sync methods may be extended to dubbed A/V sync detection but in practice are not accurate and cannot detect low levels (0.2 - 2 seconds) of desynchronizations that are perceivable by humans. For this reason, we try an approach of Audio-to-Audio Sync detection utilizing audio signatures generated via compressive sensing on audio spectrograms. At a high level, our idea is similar to Courtenay and Ellis who utilized matching pursuit and locality sensitive hashing to identify similar acoustic events in a database [10] (2009) and audio fingerprinting to identify multiple videos of an event [11] (2010). We believe we are the first work to tackle the use-case of dubbed audio synchronization as well as use compressive sensing based audio fingerprints for the same.

3. Methodology

There are two separate parts to solving dubbed A/V sync as as shown in figure 1.

1. *Lip Sync Detection* : To verify that the original language audio track is in-sync with the visual stream.

2. *Audio-to-Audio Sync Detection* : To detect the exact time offset (sync error) between the dubbed audio track and the original language audio track.

3.1. Lip Sync Detection

For lip sync detection, we use the Multi-View SyncNet architecture (SyncNet V2) [8] which predicts the time offset between "talking heads" video clips and their corresponding speech segments. The overall pipeline for SyncNet based lip sync detection is shown in figure 2. The components are described as :

Pre-Processing Sub-dividing the entire video into face-centered clips of candidate human speakers. Shot boundaries are first detected using a Histogram-of-Gradients (HoG) based classifier over Hue-Saturation-Value space [14]. For each candidate shot, face bounding boxes of the same person are detected and grouped together using off-the-self face detection and tracking.

SyncNet Inference Multi-view SyncNet is a Siamese-style [20] network that is trained to correlate audio and lip movements for short segments. It comprises of two parallel branches :

1. *Visual Representation* : Modified VGG-M CNN architecture [5] over 5 video frames (224 x 224 images).
2. *Audio Representation* :VGG-M CNN architecture [5] over input Mel-Frequency Cepstral Coefficient (MFCC) features. Input audio features are

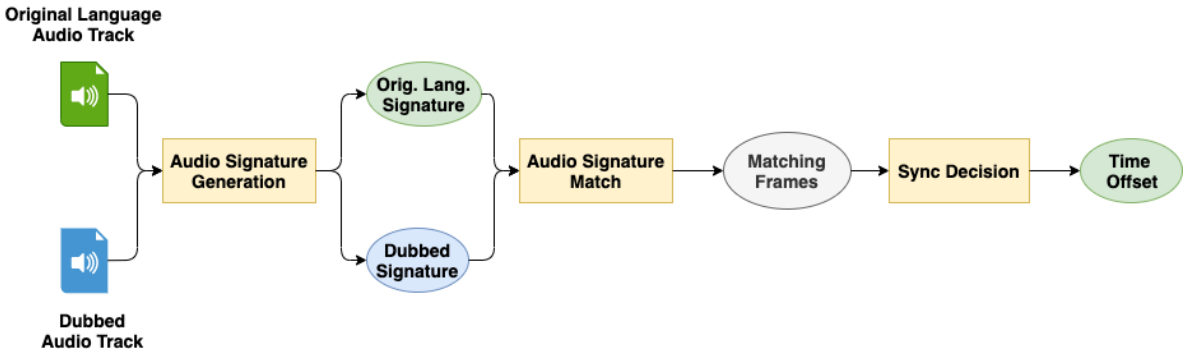

Figure 3: Overall pipeline for audio-to-audio sync detection.

represented as a heatmap image with 13 MFCC coefficients on y-axis and 20 timesteps (sampling rate $\approx$ 100 Hz for a 0.2 second signal).

Audio and visual representations are trained to be in the same semantic space by minimizing a contrastive loss, i.e., output of audio and visual networks are similar for *genuine (in-sync)* pairs and dissimilar for *false (out-of-sync)* pairs. Mathematically, the loss is defined as :

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

$$d_n = ||v_n - a_n||^2$$

where v_n and a_n are the representations via SyncNet for short visual and audio segments respectively and d_n is the $L2$ distance between them.

Sync Decision Aggregation For each frame in a candidate face track clip, the time offset can be found via a sliding window approach, i.e., finding the index of minimum L2 distance between visual segment representation (centered at current frame) and all audio segment representations in a $\pm x$ second window. Time offset for coarser granularities (such as time offset for face-track clip or entire video) is found by taking median statistics of time offset predictions at finer granularities (such as per frame and per face-track clip respectively).

Automated Correction For the scope of this paper we limit ourselves to constant sync errors that are present from the very start of the video. To correct them, we employ a simple strategy of shifting the audio stream in the opposite direction of full-video predicted time offset. To correct variable sync errors (i.e., different desynchronization per face track), Dynamic Time Warping as described by Haperin *et al*. (2019) [15] can be employed.

3.2. Audio-to-Audio Sync Detection

The key idea behind designing a solution for dubbed audio to original language audio sync detection is the fact that only certain segments of an audio track are actually dubbed over. General sound events such as a gunshot being fired, an instrument being played, background score, environmental sounds, etc. remain the same regardless of the language of audio track. Therefore we can find exact matching segments between the two audio tracks and compare their relative timestamps to predict the overall time offset. The overall pipeline for original audio to non-dubbed audio sync detection is depicted in figure 3.

3.2.1 Audio Signature Generation

Comparison between raw audio waveforms to find all matching segments is computationally inefficient, especially for long audio tracks such as movies. Therefore, we generate compressed audio signatures for both the dubbed audio track and original language audio track. The output audio signature is a per audio frame representation which allows downstream signature match logic to give per frame time offsets. Figure 4 describes the pipeline for audio signature generation via compressive sensing. Individually the components are as follows :

Spectrogram Generation : The raw audio waveform is converted into its power spectrogram using Short-Term Fourier Transform. The power spectrogram $Spec$ is a matrix with rows as frequencies, columns as time indices (audio frame) and values as loudness in dB at that frequency and time. $Spec$ has a dimensionality of $F * T$ where :

- F : dimensionality in frequency domain determined by the window size in STFT.

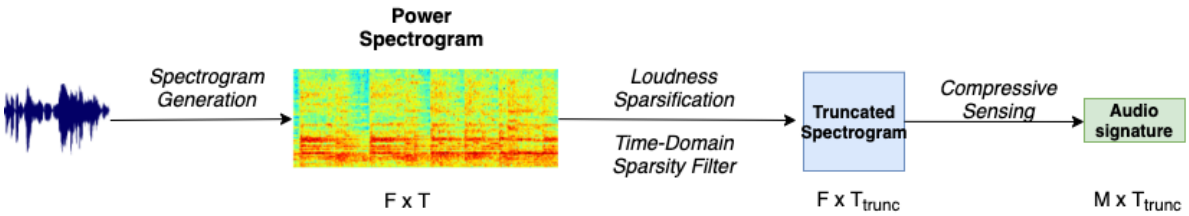

Figure 4: Audio signature generation via compressive sensing. The output signature is compressed in both time and frequency domains. M << F.

- T : dimensionality in time domain determined by the duration of audio track and size of each audio frame via STFT.

Loudness Sparsification : All loudness values less than threshold θ_{dB} are truncated to zero. This allows our signature to be robust against minor noise in a similar manner to amplitude landmark based audio signature generation methods such as Shazam [35].

Time-Domain Sparsity Filter : All columns, i.e., time indices or audio frames that are too sparse (i.e., few frequencies having 0 loudness) or too noisy (i.e., too many frequencies having non-zero loudness) are dropped from the power spectrogram. Dropping such frames makes the output audio signature have a smaller memory footprint as well as speeds up the downstream match logic. The output is a truncated spectrogram matrix $Spec_{trunc}$ of dimensionality $F * T_{trunc}$.

Compressive Sensing : A compressive sensing matrix CSM is a randomly initialized Bernoulli matrix of dimensions $M * F$ where :

- F : Frequency dimensionality which is the same as in the spectrogram generation step.
- M : Measurement dimension equal to $\frac{F}{compression\_ratio}$

We multiply the compression matrix CSM to the truncated spectrogram $Spec_{trunc}$ giving an output audio signature of dimensionality $M * T_{trunc}$. It must be noted that the same random seed is used for initialization of CSM each time to ensure consistency in audio signatures. It must be noted that the compressive sensing typically refers to both the compression as well as reconstruction of the original signal, where reconstruction can be done by $Signature * CSM^{-1}$. However, for this paper we do not reconstruct the original signal and perform sync detection by matching the signatures directly.

3.2.2 Audio Signature Match and Sync Decision

To find all the matching segments in two audio tracks, we utilize a sliding window approach. For all audio frame representations in source signature, we find the Chebyshev distance ($L_{\inf}$ norm) with audio frames in a $\pm x$ second range in the target signature. The index of minimum $L_{\inf}$ distance gives the frame level time offset.

To make sure that we only consider source audio frames that were close to exact matches in the target audio while making a full-audio sync decision, we set a threshold of θ_{min_dist}. Only audio frames with minimum $L_{\inf}$ distance $< \theta_{min\_dist}$ are considered for full-audio sync decision, with the track level time offset = median of exact match frame offsets.

4. Dataset

Statistic	Movie	Episodes	Total
Num titles	50	40	90
Num source lang	6	4	7
Num target lang	7	12	13
Num desyncs	10	10	10
Total samples	550	440	990
Hours source videos	83	22	105

Table 1: Summary statistics of evaluation dataset

We evaluate Audio-Audio sync detection both as a regression task to predict the exact time offset in between the original language and dubbed audio tracks as well, as a binary classification task (out-of-sync as positive class and in-sync as negative class). For binary classification, all offset predictions and ground truth desyncs of > 0.2 seconds or $<$ -0.2 seconds are considered with positive label (out-of-sync).

We collected 90 full-length video pairs ($\approx$ 105 hours) in their original and dubbed languages, comprising of 50 cinematic movies (duration 1 to 2.5 hours) and 40 episodes (duration 20 minutes to 1 hour). We ensured a diverse mix of languages for both source (original language) and target

Model	Signature Dimensionality	Signature Size (in MB)	Time to Fingerprint	Time to Match
Spectrogram	$1.59 * 10^8$	524.3 MB	51.72 seconds	$>$ 30 minutes
Spectrogram + Compressive Sensing	$4.96 * 10^6$	16.62 MB	51.67 seconds	175.3 seconds
Spectrogram + Time-Domain Sparsity Filtering	$2.01 * 10^7$	68.1 MB	51.48 seconds	363.2 seconds
Spectrogram + Time-Domain Sparsity Filtering + Compressive Sensing	$6.29 * 10^5$	2.23 MB	51.44 seconds	3.66 seconds

Table 2: Size and time comparisons for audio signature generation. The statistics have been averaged per hour of audio.

Model	Classification			Regression
	Precision	Recall	F1	MAE
Spectrogram + Time-Domain Sparsity Filter	0.988	1	0.994	0.02 sec
Spectrogram + Time-Domain Sparsity Filter + Compressive Sensing	0.988	1	0.994	0.02 sec

Table 3: Classification and regression performance for Dubbed Audio-to-Original Language Audio Sync Detection

(dubbed) videos, with the total number of unique languages being 15. All the dubbed videos were artificially desynchronised by $\pm$ 0.05, 0.5, 1.0, 5.0, 30.0 seconds (+ being audio lead, - being audio lag). Thus the final dataset contains 990 video pairs (90 unique titles x 11 desync values). Samples with desync of $\pm$ 0, 0.05 seconds are taken as actual negative class (in-sync) and samples with desync of $\pm$ 0.5, 1.0, 5.0, 30.0 seconds are taken as actual positive class (out-of-sync) giving a class imbalance of 8:3 (positive:negative). Table 1 gives the summary statistics for our dataset.

All source (original language) audio tracks were verified to be in sync with the visual using SyncNet as described in section 3.1. As audio-to-audio sync detection is completely unsupervised the dataset described above was completely for testing and evaluation. We used a separate held-out dataset of 10 titles (5 episodes and 5 movies) to tune hyperparameters θ_{dB} (threshold for loudness sparsification) and θ_{min_dist} (threshold for minimum frame $L_{\inf}$ distance to be considered in full-title sync decision).

5. Experimental Results

For each experiment, we generated spectogram S from STFT with sample rate of 44.1 kHz and hop length of 2048. The frequency resolution F of power spectrogram S is thus 2049 and time resolution (size of a single audio frame) is $\approx$ 46 milliseconds. With a compression ratio of 32, the frequency resolution M of output signature is 64. Table 2 shows the effect of each stage of audio signature generation on output signature size as well as time to compute.

We are able to achieve 99.6 % relative reduction in dimensionality from $1.59 * 10^8$ for basic spectrogram to $6.29{*}10^5$ for audio signature via compressive sensing. With compressive sensing, output audio signatures are just 2.23 MB per hour of audio, which is a 99.6 % improvement over storing uncompressed audio spectrogram (524 MB).

The low memory footprint of output audio signatures via compressive sensing also makes matching significantly faster, taking only $\approx$ 4 seconds per hour of audio as compared to $>$ 30 minutes of matching time for uncompressed spectrogram. It must be noted that both the optimizations of time-domain sparsity filtering and compressive sensing are highly optimized vector operations that cause an insignificant increase in time to generate audio signatures.

Table 3 gives the classification and regression results for dubbed audio-to-original language audio sync detection. We achieve near perfect sync detection with an F1 score of 99.3 and median absolute error between actual offset and predicted offset of just 0.02 seconds. This shows that the achievements for efficiency do not come at a cost of accuracy. It also indicates that finding approximate exact frame matches in audio does not require a large amount of information to be encoded.

6. Conclusions and Future Work

In this paper, we demonstrated an unsupervised audio match based solution for Dubbed A/V sync detection under the assumption that the corresponding original language audio track is available. We achieved near perfect performance for full movie/episode videos sync detection with an F1 score of 0.994 and median absolute error of just 0.02 seconds. We believe this was the first work to tackle sync quality issues in dubbed media.

Because of the requirement to have the corresponding original language audio available, this method should be treated as a pesudo-reference approach. Unlike traditional full-reference detectors, we do not require explicit presentation time-stamps or watermarks between the visual stream or any of the audio tracks. One of the areas for future research would be to develop a no-reference Dubbed A/V sync detection using CV, without simplifying the problem to one of Audio-to-Audio sync detection.

We also showed that compressive sensing is a very fast and efficient method for generating audio signatures, achieving a $\approx 99.6\%$ relative reduction in signature dimensionality compared to an uncompressed full audio spectrogram. This huge improvement in efficiency does not compromise accuracy in the downstream task of audio-to-audio sync detection.

Finally, the scope of this paper was currently limited to A/V sync detection on dubbed content. We plan to explore compressive sensing based audio fingerprinting for other use-cases such as music identification and search, music metadata matching, etc.

References

[1] Vinayak Abrol, Pulkit Sharma, and Anil Kumar Sao. Speech enhancement using compressed sensing. In *INTERSPEECH*, pages 3274–3278, 2013.

[2] Abraham Báez-Suárez, Nolan Shah, Juan Arturo Nolazco-Flores, Shou-Hsuan S Huang, Omprakash Gnawali, and Weidong Shi. Samaf: Sequence-to-sequence autoencoder model for audio fingerprinting. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 16(2):1–23, 2020.

[3] Pedro Cano, Eloi Batle, Ton Kalker, and Jaap Haitsma. A review of algorithms for audio fingerprinting. In *2002 IEEE Workshop on Multimedia Signal Processing.*, pages 169–173. IEEE, 2002.

[4] Jose Ramon Cerquides. A real time audio fingerprinting system for advertisement tracking and reporting in fm radio. In *2007 17th International Conference Radioelektronika*, pages 1–4. IEEE, 2007.

[5] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

[6] Ying Cheng, Ruize Wang, Zhihao Pan, Rui Feng, and Yuejie Zhang. Look, listen, and attend: Co-attention network for self-supervised audio-visual representation learning. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3884–3892, 2020.

[7] Joon Son Chung and Andrew Zisserman. Out of time: automated lip sync in the wild. In *Asian conference on computer vision*, pages 251–263. Springer, 2016.

[8] Joon Son Chung and AP Zisserman. Lip reading in profile. 2017.

[9] Soo-Whan Chung, Joon Son Chung, and Hong-Goo Kang. Perfect match: Improved cross-modal embeddings for audio-visual synchronisation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3965–3969. IEEE, 2019.

[10] Courtenay Cotton and Daniel PW Ellis. Finding similar acoustic events using matching pursuit and locality-sensitive hashing. In *2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 125–128. IEEE, 2009.

[11] Courtenay V Cotton and Daniel PW Ellis. Audio fingerprinting to identify multiple videos of an event. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2386–2389. IEEE, 2010.

[12] Joshua P Ebenezer, Yongjun Wu, Hai Wei, Sriram Sethuraman, and Zongyi Liu. Detection of audio-video synchronization errors via event detection. *arXiv preprint arXiv:2104.10116*, 2021.

[13] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019.

[14] Igor S Gruzman and Anna S Kostenkova. Algorithm of scene change detection in a video sequence based on the threedimensional histogram of color images. In *2014 12th International Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*, pages 1–1. IEEE, 2014.

[15] Tavi Halperin, Ariel Ephrat, and Shmuel Peleg. Dynamic temporal alignment of speech to lips. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3980–3984. IEEE, 2019.

[16] K Herkiloglu, O Gursoy, and B Gunsel. Music genre determination using audio fingerprinting. In *2006 IEEE 14th Signal Processing and Communications Applications*. 2006.

[17] Michael Iliadis, Leonidas Spinoulas, and Aggelos K Katsaggelos. Deep fully-connected networks for video compressive sensing. *Digital Signal Processing*, 72:9–18, 2018.

[18] ITU. Requirements for operational monitoring of video-to-audio delay in the distribution of television programs, 2008. Recommendation J.248.

[19] Naji Khosravan, Shervin Ardeshir, and Rohit Puri. On attention modules for audio-visual synchronization. In *CVPR Workshops*, pages 25–28, 2019.

[20] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

[21] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. *arXiv preprint arXiv:1807.00230*, 2018.

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

[23] Pedro Morgado, Ishan Misra, and Nuno Vasconcelos. Robust audio-visual instance discrimination. *arXiv preprint arXiv:2103.15916*, 2021.

[24] Shigeo Morishima, Shin Ogata, Kazumasa Murai, and Satoshi Nakamura. Audio-visual speech translation with automatic lip syncqronization and face tracking based on 3-d head model. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages II–2117. IEEE, 2002.

[25] Ahmad Mousavi, Mehdi Rezaee, and Ramin Ayanzadeh. A survey on compressive sensing: Classical results and recent advancements. *arXiv preprint arXiv:1908.01014*, 2019.

[26] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language*, 60:101027, 2020.

[27] Jordi Navarra, Argiro Vatakis, Massimiliano Zampini, Salvador Soto-Faraco, William Humphreys, and Charles Spence. Exposure to asynchronous audiovisual speech extends the temporal window for audiovisual integration. *Cognitive Brain Research*, 25(2):499–507, 2005.

[28] Chahid Ouali, Pierre Dumouchel, and Vishwa Gupta. A robust audio fingerprinting method for content-based copy detection. In *2014 12th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 1–6. IEEE, 2014.

[29] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–648, 2018.

[30] Jae Young Park and Michael B Wakin. A multiscale framework for compressive sensing of video. In *2009 Picture Coding Symposium*, pages 1–4. IEEE, 2009.

[31] Christina Saravanos, Dimitris Ampeliotis, and Kostas Berberidis. Audio-fingerprinting via dictionary learning. In *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–7. IEEE, 2020.

[32] Prarthana Shrstha, Mauro Barbieri, and Hans Weda. Synchronization of multi-camera video recordings based on audio. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 545–548, 2007.

[33] Joren Six and Marc Leman. Panako: a scalable acoustic fingerprinting system handling time-scale and pitch modification. In *15th International Society for Music Information Retrieval Conference (ISMIR-2014)*, 2014.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[35] Avery Wang et al. An industrial strength audio search algorithm. In *Ismir*, volume 2003, pages 7–13. Citeseer, 2003.

[36] Jia-Ching Wang, Yuan-Shan Lee, Chang-Hong Lin, Shu-Fan Wang, Chih-Hao Shih, and Chung-Hsien Wu. Compressive sensing-based speech enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(11):2122–2131, 2016.

[37] Dalei Wu, Wei-Ping Zhu, and MNS Swamy. A compressive sensing method for noise reduction of speech and audio signals. In *2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1–4. IEEE, 2011.

[38] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020.

[39] Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. Admm-csnet: A deep learning approach for image compressive sensing. *IEEE transactions on pattern analysis and machine intelligence*, 42(3):521–538, 2018.

[40] Jian Zhang, Debin Zhao, Chen Zhao, Ruiqin Xiong, Siwei Ma, and Wen Gao. Image compressive sensing recovery via collaborative sparsity. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2(3):380–391, 2012.