DETECTION OF AUDIO-VIDEO SYNCHRONIZATION ERRORS VIA EVENT DETECTION

Joshua P. Ebenezer [*], *Yongjun Wu, Hai Wei, Sriram Sethuraman, Zongyi Liu*

Amazon Prime Video, Seattle, WA

ABSTRACT

We present a new method and a large-scale database to detect audio-video synchronization(A/V sync) errors in tennis videos. A deep network is trained to detect the visual signature of the tennis ball being hit by the racquet in the video stream. Another deep network is trained to detect the auditory signature of the same event in the audio stream. During evaluation, the audio stream is searched by the audio network for the audio event of the ball being hit. If the event is found in audio, the neighboring interval in video is searched for the corresponding visual signature. If the event is not found in the video stream but is found in the audio stream, A/V sync error is flagged. We developed a large-scaled database of 504,300 frames from 6 hours of videos of tennis events, simulated A/V sync errors, and found our method achieves high accuracy on the task.

Index Terms— Deep Learning, Database, Audio Video Synchronization

1. INTRODUCTION

Video quality is affected by a number of distortions that may occur at different stages of the video processing pipeline, from capturing to display. Most videos are accompanied by an audio stream that is synchronized with the visual stream. When the audio stream is not synchronized with the visual stream and is offset by a certain amount, humans are able to detect the offset and this negatively affect the user's quality of experience. These errors could occur during content capture, encoding, post-production, transmission, or play-back. Studies conducted by the International Telecommunication Union [1] have found that humans typically find audio offsets of +50ms or more when audio is advanced with respect to (w.r.t.) the video stream, or -125 ms when audio is delayed w.r.t. vision, to be unacceptable. Zhao et al. [2] showed, through a human study, that humans find A/V sync errors to be even more detrimental to subjective experience than video impairments or audio impairments for virtual reality environments. Detecting such errors in A/V sync is a challenging task, but developing automated algorithms for it is vital for video-on-demand and livestreaming services to operate at scale.

Previous efforts have focused on end-to-end training of the audio and the video against a sync/not-synced binary decision. Korbar et al. [3] proposed a two-stream network for the task. They defined "easy" negatives as audio and visual segments that are from different videos, and "hard" negatives as segments taken from the same video but offset by at least half a second. They reported poor performance on hard negatives and "super-hard" negatives, which are segments offset by less than half a second. This is a major drawback in their approach. Khosravan et al. [4] studied the use of attention models for the task and showed that spatio-temporal attention models can improve performance for end-to-end learning. The drawback in the method is that, by design, it cannot detect errors that are less than 2s in magnitude. Both training and evaluation sets have error magnitudes larger than 2s. Chung and Zissermann [5] proposed an end-to-end network for lip-sync detection, but their method is not generalizable to A/V sync and is uniquely suited for lip-sync errors as it involves the tracking of the mouth in the video stream.

End-to-end paradigms suffer from the issues of not being able to predict small offsets and being difficult to interpret. Some other approaches [6, 7] propose the embedding of a unique signal into the audio and video streams, but this is not always feasible when one does not have control over content production. In this paper, we present a large-scale database of tennis videos and propose a novel method that is amenable to interpretation and can predict very small offsets with high accuracy. Videos of tennis events are labeled frame-by-frame as the events "the ball is being hit by the racquet" (hit), "the ball is bouncing" (bounce), and "the ball is not in play or it is neither being hit nor is it bouncing" (neither). We create an audio event detector (AED) as a deep network that predicts whether segments in the audio stream correspond to a hit or not. We also create a video event detector (VED) as a deep network that predicts whether groups of frames in the visual stream correspond to the ball being hit or not. During testing, the AED searches the audio stream for a hit. If a hit is found, the neighbouring frames in the visual stream are queried by the VED on whether they contain a hit or not. If no hit is detected in the visual stream in the temporal neighbourhood of the detected hit in the audio stream, an A/V sync error is flagged.

[*] The first author worked on this during his internship at Amazon. His current affiliation is with the University of Texas at Austin.

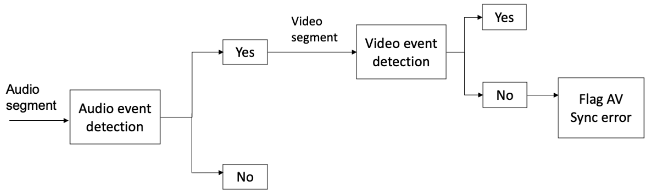

Fig. 1. Block diagram of the algorithm. The AED searches for the ball being hit in the audio stream. If it finds the event, the frames in the neighboring interval of time in the video stream are queried by the VED. If the event is not found in those frames, an A/V sync error is flagged.

2. DATABASE CREATION

Four videos of tennis events held in different courts were selected from the Amazon Prime Video catalog. Two videos were from WTA matches, and the other two videos were from ATP tours. The combined videos had a total duration of 6 hours and 504,300 frames. We did not remove advertisements and kept shots of the crowd and replays as well, since we wanted to build a model that could generalize well. All videos were of 25 fps. We created a graphical user interface to label each frame as a “hit” or a “bounce” or “neither”. Frames of replays were labeled as “neither” since the audio stream during replays generally contains commentary and not the audio of the play. Two labelers and an arbitrator labeled the frames.

The videos did not have A/V sync errors and hence the video labels were used to label the audio stream as well. The audio was captured at a sampling frequency of 48kHz in AAC-LC stereo. In order to account for the fact that the events we are trying to identify occur over a period of time, groups of three frames were considered as a single input and labeled with the label of the first of the three frames. At 25 fps, this corresponds to an interval between the first and third frames of 80ms. Audio segments of length 160ms were used as inputs to the AED, which at 48 kHz corresponds to 7680 pulse-code modulated (PCM) data points. Each segment was labeled by the label of the video frame that marked its beginning. Out of the 504,300 frames collected in this way, only 2443 frames were hits. There was thus a data imbalance of approximately 1:200. 20% of the data was set aside for testing. 80% of the data was used for training and validation, out of which 80% was used for training and the rest for validation.

3. METHODOLOGY

The method that we propose to detect A/V sync errors in tennis videos is shown in Fig. 1. In the following section, we describe the architecture and design of the audio event detector and the video event detector.

3.1. Audio Event Detector

3.1.1. Audio processing

Each audio segment is of length 7680 audio data points with a sampling frequency of 48 kHz. The mel-frequency cepstral coefficients are extracted from each segment. 2048 audio data points are used for each window, and windows are spaced by 128 audio data points. 61 MFCCs were computed for each window, so that the data formed a 61×60 input frame. The first and second temporal derivatives of the MFCCs were also computed. The input to the AED was thus a image of dimensions $61 \times 60 \times 3$.

3.1.2. Training

A ResNet [8] pre-trained on the ImageNet [9] database was used as the audio event detector. Training CNNs on MFCCs or log-mel-spectrograms has been shown to produce state-of-the-art results in audio processing for a number of applications [10, 11, 12, 13]. Pretraining CNNs on images has been shown to transfer well to audio tasks and has been used to establish benchmarks in a number of audio datasets [14, 15, 16, 17, 18]. Segments of audio were fed to the network and trained against a binary decision of “hit/not a hit”. The binary cross entropy loss was used. Early stopping was implemented by stopping the training process when the precision on the validation data dropped for three consecutive epochs. High precision is desirable for the audio event detector because if a false positive “hit” is detected by the network when there is actually no hit present, and the VED searches for a hit and does not find it, a false AV sync flag is raised. In our application scenario, operators have to manually check the stream if a sync error is raised, and frequent false flags can cause fatigue and irritation. Besides this, hits occur at a high frequency during rallies, and missing a few hits is therefore tolerable as long as a sufficient number of hits are detected within the interval of play. The network is trained on the entire training set. The input segments are separated by 40 ms, which is the time interval between two frames in the visual stream and is thus the smallest interval at which events can be detected. The audio segment is of length 160 ms because we found that the sound of some hits sometimes extends to that amount of time. However, because of the large amount of overlap between adjacent windows, we also found that a number of false positives were adjacent (in time) to false negatives during evaluation. We discuss how we resolved this in section III.

3.2. Video Event Detector

3.2.1. Video Preprocessing

The video frames were passed through the pose detector proposed in [19]. The pose detector marks the position of joints and also provides a bounding box for persons in the frame. The scenes that are found in the broadcast video are diverse

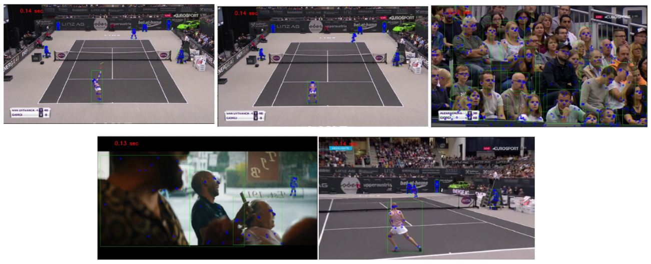

Fig. 2. Representative frames from the database after bounding box and pose detection are applied.

and include shots of the crowd, advertisements, replays, etc. It is therefore impossible to predict the number of persons that can be in a frame, and so the frames with the bounding boxes are passed as-is to the subsequent network. Scenes from a video with the bounding boxes and pose overlaid are shown in Fig.2. The frames were then resized to size 960 × 540 to keep computational costs reasonable and normalized before passing them to the network. Data augmentation was performed by randomly flipping the frames horizontally. Three frames are sent to the network at a time, and all three would have the label of the first frame in that group.

3.2.2. *Training*

We use a mixed-convolution network [20] for the video event detector. The first 6 layers of the network are from the C3D [21] network pre-trained on UCF101 [22]. At the end of 6 layers, the temporal dimension is compressed completely and subsequent layers are 2D convolutional residual blocks. Three 2D convolutional residual blocks of 512 layers each are used after the initial 3D convolutional layers taken from C3D.

We found that the video network could not learn to identify tennis hits from the video if it was trained on the whole dataset. Training typically did not converge well because of the massive data imbalance. This was probably because the visual signature of the ball being hit occupies a small part of the frame (spatially) and the rest of the frame appears similar to many other frames in the video where the ball is in play but not being hit. The features and patterns during ball hits in video frames are not as prominent as those in audio data, where a clear difference in features exists between “hits” and “not hits”. For example, when players are running in the tennis court whether there is a ball hit or not, those (many) frames look very similar to each other, whereas in the audio stream the sound of the ball being hit is distinctly heard over the background noise of play. The other difficulty is that we are targeting identification within a very small window in time (3 frames or 80 ms). Most video activity recognition algorithms test over much larger temporal durations, generally 25 frames or above. The data imbalance also exacerbates the problem since the network can classify all frames as ”not a hit” and achieve a low and stagnant loss. Random undersampling, random oversampling, and loss reweighing did not work to solve the issue since in this case the evaluation set has the same data imbalance as the training set.

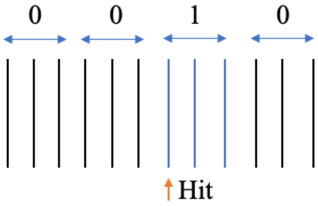

Fig. 3. Illustration of labeling and sample selection for the third stage of VED training. Each vertical line represents a video frame. Positive samples are blocks of 3 frames each starting from frames labeled as “hits”. Negative samples are blocks of 3 frames each that are immediately adjacent to positive samples from 6 frames behind to 6 frames ahead of the hit

We first tried training the VED against the *entire* database and using the VED in parallel with the AED, in order for the VED to detect hits separately from the AED and then correlate the hit detector outputs of the VED and AED across time. This was not feasible due to the aforementioned difficulties. Instead, we found that the AED could be used as a reliable filtering mechanism, and that training the VED to distinguish frames near the hit was sufficient. We trained the VED in three stages, following [3]. In the first stage, we train with randomly chosen negative (i.e. “not hits”) examples from the training set and with *all* the positive examples in the training set, such that the data is balanced. In the second stage, we train all the positives with harder negatives, which we define as blocks of frames corresponding to when the tennis ball is bouncing. Such frames always show the ball in play and the players in motion, which force the network to learn the features necessary to distinguish a hit from a scene when the ball is in play. In the third and final stage, we train with blocks of frames that are immediately adjacent to the positives. An example of negatives for the third stage of training is shown in Fig. 3.

The VED was trained with an early stopping mechanism. Training was stopped after recall on the validation set fell for three consecutive epochs. It is important to maintain high recall for the VED because if it fails to identify hits where there are actually hits, the final AV sync detector will raise a number of false flags, which is undesirable.

3.3. A/V synchronization error detector

We test blocks of three frames each, which at 25fps corresponds to 40 ms. We therefore search for hits in the video stream between 240 ms before the hit is detected in the audio,

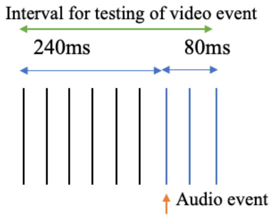

Fig. 4. Illustration of search space for the VED. If the AED detects a hit, the VED searches the frames in the video in the 6 frames preceding the time instance at which the audio event was detected and in the three frames afterwards.

and 80 ms after. An illustration is shown in Fig. 4. There are three non-overlapping groups of three frames each in this interval, and the VED searches each of these separately to find a hit. There are thus three predictions made by the VED for each audio segment identified as a hit by the AED, and if any of those predictions are a “hit”, no AV sync error is flagged.

4. EXPERIMENTS AND RESULTS

4.1. Hyperparameters

Both the AED and the VED were trained with the Adam optimizer with decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.99$. An exponentially decaying learning rate scheduler was used with a decay coefficient of 0.95 for both networks.

4.2. Audio Event Detector

The confusion matrix for the AED on the test set is shown in Table 1. We found that most false positives were immediately adjacent to false negatives. Out of the 145 false positives, 108 audio segments were immediately adjacent (i.e. within ± 40 ms) to false negatives. Most of these errors are caused by the fact that the auditory signature of hits extend over multiple segments. These errors are automatically corrected by the overall system for AV sync error detection because the VED searches for hits in the *neighborhood* of the time at which the hit was detected in the audio. Table 2 shows the confusion matrix for the AED, adjusted with adjacent false positive and false negative errors being treated as true positives or true negatives. With the adjusted result, the effective precision of the AED is 90.2% and the recall is 72.76%, which fully satisfies the requirements of our application scenario.

Table 1. Confusion matrix for AED

	Positives	Negatives
Predicted Positives	234	145
Predicted Negatives	236	100217

Table 2. Adjusted Confusion matrix for AED

	Positives	Negatives
Predicted Positives	342	37
Predicted Negatives	128	100325

4.3. AV Synchronization Error Simulation

The VED only evaluates the segments that are declared as hits by the AED. In order to simulate AV sync errors, we randomly introduced offsets between the video and audio for half of the samples that were predicted as positives by the AED. The offset was a random number of video frames selected from between -15 and $+15$, excluding the range from -3 to $+6$. The range from -3 frames to $+6$ frames was skipped because the frames in this interval would be searched for the video event.

4.4. AV Synchronization Error Detection

Table 3 shows the confusion matrix for the final task of AV sync error detection. The segments chosen by the AED (with errors simulated as described in the earlier subsection) were sent to the VED, which inspected the neighboring frames and made predictions accordingly. 100,832 audio segments were to be evaluated in the test set. There were 470 hits among these. Out of these, 379 were detected as hits by the AED. 342 of these are actually hits (after adjustment). Out of the 379, 155 examples were assigned an AV sync error by the random process described earlier. All 379 examples were queried by the VED. If no hit was found in the neighborhood of the chosen audio segment, an AV sync error was flagged. Precision obtained by the VED is 81.25% and the recall is 83.87%.

Table 3. Confusion matrix for AV Sync Error

	Positives	Negatives
Predicted Positives	130	30
Predicted Negatives	25	138

5. CONCLUSION

We proposed a novel method and created a large-scale database for detecting errors in A/V synchronization of tennis videos. This method is able to detect very small errors that are just outside the human threshold of perception and establishes a new state-of-the-art in a relatively unexplored area. We intend to explore alternative processing techniques for the audio and video streams to further advance the state-of-the-art. The excellent results that we have achieved give impetus for extending this idea to other sports and for high-frame-rate contents.

6. REFERENCES

[1] International Telecommunications Union, *ITU Rec. J.248*, 2008 (accessed September 23, 2020), https://www.itu.int/rec/T-REC-J.248/en.

[2] Junzhe Zhao, Bo Zhang, Zhaoyu Yan, Jing Wang, and Zesong Fei, "A study on the factors affecting audio-video subjective experience in virtual reality environments," in *Int. Conf. Virtual Reality Vis.*, 2017, pp. 303–306.

[3] Bruno Korbar, Du Tran, and Lorenzo Torresani, "Cooperative learning of audio and video models from self-supervised synchronization," in *Advances Neural Inf. Process. Syst.*, 2018, pp. 7763–7774.

[4] Naji Khosravan, Shervin Ardeshir, and Rohit Puri, "On attention modules for audio-visual synchronization.," in *IEEE Comp. Vis. Pattern Recognit. Workshop*, 2019, pp. 25–28.

[5] Joon Son Chung and Andrew Zisserman, "Out of time: automated lip sync in the wild," in *Asian Conf. Comp. Vis.* Springer, 2016, pp. 251–263.

[6] Dennis Laurijssen, Erik Verreycken, Inga Geipel, Walter Daems, Herbert Peremans, and Jan Steckel, "Low-cost synchronization of high-speed audio and video recordings in bio-acoustic experiments," *J. Exp. Biol.*, vol. 221, no. 4, 2018.

[7] Daniel G Baker and Thomas L Tucker, "Automated lip sync error correction," June 28 2005, US Patent 6,912,010.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[10] Dan Stowell, Michael D Wood, Hanna Pamuła, Yannis Stylianou, and Hervé Glotin, "Automatic acoustic detection of birds through deep learning: the first bird audio detection challenge," *Methods Ecology Evol.*, vol. 10, no. 3, pp. 368–380, 2019.

[11] Emre Cakir, Sharath Adavanne, Giambattista Parascandolo, Konstantinos Drossos, and Tuomas Virtanen, "Convolutional recurrent neural networks for bird audio detection," in *Eur. Signal Process. Conf.* IEEE, 2017, pp. 1744–1748.

[12] Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman, "Deep audio-visual speech recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, 2018.

[13] Hendrik Purwins, Bo Li, Tuomas Virtanen, Jan Schlüter, Shuo-Yiin Chang, and Tara Sainath, "Deep learning for audio signal processing," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 2, pp. 206–219, 2019.

[14] Kamalesh Palanisamy, Dipika Singhania, and Angela Yao, "Rethinking cnn models for audio classification," *arXiv preprint arXiv:2007.11154*, 2020.

[15] Grzegorz Gwardys and Daniel Michał Grzywczak, "Deep image features in music information retrieval," *Int. J. Electron. Telcommun.*, vol. 60, no. 4, pp. 321–326, 2014.

[16] Sainath Adapa, "Urban sound tagging using convolutional neural networks," *arXiv preprint arXiv:1909.12699*, 2019.

[17] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen, "Epic-fusion: Audio-visual temporal binding for egocentric action recognition," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2019, pp. 5492–5501.

[18] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel, "Esresnet: Environmental sound classification based on visual domain models," *arXiv preprint arXiv:2004.07301*, 2020.

[19] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang, "Deep high-resolution representation learning for human pose estimation," in *IEEE Conf. Comp. Vis. Pattern Recognit.*, 2019, pp. 5693–5703.

[20] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE Comp. Vis. Pattern Recognit.*, 2018, pp. 6450–6459.

[21] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2015, pp. 4489–4497.

[22] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.