Deep Unsupervised Multi-View Detection of Video Game Stream Highlights

Charles Ringer
Department of Computing
Goldsmiths, University of London
London, United Kingdom
c.ringer@gold.ac.uk

Mihalis A. Nicolaou
Department of Computing
Goldsmiths, University of London
London, United Kingdom
m.nicolaou@gold.ac.uk

ABSTRACT

We consider the problem of automatic highlight-detection in video game streams. Currently, the vast majority of highlight-detection systems for games are triggered by the occurrence of hard-coded game events (e.g., score change, end-game), while most advanced tools and techniques are based on detection of highlights via visual analysis of game footage. We argue that in the context of game streaming, events that may constitute highlights are not only dependent on game footage, but also on social signals that are conveyed by the streamer during the play session (e.g., when interacting with viewers, or when commenting and reacting to the game). In this light, we present a multi-view unsupervised deep learning methodology for novelty-based highlight detection. The method jointly analyses both game footage and social signals such as the players facial expressions and speech, and shows promising results for generating highlights on streams of popular games such as *Player Unknown's Battlegrounds*.

CCS CONCEPTS

• **Computing methodologies** → **Scene anomaly detection**; **Neural networks**; *Scene understanding*; *Video segmentation*;

KEYWORDS

Video game stream analysis, highlight detection, event detection

1 INTRODUCTION

Recently, live streaming services such as TWITCH.TV[1], Youtube Gaming[2], and Huya[3] have become popular platforms for video game players to broadcast themselves playing on the Internet. During a typical stream, players broadcast both game footage, as well as video of their face via a web-cam, while also communicating with viewers via audio and text chat.

In this work, we present the first, unsupervised, multi-modal, approach towards generating highlight clips, by analyzing both audio and video arising from the player's camera feed, as well as game footage (both video and audio), in order to identify novel events occurring during a stream. We use convolutional autoencoders for visual analysis of game scene and face, spectral features and component analysis for audio, while recurrent layers are utilized for fusing representations and eventually, detecting highlights on multi-view time-series data.

[1]www.twitch.tv
[2]gaming.youtube.com
[3]www.huya.com

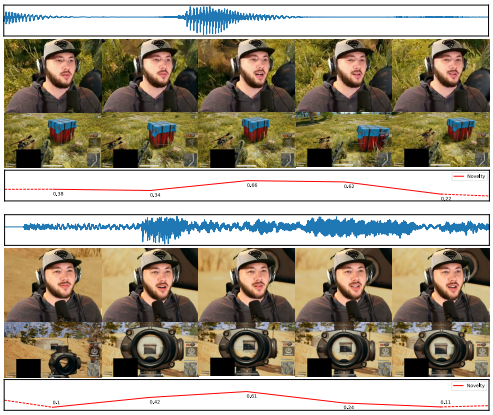


Figure 1: Example highlights. Audio waveform, face frames, game frames, and prediction error shown. Above: The streamer makes a joke after picking up a good item at a crate. Below: The streamer initiates a fire fight with another player. In both cases the highlight apex is the center frame.

2 RELATED WORK

2.1 Event and Highlight Detection

Detecting events in audio-visual data is an active area of research across a range of research domains. Perhaps the most pertinent to this study is Chu et al. [8, 9] who, studying *League of Legends* tournament streams, used in-game messages to select events and various motion based features, such as monitoring particle effects, to detect highlights.

Much event detection research has been focused on motion. Simonyan and Zisserman [27] and Feichtenhofer et al. [11] both utilize optical flow combined with object detection in order to detect actions performed by humans. Giannakopoulos et al. also considered motion in their work for the purpose of detecting violent scenes in films [13]. Xu et al. used unsupervised learning to detect events partly based on motion, when analysing scenes of pedestrians walking [31].

Sports is a popular domain for event detection research. Ren et al. studied highlight detection in soccer games, studying 4 matches [24] with good results, especially when detecting goals scored. Xu and Chua used not just audio-visual features but also external, text based, information in their work towards the detection of highlights in team sports [32]. A similar approach, applied to baseball games, is proposed by Chiu et al. [7]. Sun et al. in [29] analysed the excitement

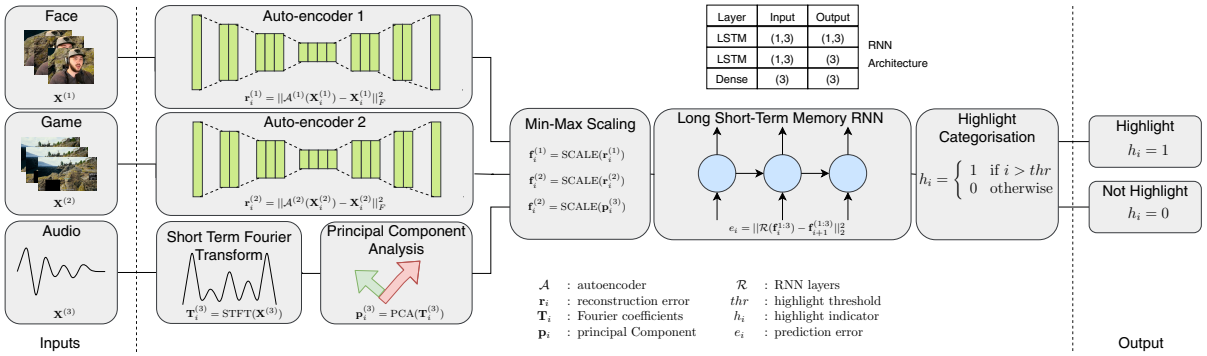


Figure 2: Overview of the system

level of sports commentators using audio features, mainly Mel Frequency Cepstrum Coefficients (MFCCs) and pitch data, to detect highlights. Nguyen and Yoshitaka [20] adopt a cinematography and motion based approach, whereby they analysed the type of camera shots used in order to detect highlights, especially emotional events.

We use a measure of novelty to identify salient points in a stream. Novelty detection, including reconstruction error based systems, has been used in a wide range of other domains. Pimentel et al.'s review of novelty detection [23] provides a comprehensive overview.

2.2 Emotion Detection

Studying streamers is, in part, the study of humans reacting to stimulus in an interactive setting. Therefore, whilst this work aims to develop event detection techniques, it is useful to consider work related to social and emotional signal processing, as it informs our approach. Related work includes research in analyzing player experience during gameplay. For example Karpouzis et al. developed the "Platformer Experience Dataset", that contains audio-visual material of subjects playing a platformer game, *Infinite Mario Bros* [17]. The dataset has been utilized in several studies. For example, Shaker et al. [26] develop player experience modeling techniques, while Asteriadis et al. [3] used this data set to develop techniques for clustering player types, with findings pointing to head movement being an indicator of player experience and skills.

Many affective computing techniques are related to those used in this study, for example EmoNets [16] use Convolutional Neural Networks for understanding facial cues. Ghosh et al. use Fourier Coefficients and MFCCs fed to a variety of autoencoders for learning affect from speech [12]. Similarly Amer et al. used extracted audio features, decomposed using Principal Component Analysis, as input to a selection of deep networks [2]. Busso et al. discuss frequency and emotion detection, confirming the result that pitch is an important indicator for emotion [5].

2.3 Video Game Scene Analysis

Little study has been undertaken into analyzing and extracting information from game scenes. The majority focuses on understanding the strategy, structure and physics of game worlds. For example, Guzdial and Riedl developed unsupervised techniques for building full game levels from observing gameplay videos [14]. Croxton and Kortemeyer studied the way players learnt about physics related game content through the study of game play videos [10]. Lewis et al. used *Starcraft* replays to discover strategies [19]. Alvernaz and Togelius used the latent space of an auto-encoder encoder to evolve agents for Visual Doom [1]. Similarly Rioult et al. used the players in-game positions to predict winners in *Defense of the Ancients* [25].

3 METHODOLOGY

3.1 Face and Game Scene Analysis

In our work, we utilize convolutional autoencoders for analyzing both player face and game footage. The networks are composed of two stacked VGG16-like networks [28], omitting the fully-connected layers. Given a video frame, the encoder produces a 512-filter encoding. The decoder is similar, employing reversed layers and up-sampling rather than max-pooling, reconstructing the input image. Each convolutional layer has a 3×3 filter window and each max-pooling layer uses 2×2 window with a stride of two, following works such as Deep Convolutional Auto-Encoder with Pooling - Unpooling layers, proposed by Turchenko et al. [30], and Stacked What-Where Auto-Encoders, proposed by Zhao et al. [34]. The network was trained using an `ADADELTA` optimiser [33] using `Tensorflow` and `Keras`. The reconstruction errors $R^{(1)}$ and $R^{(2)}$ (as shown in Fig. 2) were utilized as indicators of novelty. The complete network architecture is described in Table 1.

We note that for the face autoencoder, we used the `VGG` Face Descriptor weights [6, 22] for the encoder, which were frozen during training as no noticeable improvement was observed when fine-tuning and training end-to-end. The autoencoder for game footage was trained end-to-end for each video.

Once trained on frames from a video, the reconstruction error can be used as an indicator for novel frames in a video - which in the context of this work, we consider as proxies for highlights. More details on reconstruction-based novelty detection can be found in a recent survey by Primentel et al. [23].

3.2 Audio Stream Analysis

Since we are mostly interested in detecting arousal, we consider an approach that focuses on key audio frequencies. In order to do so, we firstly consider 400ms windows, with a sampling rate of 10 samples per second, thus having a 300ms overlap between

Layer	Input	Output
Conv2d	(224,224,3)	(224,224,64)
Conv2d	(224,224,64)	(224,224,64)
MaxPool	(224,224,64)	(112,112,64)
Conv2d	(112,112,64)	(112,112,128)
Conv2d	(112,112,128)	(112,112,128)
MaxPool	(112,112,128)	(56,56,128)
Conv2d	(56,56,128)	(56,56,256)
Conv2d	(56,56,256)	(56,56,256)
Conv2d	(56,56,256)	(56,56,256)
MaxPool	(56,56,256)	(28,28,256)
Conv2d	(28,28,256)	(28,28,512)
Conv2d	(28,28,512)	(28,28,512)
Conv2d	(28,28,512)	(28,28,512)
MaxPool	(14,14,512)	(14,14,512)
Conv2d	(14,14,512)	(14,14,512)
Conv2d	(14,14,512)	(14,14,512)
Conv2d	(14,14,512)	(14,14,512)
MaxPool	(14,14,512)	(7,7,512)

Layer	Input	Output
UpSample	(7,7,512)	(14,14,512)
Conv2d	(14,14,512)	(14,14,512)
Conv2d	(14,14,512)	(14,14,512)
Conv2d	(14,14,512)	(14,14,512)
UpSample	(14,14,512)	(28,28,512)
Conv2d	(28,28,512)	(28,28,512)
Conv2d	(28,28,512)	(28,28,512)
Conv2d	(28,28,512)	(28,28,512)
UpSample	(28,28,512)	(56,56,512)
Conv2d	(56,56,512)	(56,56,256)
Conv2d	(56,56,256)	(56,56,256)
Conv2d	(56,56,256)	(56,56,256)
UpSample	(56,56,512)	(112,112,512)
Conv2d	(112,112,256)	(112,112,128)
Conv2d	(112,112,128)	(112,112,128)
UpSample	(224,224,128)	(224,224,128)
Conv2d	(224,224,128)	(224,224,64)
Conv2d	(224,224,64)	(224,224,64)
Conv2d	(224,224,64)	(224,224,3)

Table 1: Auto-encoder layers. Left: encoder, right: decoder.

sequential windows. A large enough window was utilized in order to alleviate issues arising from asynchrony between audio and video modalities, due to the preservatory and anticipatory co-articulation phenomena entailing that audio cues are often delayed by around 0.12 seconds [18].

For each window, we compute the Short-Term Fourier Transform. We retain only the magnitudes (denoted as **T** in Figure 2), since they are deemed as good indicators for arousal [4].

Additionally, in order to isolate the streamers voice from the game sounds we discard frequencies which are uncommon in human speech[4]. Frequencies between 300 - 3400 hertz were retained, a range chosen because it is the standard frequency range for telephones calls. To capture variation while reducing dimensionality and noise, we apply Principal Component Analysis (PCA) and retain only the k first principal components.

3.3 Recurrent Layer for Late Fusion

A recurrent layer is utilized for indicating reconstruction-based novelty across all views and modalities, which in-turn can be considered as a proxy for detecting stream highlights. The recurrent layer is fed with time-series consisting of reconstruction errors from face and game scene autoencoders, as well as the extracted audio features, in-effect performing a multi-dimensional smoothing operation. Values are normalized between $[0, 1]$ to avoid biasing views that cover different ranges, while training entails forecasting values corresponding to the next frame, $t + 1$ at time t, thus effectively incorporating information from all views in a temporal late-fusion setting.

We utilize two Long Short-Term Memory (LSTM) layers [15], and a fully connected layer with a sigmoid activation. Each neuron in the LSTM layer retains a latent state capturing useful information from previous time frames, and along with the current input generates the output values. LSTMs also include a "forget gate" which allows the neuron to filter which input values are kept and which are discarded, improving training over a standard RNN layer. The network architecture is illustrated in Figure 2. We utilize the ADADELTA optimizer for training, employing a mean-squared error loss function. The architecture is similar to the one utilized by Malhotra et al. [21].

3.4 Highlight Detection as Reconstruction-based Novelty

We utilize the prediction error of the recurrent layer on a given stream session (including face, audio, and game footage) as an indicator of *stream highlights* (E_i in Figure 2). In more detail, we utilize a threshold, empirically determined as 0.01%, and classify the same percentage of frames with highest prediction error as *highlight frames*. We use the aforementioned highlight frames in order to generate the full highlight clips. To do so, we treat the detected frames as *apex frames* of a highlight event. We link proximal apex frames together[5], and similarly to the TWITCH.TV clip system, consider the highlight clip to be 10 seconds before the apex frame and 5 seconds after the last. In this way, we ensure that the appropriate context is included in the highlight clip, and that the clip is self-contained (e.g., a reaction of the streamer can be detected as a highlight frame, with a preceding game or stream event causing the reaction).

4 DATA

Data was gathered from TWITCH.TV. We recorded popular streamers playing *Player Unknown's Battlegrounds* (PUBG), a multi-player on-line battle royale game. A number of players are spawned simultaneously with a goal of exploring an island, collecting weapons, killing other players and ultimately being the last player alive. PUBG was chosen because it often has long periods of low intensity game-play and short bursts of concentrated action, making highlight detecting a worthwhile task, while an abundance of high-quality streams are available due to the popularity of the game.

Each recorded stream was segmented into videos spanning a single game, and downtime between games removed. It makes little sense to look for highlights when the game is not being played as streamers often take short breaks in-between games where they will leave the stream or browse social media etc. The data set consists of videos from two streamers, both male, one American and one German but streaming in English. There is a total of 11 videos and each video is between 19 minutes 30 seconds and 30 minutes 40 seconds long. In total, we utilize over 5 hours of stream footage.

We pre-process each video as follows. Firstly, we utilize a sampling rate of 10 frames per second, which is deemed sufficient for our task and makes training faster. We mask-out the players face for feeding the game footage into the respective autoencoder, while the cropped region including the players face is used for the face autoencoder. Finally, we resize each frame to $224 \times 224 \times 3$ in order to match the VGG Face Descriptor dimensions.

5 RESULTS

In total, we obtained 98 segmented highlight clips by applying our method on 11 game stream recordings. To evaluate our method, we manually annotated each highlight clip into 4 categories, namely "funny", "action", "social-interaction", and finally, "no highlight".

[4] A more eloquent approach could be based on methods such as Independent Component Analysis (ICA)

[5] Apex frames are linked if not doing so would cause an overlap between clips.

Video	Highlight				No Highlight
	Funny	Action	Interaction	Total	Total
S1_1	1	2	4	7	0
S1_2	0	1	1	2	4
S1_3	0	3	2	5	3
S1_4	2	2	4	8	2
S1_5	0	3	4	7	4
S1_6	0	1	1	2	2
S2_1	5	2	0	7	1
S2_2	3	1	2	6	1
S2_3	6	4	2	12	2
S2_4	3	1	3	7	1
S2_5	6	5	1	12	3
Total	26	25	24	75	23

Table 2: Generated highlight clips by category using all modalities and views.

Funny videos are streamer-focused events where the streamer makes a joke, laughs, or is in other ways amused. Action highlights stem from game-events (e.g., streamer engaged in a firefight). Highlights that are tagged as "social-interaction" include events that are community-led, where the streamer interacts with viewers in a meaningful way, e.g., thanking subscribers, answering questions, or reacting during a similar interaction. Note that "social-interaction" highlights are important for a compelling game stream, and are often found in streamer highlight clips that are manually segmented. Finally, clips containing no noteworthy events are labeled as "No Highlight". In Table 2, we show results by using all available modalities, where out of 75 clips with interesting content, 51 are tagged as"funny" or "action", with the remaining 24 labeled as "social-interaction".

5.1 Modalities and Highlight Detection

We evaluate the proposed architecture when observing different combinations of views and modalities, including "Face", "Game Footage", and "Audio", with results summarized In Table 3. Overall, we find that the model fusing all views and modalities performs the best. This is an expected result, since utilizing audio-visual information from both streamer behaviour and the game itself is deemed to provide a more informed approach. We also note that the number of detected highlights across all combinations is similar, with the exception of the "Audio Only" model, that produced considerably more. This is likely due to the impact of in-game audio (e.g., gunfire) that has not been entirely removed - further supported by the observation that 29% of highlights selected were action highlights, containing only a few funny or interaction clips. The "Face Only" stream is better at determining "funny" and "social-interaction" highlights, although has a worse precision in terms of action clips, which is expected given that using only the face means we do not consider any context regarding the game. "No Highlight" clips are comparable for the audio-only and face-only models, and are often due to unusual gestures that the streamer might perform (that can potentially occlude the face). This is expected since "face novelty" itself does not necessarily indicate a highlight.

The game-footage model is the worst highlight predictor, given our results. Whilst this might appear counter-intuitive, we can also

Modalities	No. Videos	Highlight				No Highlight
		Funny %	Action%	Interaction%	Total%	Total%
Face, Game, Audio	98	0.27	0.26	0.24	0.77	0.23
Face, Audio	95	0.22	0.23	0.28	0.74	0.26
Face Only	96	0.14	0.14	0.24	0.52	0.48
Game Only	94	0.04	0.18	0.07	0.29	0.70
Audio Only	126	0.08	0.29	0.18	0.56	0.44

Table 3: Summary comparison of highlight-detection over multiple views and modalities

see that using game footage correctly detects the vast majority of action highlights[6]. The lack of social- and game- context or scene understanding makes the problem of detecting when the game-scene is interesting to the viewer rather than merely anomalous more challenging.

Finally, we present results using a face and audio model. This approach provides better results than all single-view models, and slightly worse results compared to using all available views and modalities. This finding suggests that whilst the game alone is a poor indicated of highlights (other than action clips), it can be useful to corroborate information extracted from other modalities, and improve performance.

5.2 Highlights over Time

We observe that the number of "No Highlight" segments is reduced over time, as shown in Figure 4. This points to the conclusion that the later in the video a highlight is detected, the more likely it is to be interesting. In more detail, 61% of "No Highlight" results occur in the first 30% of a video, opposed to 19% of funny clips, 4% off action clips and 41% of interaction clips. Furthermore, the majority of action clips, 92%, and funny clips, 69%, occur in the last 50% of the video, opposed to only 22% of "No Highlights" clips. In fact 60% of action clips occur in the last 20% of the video duration. By considering only detections in the last half of each video, we find that 91% of clips are interesting in some way.

Based on our observations of the streams, we can attribute this to several reasons that are mostly related with game design. Firstly, the game is designed in such way the the play area shrinks over time, in a way that forces interaction between players towards the end of a game, hence the larger amount of action highlights towards the end of the video. Secondly, there are fewer viewer interactions as the game progresses, since the game intensity increases and players are required to focus more on the game.

5.3 Novelty Across Modalities

In this section, we discuss the detection of novel events across modalities. In Figure 5, we show (a) the RNN prediction error fusing all modalities and views, (b,c) the face and game footage autoencoder reconstruction errors, and (d) the first principal component of the Fourier coefficients of the audio channel. We plot the errors over time for a particular video S1_1, while coloring errors that correspond to selected highlight frames in red. In general, we can observe that for face and game, sharp "spikes" pointing to highlights can be clearly observed in the distribution, with errors on

[6]Although it is possible that on-screen events are indicators of other highlight types, for example new subscriber pop-ups or humorous on-screen events.

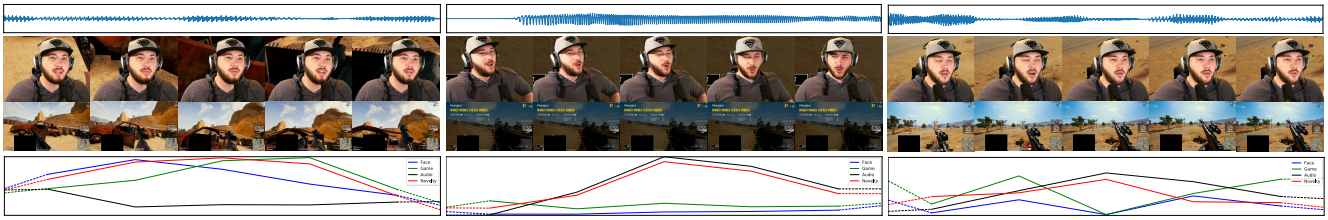


Figure 3: Example highlights discovered by the proposed method. Left: The streamer begins aiming their rifle which changes the game scene enough to trigger a highlight, in agreement with a change in the facial expression. Center: The streamer wins a game and shouts in celebration; detected by indicating novelty in the audio features. Right: During a firefight, gunfire causes a spike in the audio which triggers a highlight.

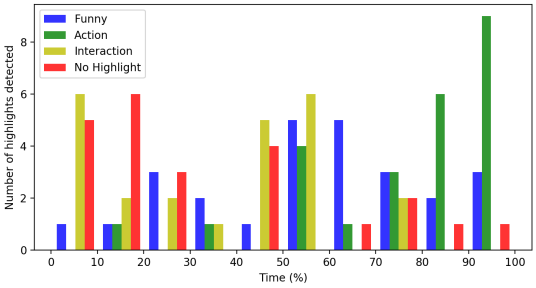
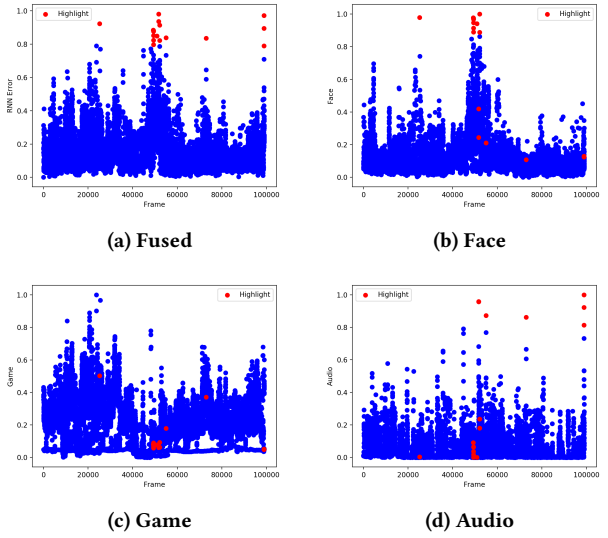


Figure 4: Highlights by type over time



(a) Fused **(b) Face**

(c) Game **(d) Audio**

Figure 5: Errors over time indicating novel events for a particular video (S1_1). (a) Fused prediction error. (b) Face video reconstruction error. (c) Game footage reconstruction error. (d) Audio features over time.

game footage being less clear, showing a higher average and wider spread, likely due to the lack of a baseline/context. For this particular stream, we can also observe that the game footage impacts the final highlight frames *less* than the face and audio views, where clear spikes are transferred to the fused results. Observing the final results, we can see that spikes in only one view are smoothed out, while spikes appearing in more than one views are accentuated.

6 CONCLUSIONS

We presented an unsupervised deep learning architecture for detection of highlight clips based on audio-visual data, broadcasted during a typical game stream. We consider a measure of reconstruction-based novelty as a proxy for indicating highlights, while *jointly* analyzing facial footage of the player, footage of the streamed game, as well as audio. We discuss several insights arising from our analysis, while we show that the proposed method is successful in terms of detecting both social and game-related highlights in video game streams, further pinpointing the significance of considering social signals towards detecting interesting highlights in game streams. Future works into this domain would include widening the study to a more streamers playing a wide range of games.

ACKNOWLEDGMENTS

This work was supported by EPSRC grant EP/L015846/1 (IGGI). We thank the NVIDIA Corporation for providing a Titan X Pascal GPU used in this work. The authors would also like to thanks Mats Kathage (P4wnyhof) for permission to present footage from his stream in this work.

REFERENCES

[1] S. Alvernaz and J. Togelius. 2017. Autoencoder-augmented neuroevolution for visual doom playing. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*. 1–8. DOI: http://dx.doi.org/10.1109/CIG.2017.8080408

[2] M. R. Amer, B. Siddiquie, C. Richey, and A. Divakaran. 2014. Emotion detection in speech using deep networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 3724–3728. DOI: http://dx.doi.org/10.1109/ICASSP.2014.6854297

[3] Stylianos Asteriadis, Kostas Karpouzis, Noor Shaker, and Georgios N. Yannakakis. 2012. Towards detecting clusters of players using visual and gameplay behavioral cues. *Procedia Computer Science* 15 (2012), 140–147. DOI: http://dx.doi.org/10.1016/j.procs.2012.10.065

[4] Jo-Anne Bachorowski. 1999. Vocal Expression and Perception of Emotion. *Current Directions in Psychological Science* 8, 2 (1999), 53–57. DOI: http://dx.doi.org/10.1111/1467-8721.00013

[5] Carlos Busso, Sungbok Lee, and Shrikanth Narayanan. 2009. Analysis of emotionally salient aspects of fundamental frequency for emotion detection. *IEEE*

Transactions on Audio, Speech and Language Processing 17, 4 (2009), 582–596. DOI: http://dx.doi.org/10.1109/TASL.2008.2009578

[6] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. 2017. VGGFace2: A dataset for recognising faces across pose and age. (2017). http://arxiv.org/abs/1710.08092

[7] Chih Yi Chiu, Po Chih Lin, Sheng Yang Li, Tsung Han Tsai, and Yu Lung Tsai. 2012. Tagging webcast text in baseball videos by video segmentation and text alignment. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 7 (2012), 999–1013. DOI: http://dx.doi.org/10.1109/TCSVT.2012.2189478

[8] Wei Ta Chu and Yung Chieh Chou. 2015. Event detection and highlight detection of broadcasted game videos. *HCMC 2015 - Proceedings of the 2nd Workshop on Computational Models of Social Interactions: Human-Computer-Media Communication, co-located with ACM MM 2015* (2015), 1–8. DOI: http://dx.doi.org/10.1145/2810397.2810398

[9] Wei Ta Chu and Yung Chieh Chou. 2017. On broadcasted game video analysis: event detection, highlight detection, and highlight forecast. *Multimedia Tools and Applications* 76, 7 (2017), 9735–9758. DOI: http://dx.doi.org/10.1007/s11042-016-3577-x

[10] DeVaughn Croxton and Gerd Kortemeyer. 2018. Informal physics learning from video games: a case study using gameplay videos. *Physics Education* 53, 1 (2018), 015012. http://stacks.iop.org/0031-9120/53/i=1/a=015012

[11] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. 2016. Convolutional Two-Stream Network Fusion for Video Action Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[12] Sayan Ghosh, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer. 2015. Learning Representations of Affect from Speech. (2015), 1–10. http://arxiv.org/abs/1511.04747

[13] Theodoros Giannakopoulos, Alexandros Makris, Dimitrios Kosmopoulos, Stavros Perantonis, and Sergios Theodoridis. 2010. Audio-Visual Fusion for Detecting Violent Scenes in Videos. In *Artificial Intelligence: Theories, Models and Applications*, Stasinos Konstantopoulos, Stavros Perantonis, Vangelis Karkaletsis, Constantine D. Spyropoulos, and George Vouros (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 91–100.

[14] Matthew Guzdial and Mark Riedl. 2016. Game Level Generation from Gameplay Videos. (2016).

[15] Sepp Hochreiter and JÃijrgen Schmidhuber. 1997. Long Short-term Memory. 9 (12 1997), 1735–80.

[16] Samira Ebrahimi Kahou, Xavier Bouthillier, Pascal Lamblin, Caglar Gulcehre, Vincent Michalski, Kishore Konda, Sébastien Jean, Pierre Froumenty, Yann Dauphin, Nicolas Boulanger-Lewandowski, Raul Chandias Ferrari, Mehdi Mirza, David Warde-Farley, Aaron Courville, Pascal Vincent, Roland Memisevic, Christopher Pal, and Yoshua Bengio. 2015. EmoNets: Multimodal deep learning approaches for emotion recognition in video. 1 (2015). DOI: http://dx.doi.org/10.1007/s12193-015-0195-2

[17] K. Karpouzis, G. N. Yannakakis, N. Shaker, and S. Asteriadis. 2015. The platformer experience dataset. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*. 712–718. DOI: http://dx.doi.org/10.1109/ACII.2015.7344647

[18] Aggelos K. Katsaggelos, Sara Bahaadini, and Rafael Molina. 2015. Audiovisual Fusion: Challenges and New Approaches. *Proc. IEEE* 103, 9 (2015), 1635–1653. DOI: http://dx.doi.org/10.1109/JPROC.2015.2459017

[19] Jm Lewis, Patrick Trinh, and David Kirsh. 2011. A corpus analysis of strategy video game play in starcraft: Brood war. *Proceedings of the 33rd Annual Conference of the Cognitive Science Society* (2011), 687–692. http://mindmodeling.org/cogsci2011/papers/0138/paper0138.pdf

[20] Ngoc Nguyen and Atsuo Yoshitaka. 2014. Soccer video summarization based on cinematography and motion analysis. *Multimedia Signal Processing (MMSP)* (2014), 1–6. http://ieeexplore.ieee.org/xpls/abs

[21] Gautam Shroff Pankaj Malhotra, Lovekesh Vig and Puneet Agarwal1. 2015. Long Short Term Memory Networks for Anomaly Detection in Time Series. In *23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*.

[22] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. 2015. Deep Face Recognition. *Procedings of the British Machine Vision Conference 2015* Section 3 (2015), 41.1–41.12. DOI: http://dx.doi.org/10.5244/C.29.41

[23] Marco A.F. Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko. 2014. A review of novelty detection. *Signal Processing* 99 (2014), 215–249. DOI: http://dx.doi.org/10.1016/j.sigpro.2013.12.026

[24] Reede Ren, Jeomon Jose, and He Yin. 2007. Affective sports highlight detection. *European Signal Processing Conference* Eusipco (2007), 728–732.

[25] François Rioult, Jean-Philippe Métivier, Boris Helleu, Nicolas Scelles, and Christophe Durand. 2014. Mining Tracks of Competitive Video Games. *AASRI Procedia* 8, Secs (2014), 82–87. DOI: http://dx.doi.org/10.1016/j.aasri.2014.08.014

[26] Noor Shaker, Stylianos Asteriadis, Georgios N. Yannakakis, and Kostas Karpouzis. 2013. Fusing visual and behavioral cues for modeling user experience in games. *IEEE Transactions on Cybernetics* 43, 6 (2013), 1519–1531. DOI: http://dx.doi.org/10.1109/TCYB.2013.2271738

[27] K. Simonyan and A. Zisserman. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems*.

[28] K. Simonyan and A. Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.

[29] Yi Sun, Zhijian Ou, Wei Hu, and Yimin Zhang. 2010. Excited commentator speech detection with unsupervised model adaptation for soccer highlight extraction. *ICALIP 2010 - 2010 International Conference on Audio, Language and Image Processing, Proceedings* (2010), 747–751. DOI: http://dx.doi.org/10.1109/ICALIP.2010.5685077

[30] Volodymyr Turchenko, Eric Chalmers, and Artur Luczak. 2017. A Deep Convolutional Auto-Encoder with Pooling - Unpooling Layers in Caffe. (2017), 1–21. http://arxiv.org/abs/1701.04949

[31] Dan Xu, Elisa Ricci, Yan Yan, Jingkuan Song, and Nicu Sebe. 2015. Learning Deep Representations of Appearance and Motion for Anomalous Event Detection. (2015). DOI: http://dx.doi.org/10.5244/C.29.8

[32] Huaxin Xu and Tat-Seng Chua. 2006. Fusion of AV features and external information sources for event detection in team sports video. *ACM Transactions on Multimedia Computing, Communications, and Applications* 2, 1 (2006), 44–67. DOI: http://dx.doi.org/10.1145/1126004.1126007

[33] Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. (2012). http://arxiv.org/abs/1212.5701

[34] Junbo Zhao, Michael Mathieu, Ross Goroshin, and Yann LeCun. 2015. Stacked What-Where Auto-encoders. 1, i (2015), 1–12. http://arxiv.org/abs/1506.02351