Deep Unsupervised Multi-View Detection of Video Game Stream Highlights

unsupervised deep learning을 통해서 게임 장면 뿐만 아니라 게임 참여자들의 표정, 말투 같은 social signals를 이용해서 하이라이트를 추출.

게임 하이라이트를 추출하기 위해서 플레이어의 표정, 오디오, 게임 장면을 이용.

플레이어의 표정, 게임 장면 등을 분석 => convolutional autoencoder을 사용.

Audio 분석 => spectral features & component analysis 사용.

Face & game scene analysis (autoencoder)

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.

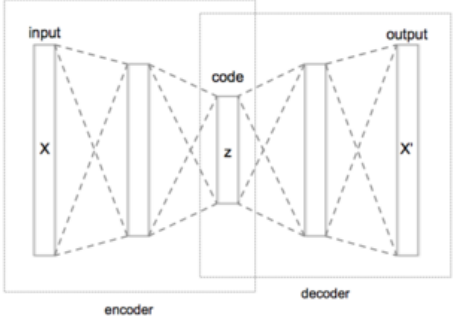
오토인코더 복습

차원 축소 등을 위해 representation learning 또는 feature learning을 비지도 학습의 형태로 학습하는 신경망.

Encoder: 입력 값을 받아 특징 값으로 변화

Decoder: 특징 값을 출력 값으로 변환 (latent feature 가 정말로 잘 추출 되었는지를 알기 위해 decoder를 통해서 다시 출력 값(입력값)을 확인)

Latent feature(z) 신경망 내부에서 추출된 특징적인 값들



Audio stream 분석

- Audio frequency를 이용.

- 400ms windows 를 사용 => audio, video modalities의 asynchrony 를 완화하기 위해서

- 각 window에 대해 Short-Term Fourier Transform 수행 (https://en.wikipedia.org/wiki/Short-time_Fourier_transform참고)

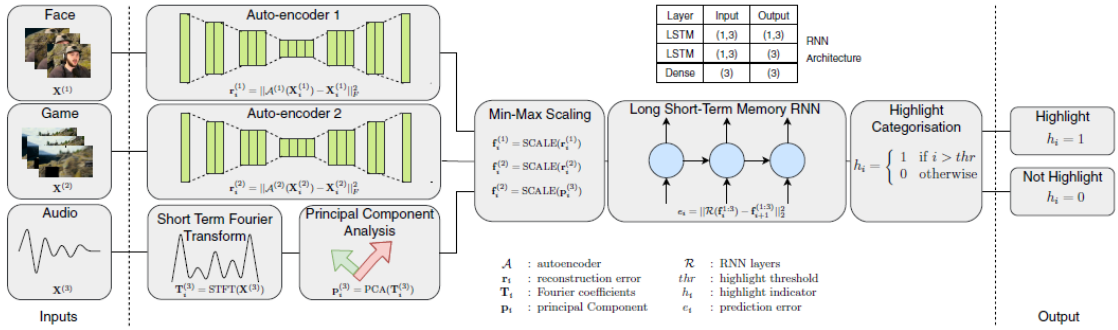
- 플레이어의 목소리와 게임 소리를 분리하기 위해서 사람 목소리가 아닌 주파수를 버림(300 ~ 3400hertz 만 남김)

Recurrent layer for late fusion

- 위에 분석한 것들을 융합 + 다중 뷰 시계열 데이터에서 하이라이트를 감지하는 계층

- Autoencoder에서 나온 결과와 오디오의 결과가 시계열로 input

- 2개의 LSTM layer + 1개의 fully connected layer (with sigmoid activation)



Recurrent 계층의 prediction error를 사용.

⇨ utilize a threshold, empirically determined as 0.01%, and classify the same percentage of frames with highest prediction error as highlight frames

data : battleground 게임, twitch tv에서 추출

1초에 10 frame을 샘플링 함, 게임 이미지와 플레이어의 얼굴을 따로 뽑아서 각각 autoencoder에 넣음.

결과

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.

여기서 action은 게임 이벤트(ex 총격전),

Interaction은 커뮤니케이션(ex 시청자와 소통), no highlight는 주목할 만한 이벤트

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

위 표에서 볼 수 있듯이 face, game, audio를 합친 것이 가장 잘 작동한다.