AI 기반 자동 편집 기술 동향

□ 홍순기 / ㈜에스비에스

Ⅰ. 개 요

최근 미디어 소비에 대한 주도권은 방송사에서 시청자에게 넘어갔다. 시청자들은 원하는 미디어를 원하는 시간에 소비하고자 한다. 변화된 시청자의 요구에 대응하기 위해 미디어 서비스 업체들은 호흡이 긴 동영상을 짧은 길이의 동영상으로 편집하여 제공하는 클립형 미디어 서비스를 제공하고 있다. 이러한 클립형 미디어 서비스를 위해서는 동영상 편집이 필수적으로 필요한데, 현재 대부분의 서비스 업체들은 동영상 편집을 수동으로 진행하고 있다. 따라서 수동으로 동영상을 편집하는데 소요되는 경제적/시간적 비용을 줄이기 위해 예전부터 다양한 동영상 자동 편집 기술이 시도되었다[1][2]. 비교적 최근까지 연구된 자동 편집 기술은 이미지 인식/분석 기술을 활용하여 이미지의 품질을 다양한 관점으로 수치화하고, 선형 예측 기술을 통해 수치화된 이미지 품질들로부터 해당 이미지의 중요도를 추정한 후, 목표 시간에 맞추어 중요도가 낮은 프레임을 소거하는 방법을 사용하였다. 이러한 방법을 통해 동영상 편집의 자동화 가능성을 확인할 수 있었으나, 이미지 인식/분석 기술의 한계에 의해 자동 생성된 편집 결과물을 바로 클립형 미디어 서비스에 적용할 수 있을 정도의 정확도를 확보하지는 못하였다. 하지만 최근에 이미지 인식/분석 능력을 비약적으로 발전시킨 딥러닝 기술을 동영상 자동 편집 기술에 적용하여 자동 생성된 축약 결과물의 정확성을 크게 향상시킨 기술들이 제안되었다.

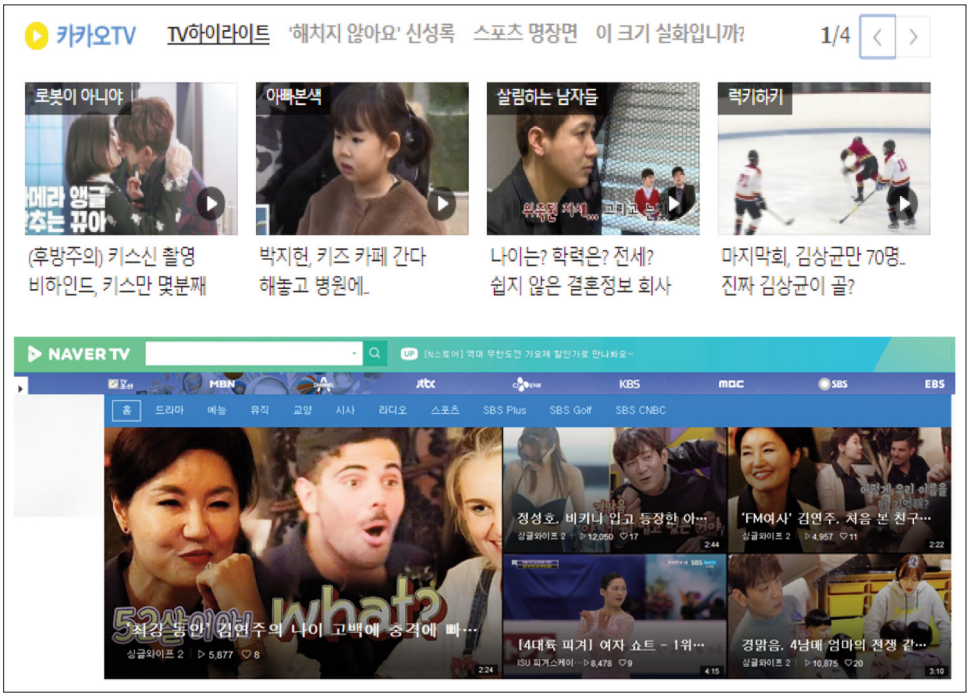
이에 본 기고에서는 클립형 미디어 서비스의 개념과 국내 서비스 동향을 소개하고, 클립형 서비스를 위한 동영상 자동 편집 기술의 발전 과정에 대해 살펴보려 한다. 마지막으로 동영상 자동 편집이 극복해야 하는 근본적인 문제점에 대해 논의하면서 글을 마무리 하려고 한다.

Ⅱ. 클립형 미디어 서비스 현재와 미래

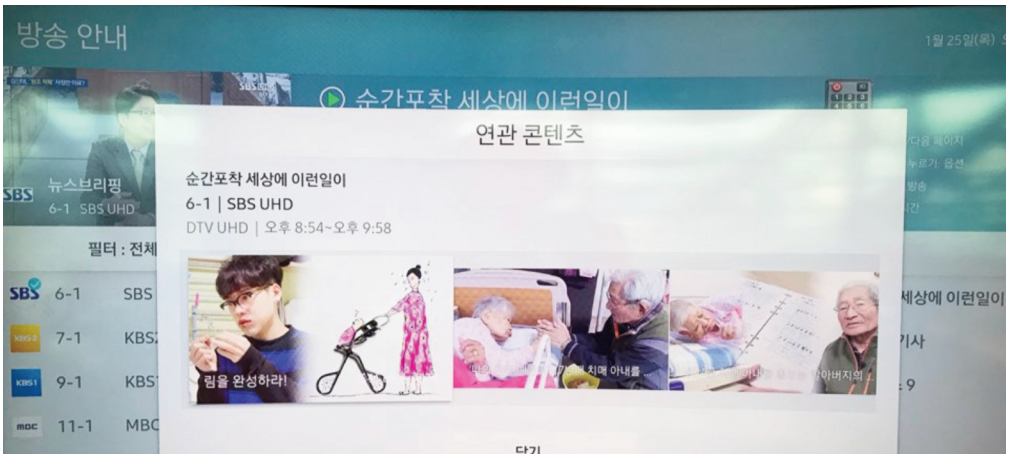
동영상을 시청하는 단말이 고정형 TV에서 이동형 모바일로 바뀌면서, 시청자들은 원하는 시간에 원하는 콘텐츠를 소비하고자 하는 욕구가 증가했다. 변화된 시청자의 욕구를 충족시키기 위해 방송사들은 스마트미디어렙(SMR)을 통해 방송사의 콘텐츠를 짧은 동영상 클립으로 만들어서 포탈에 공급하고 있다. 포탈을 통한 방송사 동영상의 클립형 미디어 서비스는 이미 일반화되어 있으며, 방송사 콘텐츠에 대한 홍보뿐만 아니라 프리롤 광고를 통한 광고 수익도 얻고 있다. 포털을 통한 클립형 미디어 서비스의 성공 이후로, 각 방송사들은 페이스북 등의 SNS에 각 사의 채널을 개설한 후 독자적인 클립형 미디어 서비스를 제공하고 있다. 이렇듯 클립형 미디어 서비스는 시대적인 요구에 의해 점점 확대되고 있으며, 이는 2017년 5월 방송을 시작한 지상파 UHDTV 서비스에서도 예외일 수 없다.

지상파 UHDTV는 ATSC3.0 표준을 기반으로 하고 있으며, 이는 방송 통신을 융합한 새로운 양방향 서비스를 구현할 수 있는 가능성을 열었다. 특히 클립형 미디어 서비스와 관련하여, 지상파 방송 3사는 양방향 방송 안내(Advanced ESG) 서비스를 제공하고 있다. 기존 TV 및 유료 방송 사업자를 통해 제공되고 있는 "방송 안내" 기능에 그치지 않고, 시청자가 원하는 프로그램에 대한 상세 정보 및 썸네일뿐만 아니라 하이라이트 영상, 예고 방송 등의 클립형 미디어를 제공받을 수 있다.

또한 지상파 방송 3사는 지상파 UHD 채널과 온



<그림 1> 포탈을 통한 클립형 미디어 서비스



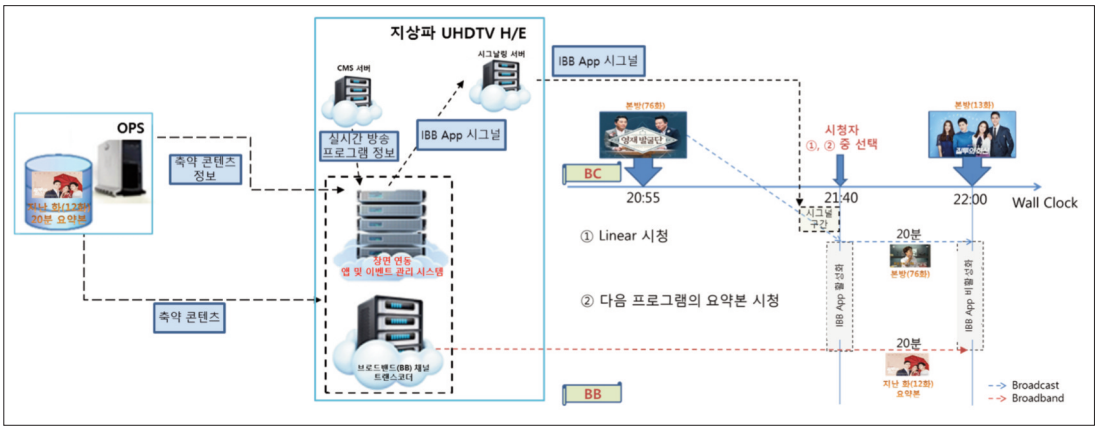
<그림 2> 지상파 UHDTV A-ESG 서비스

<그림 3> 지상파 UHDTV 티비바(TIVIVA) 서비스

라인 동영상 서비스가 결합된 세계 최초 지상파 UHD 양방향 서비스인 티비바(TIVIVA) 서비스를 런칭하였다. 티비바 서비스를 통해 시청자는 지상파 UHD 방송을 보다가 언제든지 다시 보기(VOD)나 관련 영상 등 시청자가 원하는 서비스를 선택하여 시청할 수 있다. 〈그림 3〉의 왼쪽 그림에서 볼 수 있듯이, 시청자가 지상파 UHDTV 방송에 진입하면 안테나를 통해 전달된 시그널을 해석하여 TV에서 실시간 방송 하단에 실시간 영상과 관련된 클립형 미디어와 VOD를 관람할 수 있는 티비바 홈(〈그림 3〉의 오른쪽 그림) 링크를 포함하고 있는 티비바 미니런처를 실행하여 화면에 오버레이 해준다.

앞서 살펴본 바와 같이, 클립형 미디어 서비스는 현재 방송 서비스 체계에서도 이미 중요한 위치를 차지하고 있지만, 앞으로 펼쳐질 지상파 UHDTV 방송 체계를 기반으로 한 신규 방송 서비스에서도 매우 중요한 역할을 할 것으로 예상된다.

IBB 표준은 지상파 UHD 방송을 시청하면서 방송망 및 브로드밴드 망을 통해 웹 기반의 서비스 애



<그림 4> 미래의 시청자 선택형 클립 미디어 서비스

플리케이션을 제공하는데 필요한 방법을 정의하고 있다. 따라서 지상파 UHD 표준, IBB 표준 그리고 동영상 자동 축약 기술을 활용하여 〈그림 4〉와 같은 지상파 UHD 방송 규격과 연계된 시청자 선택형 클립 미디어 서비스를 제공할 수도 있을 것이다. 자동 편집 시스템은 방송사의 온라인 배포 시스템으로부터 드라마 12화의 콘텐츠와 콘텐츠 정보를 수신하여 자동으로 동영상을 축약한 후, 축약된 콘텐츠 정보를 장면 연동 이벤트 관리 시스템으로 송부한다. 장면 연동 이벤트 관리 시스템은 실시간 방송 프로그램 정보와 축약된 동영상의 길이에 맞추어 드라마 13화가 방송되기 전에 시청자 TV로 웹 기반 서비스 애플리케이션을 포함한 IBB App 신호를 전송한다. IBB App 신호를 수신한 시청자는 (1) Linear 방송을 선택하여 현재 방송 중인 예능 76회를 그대로 시청할 수도 있고, (2) 다음 프로그램의 요약본 시청을 선택하여 드라마 12화 요약본을 시청할 수도 있다. 즉, 시청자에게 시청자의 선택에 따라 시청자가 원하는 유형의 방송을 시청할 수 있는 선택권을 부여하는 신규 클립형 서비스를 제공할 수 있다.

물론 상기한 신규 클립형 서비스를 실제로 제공하기 위해서는 수신한 IBB App 신호를 단말에서 처리할 수 있는 IBB 표준 개정이 방송사와 단말 제조사 간의 협의를 통해 도출되어야 하고, 단말에서 IBB App 신호에 포함되어 있는 정확한 타이밍에 본 방송을 클립 미디어로 대체할 수 있는지 검증하는 과정이 필요하다. 하지만 이러한 문제점은 방송사와 단말 제조사의 협력에 의해 해결될 것으로 보이며, 상기한 서비스에 한정되지 않고 지상파 UHD 방송 규격을 활용한 시청자 선택형 클립 미디어 서비스는 필연적으로 가까운 미래에 선보여질 것으로 예측된다.

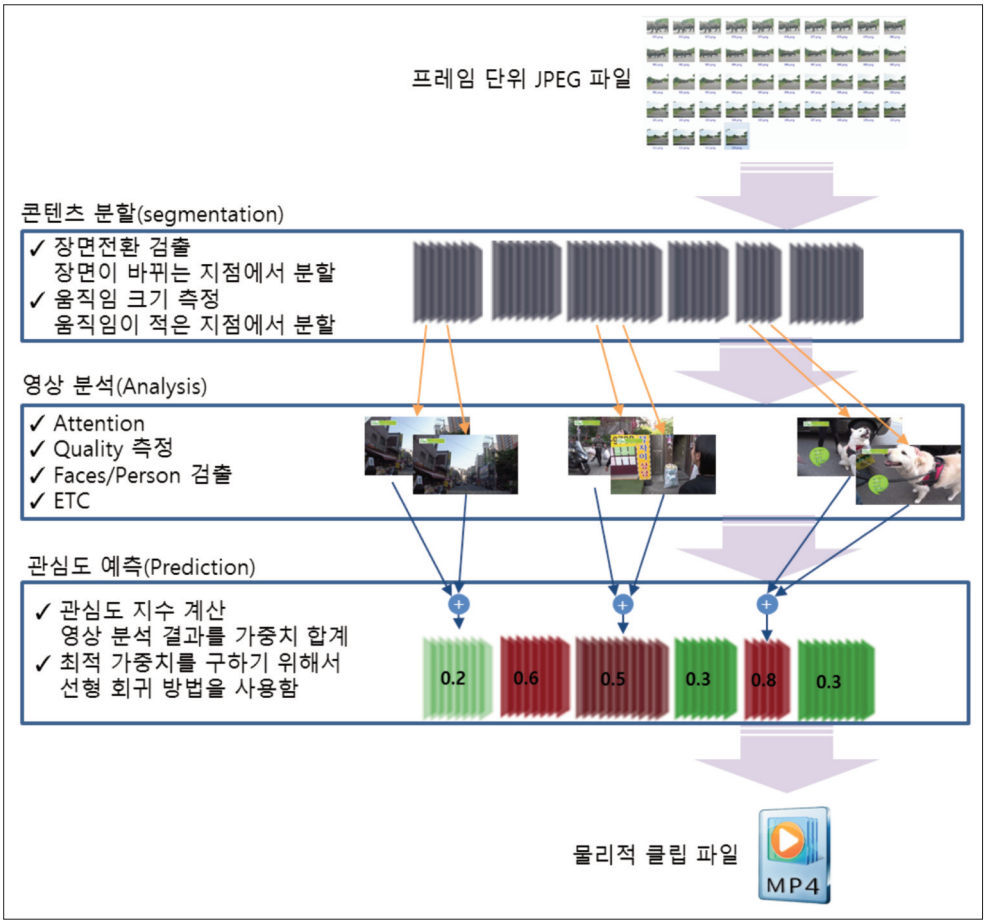
Ⅲ. 동영상 자동 편집 기술의 발전 과정

클립형 미디어 서비스를 제공하기 위해서는 동영상을 축약하는 과정이 필수적이다. 현재 동영상 축약 과정은 사람이 전체 영상을 확인하고 클립형 미

디어로 생성할 짧은 길이의 구간을 발췌하여 파일로 생성하고 있다. 이러한 수동 축약 과정을 거칠 경우 제작 비용과 서비스 품질 측면에서 한계점이 존재한다. 현재 클립형 미디어 서비스가 선택하고 있는 수동 축약 과정은 클립형 미디어 서비스를 제공하는 채널과 프로그램이 많아질수록 제작 비용이 크게 증가할 수 밖에 없다. 또한 수동으로 축약을 진행하는 시간에 의해 발생하는 서비스 지연은 서비스 품질 측면에서 시청자의 만족도를 저해하는 요소가 된다. 따라서 위의 문제를 보완하기 위해 동영상을 자동으로 축약하기 위한 자동 편집 기술의 개발이 필수적으로 필요하다.

1. 딥러닝 이전의 자동 편집 기술

딥러닝 기술 이전에도 이미지 분석 기술을 이용한 자동 편집 기술에 대한 연구는 계속되어 왔다. 대표적으로 M. Gygli[3]에 의해 제안된 동영상 편집 기술은 기존의 다양한 이미지 분석 기법을 활용하여 유의미한 결과를 보여주었다. 따라서 본 장에



<그림 5> 이미지 분석 기술을 활용한 동영상 자동 편집 기술

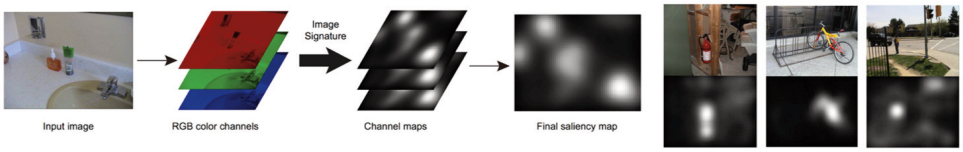
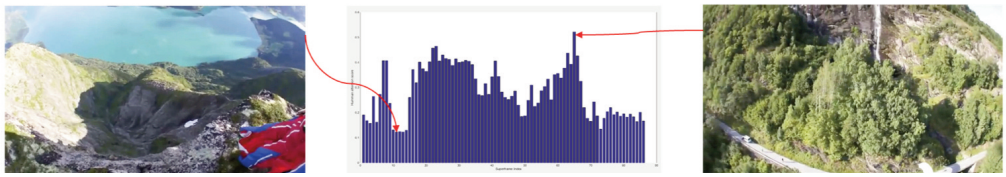
서는 M. Gygil에 의해 제안된 기술을 분석하여 딥러닝 이전의 자동 편집 기술에 대해 알아보려 한다.

〈그림 5〉와 같이, 이미지 분석 기술을 활용한 동영상 편집 기술은 크게 콘텐츠 분할, 영상분석, 관심도 예측의 3단계 과정으로 구성되어 있다. 기본적으로 연속된 이미지들을 유의미한 단위의 세그먼트로 분할(1단계)하고, 각 세그먼트에 속해 있는 이미지에서 특징 값들을 추출(2단계)한 후, 각 이미지 특징 값들의 선형 조합으로 예측한 관심도 지수(Interestness)를 세그먼트 단위로 평균 내어 세그먼트의 평균 관심도 지수(단계3)를 구한다. 이후에는 주어진 목표 시간을 고려하여 관심도 지수가 작은 세그먼트부터 삭제하는 방식으로 편집을 수행한다. 좀 더 자세히 알아보면, 본 기술은 이미지 분석을 기반으로 하고 있기 때문에, 준비단계에서 동영상을 프레임 단위의 이미지 파일로 복호한다. 이후 콘텐츠 분할 단계에서는 연속적인 프레임 단위의 이미지들을 유의미한 단위로 분할하여 편집을 수행하기 위한 최소단위인 세그먼트(segment)를 생성한다. 이때 분할 지점을 선택하는 기준은 장면전환이 일어난 지점과 객체 인식 및 추적(SURF[9])을 통해 객체 움직임이 가장 적은 지점으로, 이는 세그먼트를 이어 붙였을 때 연속된 두 개의 세그먼트가 부자연스럽게 이어지는 것을 방지하기 위해서이다. 다음 단계는 각각의 세그먼트 안에 속해 있는 이미지들에서 영상 분석 기술을 통해 특징 값들을 찾고 이를 이용하여 품질 지수를 생성해 내는 영상 분석 단계이다. 이때 사용하는 영상 분석 기술들은 Saliency Map 검출[4]에 기반한 시각적 주의 지수(visual attention score) 검출[5], 에지/칼라 분포에 따른 이미지 품질 예측[6], 이미지 내에서 얼굴과 인물이 차지하는 비중[7, 8] 등이다. 자세한 내용은 〈표 1〉에 정리하였다.

영상 분석을 통해 생성한 특징 값 i를 사용하여

<표 1> 영상 분석 기술 상세 내용

시각적 주의 지수(Visual attention score) 검출 기술

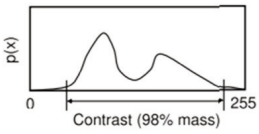
• Saliency Map Detection
  • 영상에서 인간이 생물학적으로 관심있어 할 만한 영역 또는 객체를 검출하는 알고리즘
  • 이미지를 주파수 정보로 변환하고, 주파수 영역에서 페이즈 변화 정보만을 추출한 후, 주파수 정보를 화소 영역으로 역변환하는 방법으로 검출함.
 
 	
• Human attention score
  • 프레임 별로 saliency map을 생성한 후, saliency map을 [0 ~ 1] 사이의 값으로 normalizing 함.
  • saliency map내 0인 값들을 제외하고 나머지 값들의 평균값이 1에 가까우면 사람들이 관심있어 하는 프레임으로 결정함.
 

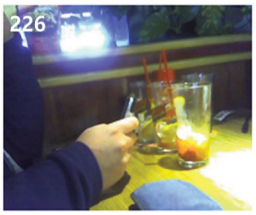
이미지 품질 예측 기술

- 이미지 품질 = Contrast와 에지 분포 정보를 이용해서 사진의 품질을 예측하는 알고리즘
- Contrast가 높을 수록 좋은 품질의 영상일거라 가정함

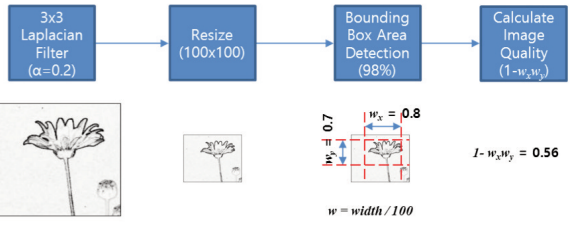
$$H(i) = H_r(i) + H_b(i) + H_g(i)$$

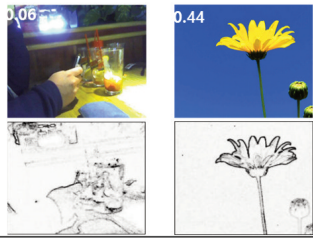
Color channel (Red/Green/Blue) histogram , H_r / H_g / H_b
Combined histogram H는 영상 크기로 나누어줌.





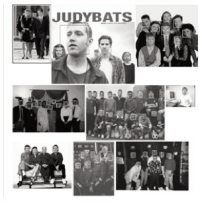
- 에지가 영상의 중심부에 넓게 분포할 수록 좋은 품질의 영상일거라 가정함.



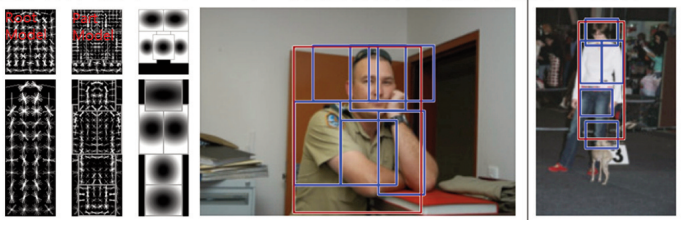


얼굴과 인물 검출 기술

- Quality Score = the area of the bounding box of faces and persons / the frame size
- Faces Detection
  - Viola, P., etc., Robust real-time face detection. IJCV (2004)
  - 높은 검출률을 달성하면서도 극도로 빠르게 얼굴 검출을 수행할 수 있는 프레임워크



- Person Detection
  - Felzenszwalb, P.F., etc.,.: Object detection with discriminatively trained part based models. PAMI (2010)
  - 높은 검출률을 달성하면서도 극도로 빠르게 얼굴 검출을 수행할 수 있는 프레임워크



구한 품질 지수를 u_i로 정의하고 이미지 한 장에 총 N개의 특징 값들이 존재할 때, 이미지 한 장의 관심도 지수를 아래와 같은 선형 예측 모델을 통하여 구한다.

$$i_k = w_0 + \sum_{i=1}^{N} w_i u_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} w_{i,j} u_i u_j$$

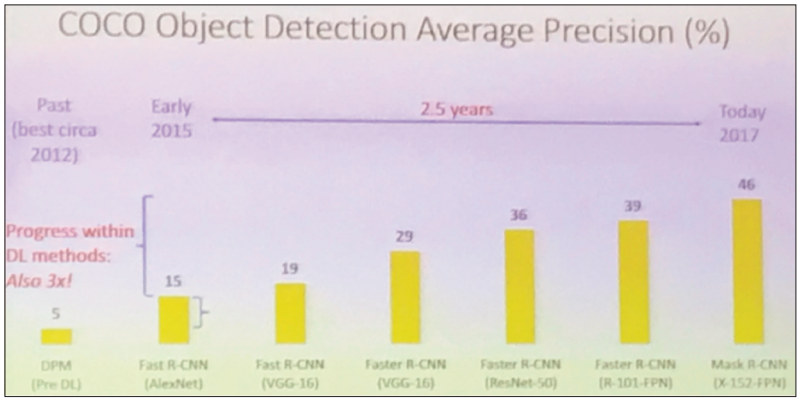
위의 식에서 알 수 있듯이, 특징 값 i를 사용하여 구한 품질 지수 u_i에 곱해지는 적절한 가중치값 w를 구하는 것이 핵심 포인트인데, 특징 값들 간의 상호작용을 고려했다는 특징이 있다. 해당 논문에서는 사람이 직접 축약을 수행하여 Ground truth를 생성하였으며, 최소 자승(Least square)법을 사용하여 가중치 값 w를 학습시켰다.

2. 초기 딥러닝 기반 자동 편집 기술 – CNN 활용

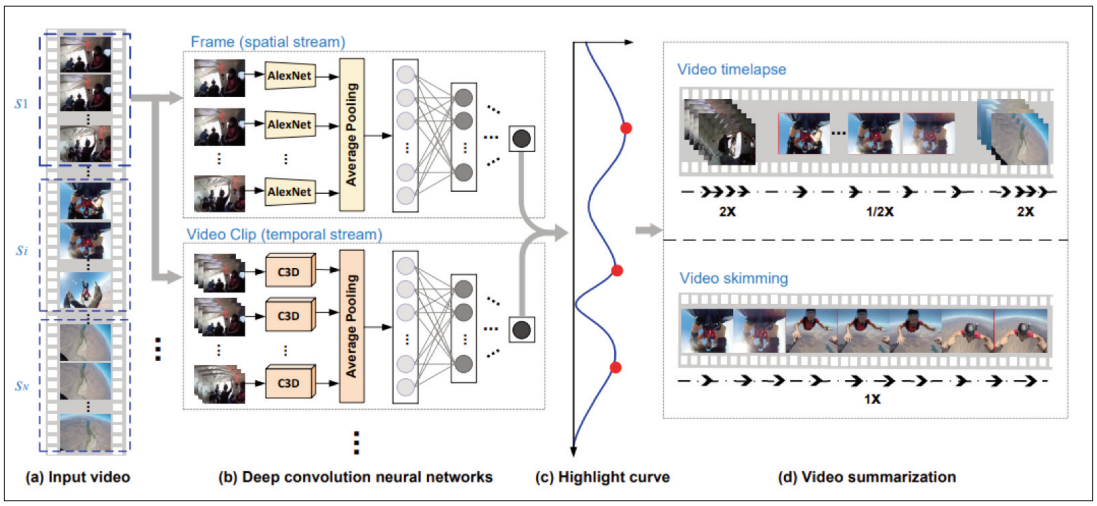
앞에서 살펴본 것과 같이 자동 편집 기술의 핵심은 영상을 분석하여 사용자에게 유의미한 특징 값들을 얼마나 정확하게 찾아내느냐로 정의할 수 있다. 특히 보통 사용자에게 유의미한 특징 값들은 영상 내에 존재하는 객체(Object)에 의해 좌우되는 경우가 많으므로, 동영상 편집 기술은 영상 내에 존재하는 객체를 정확하게 인식하는 것이 매우 중요하다. 〈그림 6〉에서 알 수 있듯이, 최근 딥러닝 기술의 등장과 함께 객체 인식 기술의 정확도가 딥러닝 이전의 기술과 비교하여 매우 가파르게 향상되고 있다. 실제로 MS사가 제공하는 머신러닝 데이터셋인 Common Objects in Context(COCO)[19]를 이용하여 객체 인식 성능을 테스트하였을 때, 2015년 딥러닝 기술에 의해 기존 알고리즘과 비교하여 3배의 성능이 향상(5% → 15%)되었으며, 이후 2017년까지 2.5년 동안 딥러닝 기술이 발전하면서 3배의 추가 성능 향상(15% → 46%)이 이루어졌다.

한편, Tensorflow[15], Darknet[16] 등과 같이 딥러닝 신경망을 쉽게 구성할 수 있는 플랫폼들이 오픈소스로 공개되고, 이러한 플랫폼 위에서 구동할 수 있는 Faster R-CNN[17], Single Shot Multibox Detection(SSD)[18], You Only Look Once(YOLO)[20]와 같은 객체 검출 신경망 모델들도 오픈소스로 공개되면서 딥러닝 기술을 활용하기 위한 문턱이 많이 낮아졌다. 따라서 자연히 딥러닝 기반의 객체 인식 기술을 자동 편집 기술에 적용하려고 하는 움직임도 생겨났다.

MS사의 Yao[14]는 〈그림 7〉과 같이 각각의 이미



<그림 6> 객체 인식 기술 성능 향상



<그림 7> MS사의 Yao 알고리즘, 시간 및 공간적 분석을 위한 별도의 딥러닝 신경망을 사용하며, 하이라이트 지수를 활용하여 동영상 편집을 수행함

지(Frame)를 공간적으로 분석하기 위한 딥러닝 신경망(AlexNet)[12]과 순차적인 이미지들(Clip)을 시간적으로 분석하기 위한 딥러닝 신경망(C3D)[13]을 사용하여 각 세그먼트의 하이라이트 지수를 구한 후, 하이라이트 지수를 활용하여 동영상 편집을 수행하는 알고리즘을 제안하였다. 딥러닝 신경망을 통해서 기존의 알고리즘에 비해 객체 검출 성능을 향상시켰다는 점과 순차적인 이미지들을 사용하여 시간적 분석을 하는 신경망을 추가했다는 점에서 제안된 알고리즘은 이후 딥러닝을 사용한 동영상 편집 알고리즘에 많은 영향을 주었다. 또한 동영상 편집의 형태를 Timelapse와 Skimming의 두 가지 타입으로 정의한 것도 의미가 있다. 두 가지 형태 모두 하이라이트 지수를 활용한다는 공통점이 있지만, Timelapse의 경우는 하이라이트 지수가 낮을수록 빠른 속도로 프레임을 재생하고 Skimming의 경우는 하이라이트 지수가 낮은 프레임을 삭제하는 방식으로 동영상을 편집한다는 차이점이 있다. 위의 장점에도 불구하고 Yao의 알고리즘은 프레임 레벨의 학습 데이터가 필요하다는 점과 공간적/시간적 분석을 위한 딥러닝 신경망이 분리되어 있어 신경망 학습이 힘들고 알고리즘 수행 시간이 오래 걸린다는 단점이 있다. 따라서 이러한 문제를 극복하기 위해 세그먼트 레벨의 학습 데이터와 공간적/시간적 분석이 하나의 딥러닝 신경망에서 수행되도록 하기 위한 연구가 진행되었다.

Panda[11]는 영상의 상황에 따라 딥러닝 네트워크 상에서 공간적으로 활성화되는 특정 영역이 존재하는 것(예를 들어, 〈그림 8〉과 같이 양치질 영상의 경우에는 "손에 들려있는 칫솔과 입 주변" 영역이 활성화 됨[10])에 영감을 얻어 DeSumNet이라는 영상 편집 솔루션을 제안하였다. DeSumNet은 입력 영상을 시공간적으로 분석할 수 있는 3D 합성곱 신경망(Convolution Neural Network: CNN) 구조를 활용하여 특정 상황에서 시공간적으로 활성화되어 있는 영역을 찾는 "합성곱 신경망"과 사용자의 의도를 반영하여 활성화 된 영역을 중요도 점수(Importance score)로 변환해 주는 "완전히 연결된

<그림 8> 딥러닝 신경망에서 영상의 상황에 따른 활성화 영역 표시

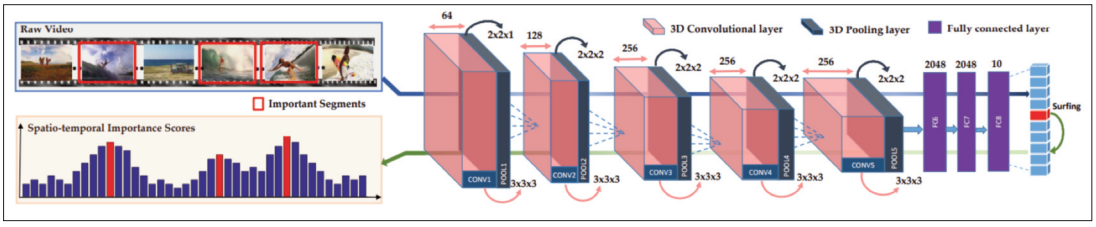
망"으로 구성되어 있다.

예를 들어, 〈그림 9〉와 같이 서핑(Surfing) 영상이 입력 영상인 경우, 영상을 일정한 크기의 클립으로 나눈 후 나뉘어진 클립 중에서 사용자가 중요하다고 생각한 클립을 선택하여 DeSumNet을 학습시키면, 차후에 테스트 영상을 DeSumNet에 공급하였을 때 사용자가 선택한 영상과 유사한 클립의 중요도 점수가 높게 계산되어 출력된다. 이후 중요도 점수가 높은 클립들을 우선적으로 추려서 편집 동영상을 생성하게 된다.

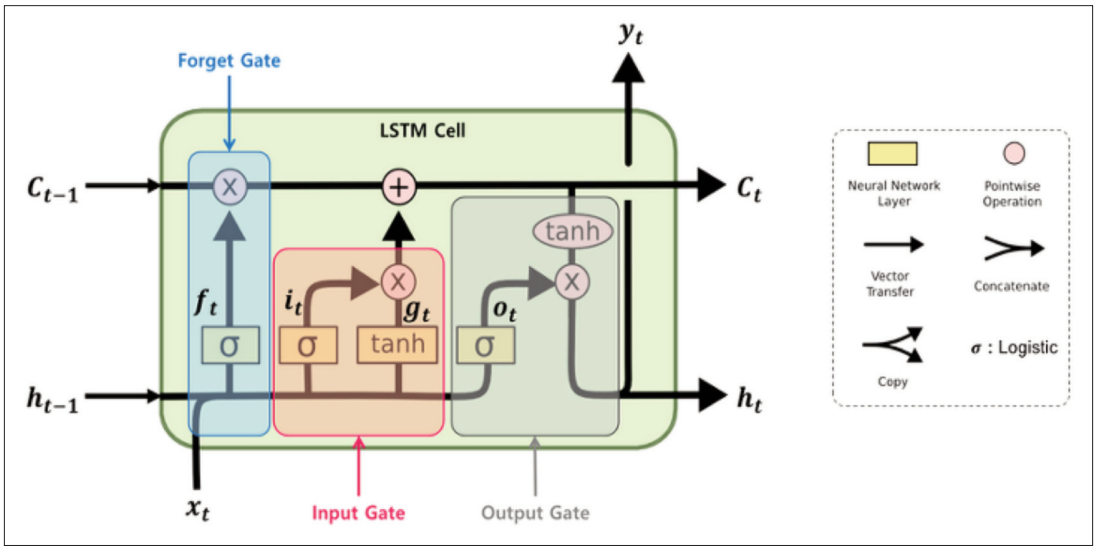
Panda의 알고리즘은 하나의 통합된 딥러닝 신경망인 DeSumNet을 세그먼트 레벨의 학습 데이터를 활용하여 학습시키고, 이를 이용하여 동영상 편집을 수행할 수 있다는 점에서 딥러닝 신경망을 활용한 알고리즘 중에서도 매우 의미 있는 결과를 보여줬다고 할 수 있다.

3. 최근 딥러닝 기반 자동 편집 기술 – LSTM, GAN, Attention Etc.

앞서 살펴본 CNN 기반의 자동 편집 기술들은 2차원 CNN을 3차원으로 확장한 버전으로 프레임



<그림 9> DeSumNet 구조 - 3D CNN을 사용하여 시공간 중요도 점수를 계산함

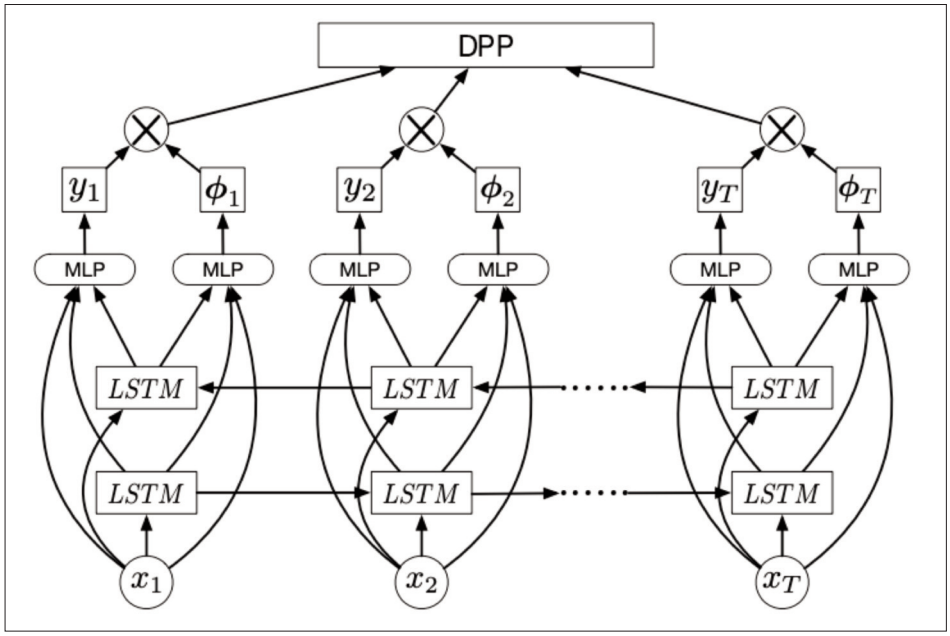


<그림 10> Long Short-Term Memory(LSTM) 구조. (원 출처: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

간 맥락을 포착하기에는 한계가 있었다. 따라서 이후 이러한 한계를 극복하기 위해 LSTM 기반의 기술이 제안되었다. LSTM(Long Short-Term Memory)는 S. Hochreiter와 J. Schmidhuber가 1997년에 제안한 셀로써, RNN 셀이 가지고 있는 장기 의존성 문제(기억이 오래될수록 현재의 상태에 영향을 주지 못하는 문제)를 해결하려고 제안되었다.

〈그림 10〉에서 보면 LSTM 셀에서는 상태(state)가 두 개의 벡터 h_t와 c_t로 나누어진다는 것을 알 수 있다. h_t를 단기 상태(short-term state), c_t를 장기 상태(long-term state)라고 볼 수 있다. LSTM의 핵심은 네트워크가 장기 상태(c_t)에서 기억할 부분, 삭제할 부분, 그리고 읽어들일 부분을 학습하는 것이다. 장기 기억(c_{t-1})은 셀의 왼쪽에서 오른쪽으로 통과하게 되는 forget gate를 지나면서 "일부의 기억"을 잃고, 그 다음 덧셈 연산으로 input gate로부터 새로운 기억 일부를 추가한다. 이렇게 타임 스텝마다 일부의 기억을 삭제하고 추가하는 과정을 거쳐 만들어진 장기 상태(c_t)는 별도의 추가 연산 없이 바로 출력된다. 그리고 덧셈 연산 후에 장기 상태(c_t)는 복사되어 output gate의 tanh 함수로 전달되어 단기 상태 h_t와 셀의 출력인 y_t를 만든다. 결과적으로 forget gate, input gate를 통해 장기 상태를 끊임없이 갱신하는 방법으로 시계열 문맥 정보를 현재의 출력에 반영할 수 있게 된다.

Zhang[22]은 LSTM과 DPP(Determinantal Point Process)를 결합한 자동 편집 기술을 제안하였다. 〈그림 11〉과 같이 양방향 LSTM을 이용하여 상대적으로 긴 호흡의 프레임 진행에 따른 문맥 정보를 습득하였으며, 해당 정보를 Multi-Layer Perceptron 네트워크에 제공하여 프레임 레벨의 중요도 지수 y_t와 프레임 간의 유사도 지수 ϕ_t를 계산하였다. 마지막으로 중요도 지수와 유사도 지수의 내적을 계산하여 DPP를 수행한다. DPP는 한정된 수의 샘플로 모집단을 최대한 잘 표현하기 위해 사

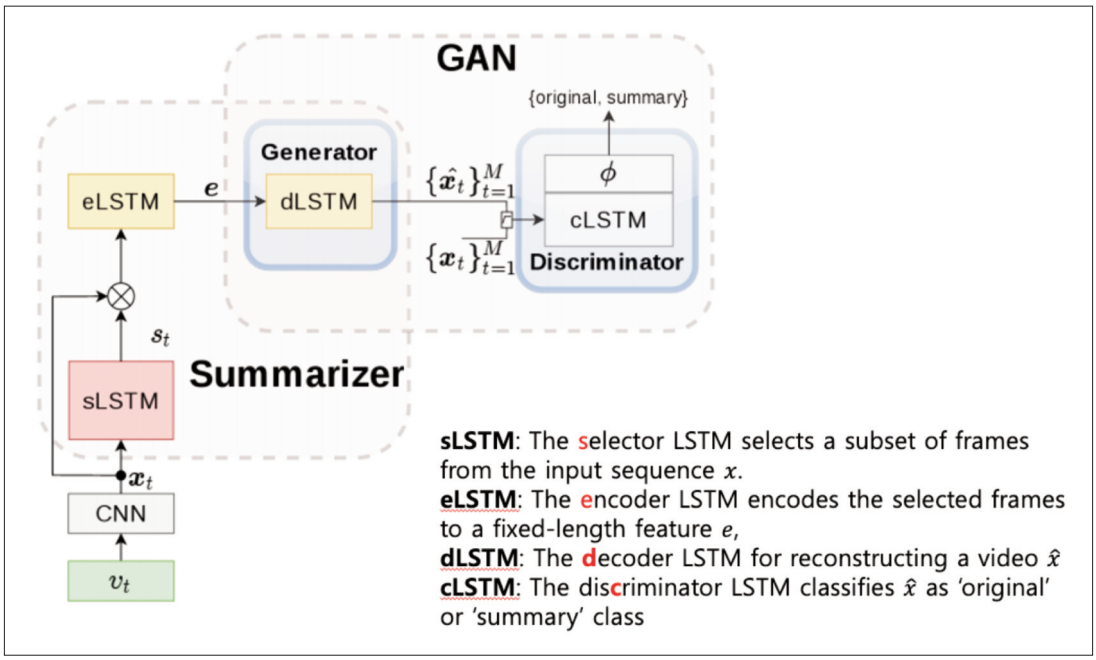


<그림 11> Zhang 알고리즘, 양방향 LSTM과 DPP를 활용한 자동 편집 기술, dppLSTM

용하는 기술로, 중요도 지수가 높은 유사한 프레임들이 많이 선택되는 문제를 억제해 주는 효과가 있다.

Zhang의 알고리즘은 LSTM을 통해 중요도 지수를 계산하고 DPP를 이용하여 프레임간 중복성을 고려해 주는 방법으로 자동 편집 성능을 향상시켰지만, 근본적으로 네트워크를 학습하기 위해서는 프레임 레벨의 중요도 지수를 포함한 방대한 학습 데이터가 필요한 한계가 있다. 따라서 이를 극복하기 위해 Mahasseni[23]는 GAN(Generative Adversarial Network)을 사용한 알고리즘을 제안하였다.

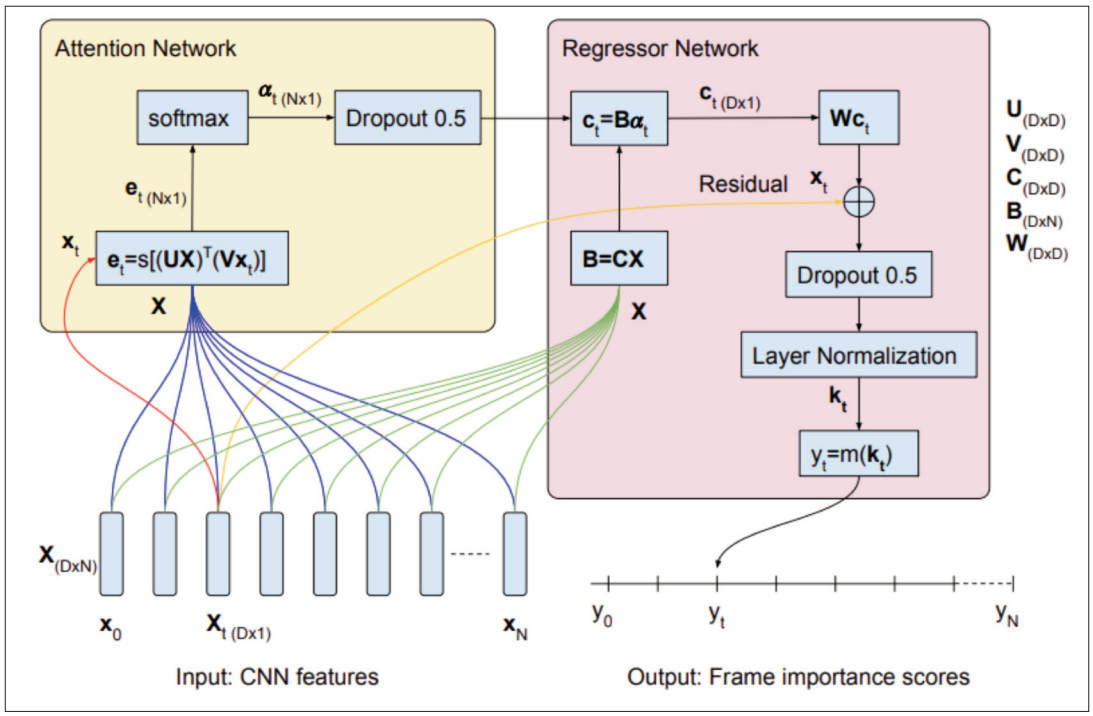
〈그림 12〉에서 알 수 있듯이 LSTM을 기반으로 영상의 프레임간 시계열 문맥을 획득하는 것은 Zhang과 동일하다. Mahasseni 알고리즘은 크게 Summarizer 블록과 GAN 블록으로 나뉜다. Summarizer 블록의 초반에서는 Zhang의 알고리즘과 유사하게 CNN을 통해 이미지 v_t로 부터 추출한 특징 값 x_t를 중요도 지수 계산을 위한 LSTM 기반 네트워크(sLSTM)에 넣어 프레임 단위 중요도 지수인 s_t를 구한다. 중요도 지수 s_t는 0과 1 사이의 값을 가지며, 모든 프레임의 특징 값 x_t은 중요도 지수에 의해 가중치가 적용되게 된다. 중요도 지수가 0과 1의 값만 갖는 특수한 경우에는 일부 프레임의 특징 값만을 선별하는 역할을 하게 된다. 다음으로 eLSTM은 가중치가 적용된 프레임의 특징 값 시퀀스를 고정된 길이의 특징 값(Latent feature) e로 변환한다. 마지막으로 dLSTM은 e를 입력 받아 입력 프레임의 특징 값 x_t에 대응하는 새로운 특징 값을 $\hat{x}_t$복호하여, 최종적으로 복원된 프레임의 특징 값 시퀀스 $\hat{x}=\{\hat{x}_t, \hat{x}_2, \cdots \hat{x}_M\}$를 만들어 낸다. GAN 블록은 생성자(Generator)와 판별자(Discriminator)로



<그림 12> Mahasseni 알고리즘, GAN활용한 비지도 학습 기반 자동 편집 기술

구성되어 있는데, 생성자 역할은 앞서 설명한 dLSTM이 수행하고 있다. 판별자의 경우는, 오리지날 특징 값 시퀀스 x와 Summarizer를 거쳐 복원된 특징 값 시퀀스 $\hat{x}$를 입력으로 받아, $\hat{x}$이 original인지 summary인지를 분류하는 기능을 한다. 좀 더 자세히 설명하면, dLSTM은 두 벡터 x와 $\hat{x}$의 거리를 측정하여 두 벡터의 거리가 가까운 경우는 $\hat{x}$을 original로 판별하고, 그 반대의 경우는 summary로 판별하게 된다. 결국 dLSTM과 cLSTM이 생산적 적대 신경망을 구성하게 되며, 판별자가 생성자가 생성한 프레임 특징값 시퀀스와 원본 프레임 특징값 시퀀스를 구분하지 못하게 될 때까지 학습을 진행하게 된다.

Mahasseni의 알고리즘은 GAN을 자동 편집 기술 문제에 적용하여 비지도 학습 방식을 도입함으로써 라벨링된 학습 데이터가 존재하지 않는 경우에도 대응할 수 있게 하였다. 하지만 프레임간 시계열 문맥을 획득하는 방법으로 여전히 LSTM을 사용하고 있다. LSTM은 〈그림 10〉과 같이 시계열 문맥 정보를 장기 상태(c_t)에 저장할 때 forget gate와 input gate를 통해 관리함으로써 기존의 RNN 셀이 가지고 있던 장기 기억 희석 문제를 어느 정도 해결할 수 있었다. 그러나 시계열 문맥 정보가 한정된 크기를 가지는 장기 상태 벡터(c_t)에 저장되는 한계가 있어, 편집과 같이 영상 전체를 아우르는 문맥 정보가 필요한 경우 장기 상태에 저장된 시계열 문맥 정보로는 정보가 부족한 문제가 발생한다. 이러한 문제를 해결하기 위해, Fajtl[24]은 현재 프레임의 중요도 지수를 생성하는 매 시점마다 전체 입력 영상을 참고할 수 있는 Attention 매커니즘을 적용하여 VASNet 알고리즘을 만들었다. 물론 전체 입력 영상을 동일한 비율로 참고하는 것이 아니라, 해당 프레임의 중요



<그림 13> Fajtl알고리즘, Self-attention 기반 자동 편집 기술, VASNet

도 지수와 연관이 있는 입력 영상을 좀 더 집중(attention, 어텐션)해서 보게 된다. 어텐션을 구하는 여러 방식이 존재하지만, Fajtl는 입력 영상과 다른 나머지 입력 영상들의 연관 관계를 통해 구하는 셀프 어텐션(Self-attention) 을 사용하였다.

〈그림 13〉에서 알 수 있듯이, 입력 영상들의 연관 관계를 고려하여 셀프 어텐션(a_t)을 생성하는 Attention Network와, 셀프 어텐션과 입력 영상들을 이용하여 중요도 지수를 생성하는 Regressor Network로 구성된다.

4. 자동 편집 성능

앞 절에서 딥러닝 기반 자동 편집 기술에 대해서 알아보았다. 본 절에서는 해당 기술들의 성능을 정리하려고 한다. 자동 편집 기술 분야에서는 객관적인 성능 검증을 위해, 〈표 2〉의 공개 데이터 셋을 사용한다. 데이터 셋에서 Annotation Type은 keyshots, frame-level 중요도 지수(importance scores), keyframes로 나뉜다. Keyshots는 영상의 연속된 일정 범위를 중요 클립으로 선택하는 방법이고, frame-level 중요도 지수는 모든 프레임에 0과 1 사이의 실수로 중요도 값을 표기한 방법이다. 마지막으로 Keyframes는 모든 프레임을 0과 1로 표시하여 중요한 프레임을 표기한 방법이다.

〈표 3〉은 SumMe, TvSum 데이터 셋을 이용해 검증한 자동 편집 기술의 성능을 나타낸다. 성능 지표는 harmonic mean F-score를 사용하였으며,

<표 2> 자동 편집 학습 및 검증을 위한 데이터 셋[24]

Dataset	Videos	User annotations	Annotation type	Video length (sec)		
				Min	Max	Avg
SumMe	25	15-18	keyshots	32	324	146
TvSum	50	20	frame-level importance scores	83	647	235
OVP	50	5	keyframes	46	209	98
YouTube	39	5	keyframes	9	572	196

<표 3> 자동 편집 기술의 성능[24]

Method	SumMe		TvSum	
	Canonical	Augmented	Canonical	Augmented
dppLSTM	38.6	42.9	54.7	59.6
M-AVS	44.4	46.1	61.0	61.8
DR-DSN$_{sup}$	42.1	43.9	58.1	59.8
SUM-GAN$_{sup}$	41.7	43.6	56.3	61.2
SASUM$_{sup}$	45.3	-	58.2	-
Human	64.2	-	63.7	-
VASNet (proposed method)	**49.71**	**51.09**	**61.42**	**62.37**

100에 가까울수록 우수한 성능을 나타낸다. 위의 〈표 3〉에서 Canonical 방식은 하나의 데이터 셋으로 학습, 검증, 그리고 테스트 셋을 생성하여 성능을 측정하는 방식이고, Augmented 방식은 하나의 데이터 셋(예, SumMe)에서 20%를 추출해 테스트 셋을 생성하고 나머지 80%와 다른 세 개의 데이터 셋(예, TvSum, OVP, YouTube)의 데이터를 합쳐서 학습과 검증 셋을 생성하는 방법이다. 일반적으로 Augmented 방식을 사용할 경우 학습 데이터가 증가하는 효과가 있으므로, 성능이 향상되는 것을 확인할 수 있다. 〈표 3〉에서 dppLSTM은 Zhang[22]의 알고리즘, SUM-GAN$_{sup}$은 Mahasseni[23]의 알고리즘, 그리고 VASNet은 Fajtl 알고리즘[24]을 뜻한다.

앞선 절에서 소개한 것과 같이 지속적인 알고리즘 개발을 통해, VASNet은 두 데이터 셋 모두에서 괄목할 만한 성능 향상(38.6→49.71, 54.7→61.42)을 얻은 것을 알 수 있다. 특히 Frame-level importance score가 주어져 있는 TvSum 데이터 셋에서는 사람이 수행한 성능에 거의 근접한 결과를 보여준다.

Ⅳ. 자동 편집 기술의 한계 및 과제

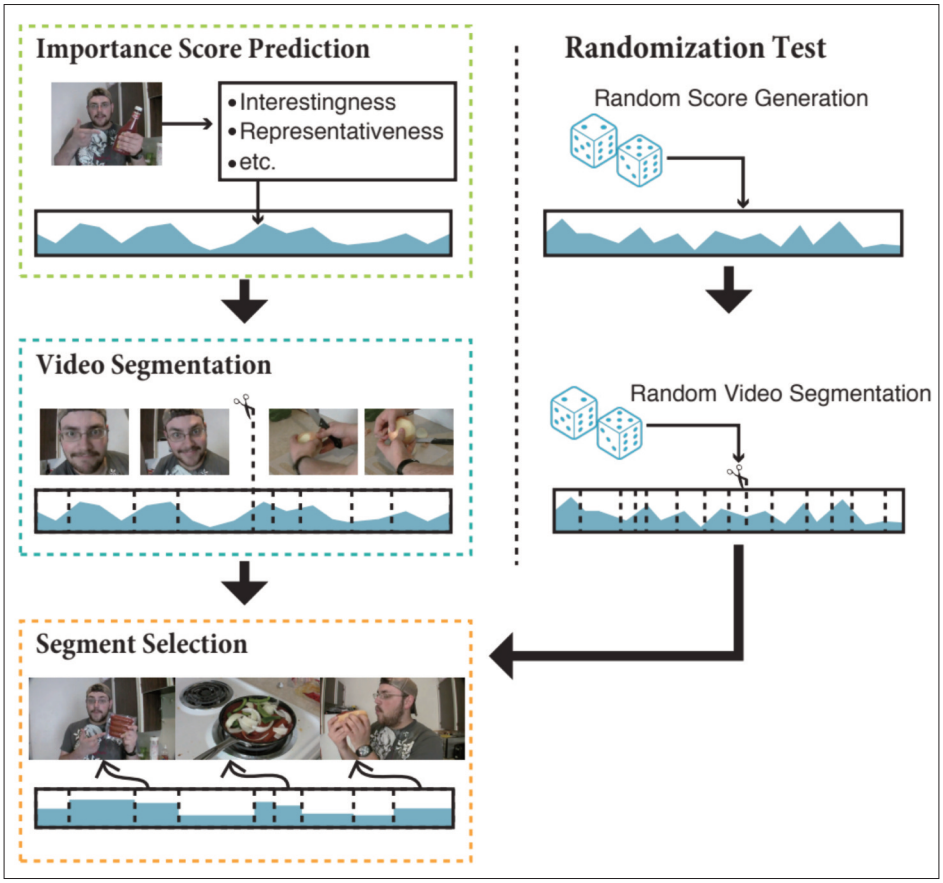
1. 품질 높은 학습데이터 생성의 어려움

앞선 절의 〈표 2〉에서 알 수 있듯이, 자동 편집 기술을 위해 사용할 수 있는 공개된 학습 데이터의 양이 매우 적다. 〈표 2〉에 있는 모든 학습 데이터의 양을 합쳐서 약 8시간 정도 밖에 되지 않는다. 따라

서 〈표 3〉과 같이, 서로 다른 데이터 셋을 섞어서 쓰는 방법을 사용하여 학습 데이터 부족을 해결하려고 하는 시도도 존재한다. 이러한 학습 데이터의 부족의 주된 원인은 자동 편집을 위한 학습 데이터를 생성하는 것이 매우 어려운 일이기 때문이다. 하나의 영상에 대해서 중요하다고 생각하는 프레임은 매우 주관적인 것으로, 〈표 2〉와 같이 하나의 영상에 대해서 복수의 Annotation을 생성해야 한다. 따라서 학습 데이터를 생성하는 시간과 비용이 기하급수적으로 늘어난다. 또한 〈표 3〉의 Human 항목에서 알 수 있듯이, 사람이 생성한 Annotation을 이용하여 F-score를 계산하여도, 100에 한참 부족한 60 중반의 값이 계산된다. 결국 편집이라는 분야는 편집자의 성향에 따라 유동적인 부분이 많으므로, 학습을 진행할 때 목표값을 여러 사람들이 생성한 Annotation을 평균 내어 사용하게 된다.

2. 오래된 워크플로우와 연구 주제의 편중

자동 편집 기술은 많은 발전을 이루었지만, 워크플

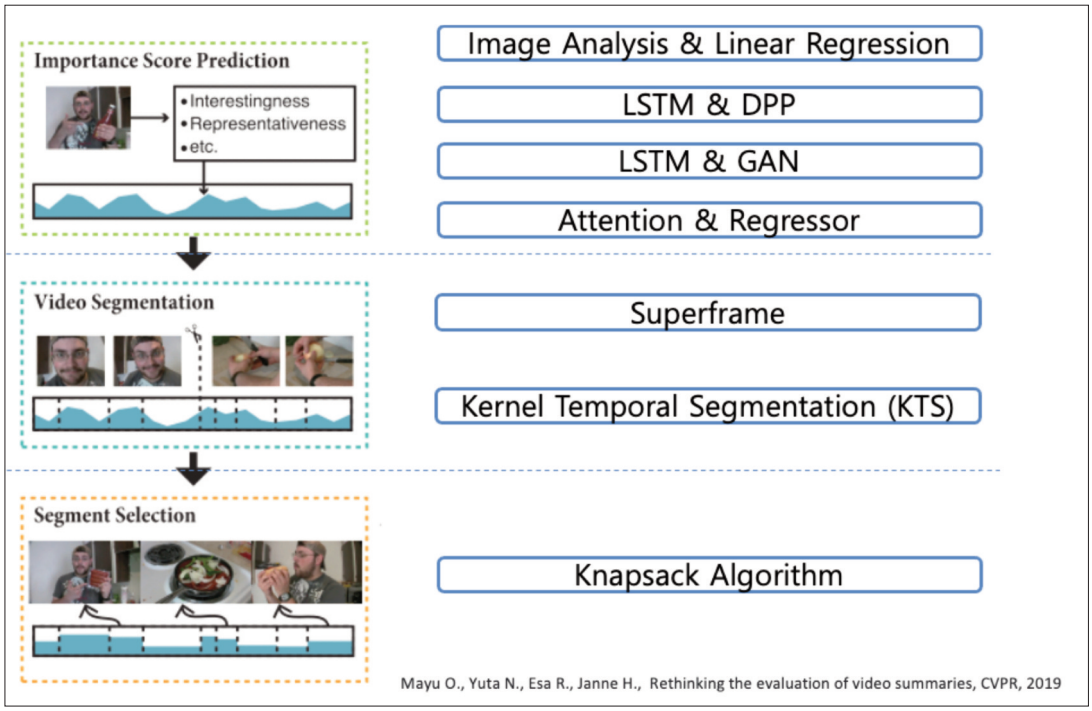


<그림 14> 자동 편집 기술의 워크플로우와 대응되는 연구 내용

로우는 과거로부터 한 번도 바뀌지 않고 〈그림 14〉와 같이 유지되었다. 즉, Importance Score를 예측하여 생성하고, 영상을 분석하여 다양한 크기의 Segment로 분절한 후, Importance Score를 고려하여 일부 세그먼트를 선택적으로 선택하는 방법으로 자동 편집을 수행하는 것이다. 이러한 전통적인 워크플로우 상에서 Importance Score를 계산하기 위한 알고리즘은 그동안 다양하게 제안되었지만, 상대적으로 Segment를 생성하는 알고리즘은 적게 제안되었다. 더 나아가 분할된 Segment를 선택하는 방법은 Knapsack 알고리즘만을 거의 모두 사용하고 있다. 총체적인 자동 편집 기술의 성능 향상을 위해서는 상기 워크플로우의 모든 부분에서 혁신이 일어나야 하지만 지금은 특정한 분야에만 연구가 집중되고 있어, 성능 향상이 제한되는 문제가 발생한다. 특히 마지막 부분인 Segment 선택 부분은 심각한 문제를 가지고 있음을 최근 연구[25]를 통해 알려지게 되었는데, 이를 다음 절에서 자세히 알아보려 한다.

3. Knapsack을 이용한 Segment 선택의 문제

Mayu[25]는 전통적인 자동 편집 기술 워크플로우에서 어떤 부분이 전체 성능에 많은 영향을 끼치는지에 대한 연구를 진행하였다. 이러한 연구를 위해 주요 부분 중 일부를 랜덤으로 결과를 생성하는 알고리즘으로 대체하여 전체 성능이 어떻게 변경되

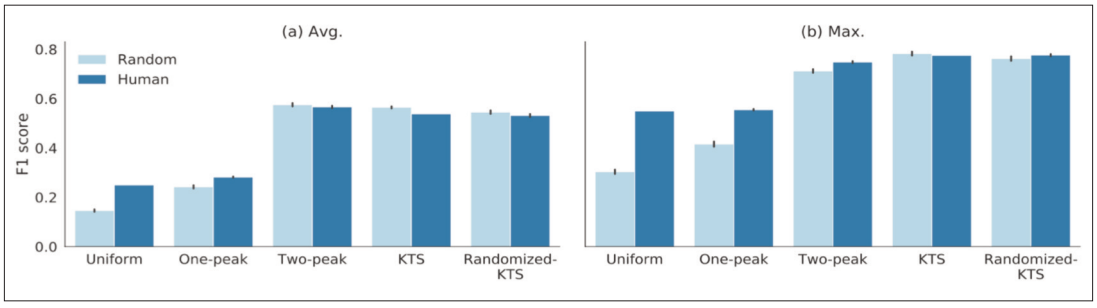


<그림 15> 자동 편집 기술 워크플로우에서 각 파트가 전체 성능에 끼치는 영향을 점검하기 위한 실험 계획[25], 랜덤으로 Importance Score를 생성하거나, 랜덤으로 Segment를 선택함

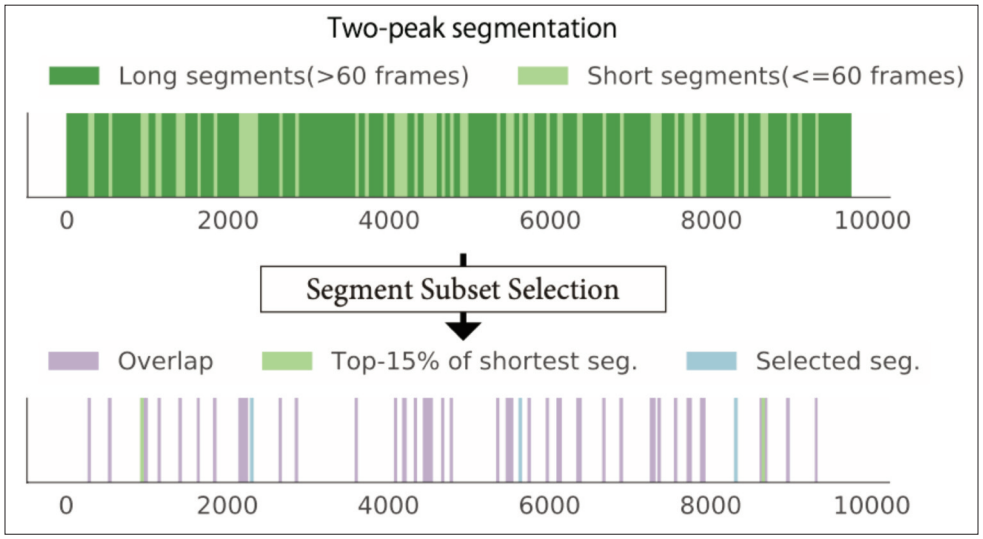
는지 확인해 보았다.

〈그림 16〉에서 Uniform하게 세그먼트를 분할하는 케이스를 제외하면 모든 세그먼트 분할 알고리즘 상에서 사람이 생성한 중요도 지수와 랜덤으로 생성한 중요도 지수의 최종 성능 차이가 거의 없음을 확인할 수 있다. 이러한 현상은 현재까지 이 분야의 연구가 중요도 지수를 생성하는데 집중되어 있다는 점에서 시사하는 바가 매우 크다. 즉, 세그먼트 분할을 균등하게 하는 케이스를 보면 중요도 지수가 최종 성능에 영향을 끼치긴 하지만, 최종 성능을 크게 좌우하는 것은 세그먼트 분할 방법이라는 것을 알 수 있기 때문이다. 물론 이러한 현상은 앞으로 설명할 Knapsack 알고리즘을 세그먼트 선택에 사용함으로써 발생하는 구조적 문제이다.

〈그림 17〉에서 위쪽 그래프의 녹색 영역은 two-peak 세그먼트 분할 방법에 의해 생성된 세그먼트의 경계를 표현한 것이다. 세그먼트 분할에 의해 길이가 서로 다른 여러 세그먼트가 발생하게 되는데,



<그림 16> 서로 다른 세그먼트 분할 알고리즘에 중요도 지수(Importance Score)를 랜덤으로 생성하거나 사람이 생성한 경우를 적용하여 F1-Score 계산함. TvSum 데이터 셋 사용함[25]



<그림 17> Knapsack 알고리즘을 사용할 때, 길이가 긴 세그먼트들이 묵시적으로 버려지고 길이가 짧은 세그먼트들 위주로 선택되는 현상을 보여주는 그래프[25]

세그먼트 길이가 짧은 순으로 나열하였을 때, 상위 15%에 해당하는 세그먼트를 〈그림 17〉의 아래쪽 그래프에 녹색으로 표시하였다. 또한 Knapsack 알고리즘에 의해 선택된 세그먼트는 파란색으로 표시하였다. 따라서 전체 세그먼트의 15%에 해당하는 세그먼트와 Knapsack 알고리즘에서 선택된 세그먼트가 겹치는 경우, 아래쪽 그래프에서 보라색으로 보이게 된다. 결과적으로 아래쪽 그래프를 보면 거의 모든 세그먼트가 길이가 짧은 세그먼트로 선택되는 것을 알 수 있다. 이것은 Knapsack 알고리즘이 주어진 목표 길이를 넘지 않으면서 중요도 지수의 총합을 최대화하는 방식으로 동작하기 때문이다. 따라서 길이가 긴 세그먼트는 상대적으로 선택될 가능성이 낮아진다. 이러한 현상으로 인해, 〈그림 16〉과 같이 균등하게 세그먼트 분할을 하는 경우를 제외한 다른 세그먼트 분할 방식에서는 세그먼트를 선택할 때 중요도 지수의 영향력이 적어지는 것이다. 따라서 세그먼트 선택 방식의 개선이 매우 시급한 당면 과제이다.

V. 결 론

본 기고에서는 자동 편집 기술의 개념과 국내 서비스 동향을 소개하고, 자동 편집 기술의 발전 과정에 대해 살펴보았다. 수동 편집 과정에서 발생하는 비용과 서비스 지연 문제를 해결하기 위해 기존에는 이미지 분석 기술을 활용한 접근을 하였으며, 최근에는 딥러닝 기술을 활용한 접근이 활발하게 이루어지고 있다는 것을 확인할 수 있었다. 따라서 딥러닝 기술에 기반한 최근 알고리즘들을 살펴 보았으며, 어떠한 기술을 활용하여 성능을 향상시킬 수 있었는지를 확인할 수 있었다. 마지막으로는 자동 편집 기술이 가지고 있는 근본적인 문제들에 대해서 살펴 보면서, 현재 세그먼트 분할과 선택 과정에서 예측을 통해 생성한 중요도 지수의 영향력이 감소하는 문제가 있음을 확인할 수 있었다. 이를 통해 자동 편집기술의 성능 향상을 위해서는 세그먼트 분할과 선택 분야에 대한 활발한 기술 개발이 필요함을 확인하였다.

참 고 문 헌

[1] Gygli, M., Grabner, H., Van Gool, L.: Video summarization by learning submodular mixtures of objectives. In: CVPR (2015)
[2] Zhang, K., Chao, W.l., Sha, F., Grauman, K.: Summary transfer: exemplar-based subset selection for video summarization. In: CVPR (2016)
[3] M. Gygil, etc., Creating Summaries from User Video, ECCV2014, pp. 505~520

[4] X. Hou, J. Harel and C. Koch, Image Signature: Highlighting Sparse Salient Regions, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 1, pp. 194-201, Jan. 2012
[5] N. Ejaz, I. Mehmood, SW Baik, Efficient visual attention based framework for extracting key frames from videos, in Signal Processing: Image Communication 28 (1), 34-44, Jan. 2013
[6] Y. Ke, X. Tang and F. Jing, The Design of High-Level Features for Photo Quality Assessment, in CVPR, 2006
[7] Viola, P., Jones, M.: Robust real-time face detection. IJCV (2004)
[8] Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. PAMI (2010)
[9] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool SURF: Speeded Up Robust Features, Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346-359, 2008
[10] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization, CVPR 2016
[11] R Panda, A Das, Z Wu, J Ernst, AK Roy-Chowdhury, Weakly supervised summarization of web videos, 2017 IEEE International Conference on Computer Vision (ICCV), 3677-3686
[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012
[13] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In ICCV, 2015
[14] T. Yao, T. Mei, and Y. Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 982-990, 2016
[15] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. A. Tucker, V. Vanhoucke, V. Vasudevan, F. B. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng., TensorFlow:Large-scale machine learning on heterogeneous distributed systems, arXiv preprint, 1603.04467, 2016. arxiv.org/abs/1603.04467. Software available from tensorflow.org
[16] Joseph Redmon, "Darknet: Open Source Neural Networks in C," Software available from http://pjreddie.com/darknet/, 2013-2016
[17] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015
[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. E. Reed. SSD: single shot multibox detector. CoRR, abs/1512.02325, 2015
[19] COCO:Common Objects in Context (2016). http://mscoco.org/dataset/#detections-leaderboard. Accessed 25 July 2016
[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. arXiv preprint arXiv:1506.02640, 2015
[21] 이동관, 지상파 UHD 현황 및 부가서비스, 방송과 기술, 14 Oct. 2016
[22] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Video summarization with long short-term memory. In European Conference on Computer Vision (ECCV), pages 766-782, may 2016
[23] Behrooz Mahasseni, Michael Lam and Sinisa Todorovic, Unsupervised Video Summarization with Adversarial LSTM Networks, CVPR, 2017
[24] Fajtl, J., Sokeh, H., Argyriou, V., Monekosso, D., & Remagnino, P., Summarizing Videos with Attention, 11367 LNCS, 39 - 54, 2019
[25] Mayu O., Yuta N., Esa R., Janne H., Rethinking the evaluation of video summaries, CVPR, 2019

필 자 소 개

홍 순 기

– 2006년 9월 : 연세대학교 전기전자공학과 학사
– 2016년 2월 : 연세대학교 전기전자공학과 박사
– 2013년 9월 ~ 2016년 9월 : 삼성전자 DMC 연구소 책임연구원
– 2016년 10월 ~ 현재 : SBS 미디어솔루션팀 매니저
– 주관심분야 : 비디오/영상 신호 처리, 모바일 비디오 통신, 방통 융합 미디어 서비스