논문 20-45-07-03 The Journal of Korean Institute of Communications and Information Sciences '20-07 Vol.45 No.07
https://doi.org/10.7840/kics.2020.45.7.1152

시간적 행동 검출: 서베이

은 현 준◆, 문 진 영*, 박 종 열*, 정 찬 호°, 김 창 익**

Temporal Action Detection: A Survey

Hyunjun Eun◆, Jinyoung Moon*, Jongyoul Park*, Chanho Jung°, Changick Kim**

요 약

컴퓨터 비전 분야에서 비디오 이해를 위한 행동 인식 연구는 오랫동안 행해져 왔다. 하지만 행동 인식 연구에서 사용하는 비디오들은 전문가에 의해 가공된 데이터로 특정 행동들을 잘 나타내는 정형 비디오로 볼 수 있다. 최근 많은 미디어 플랫폼의 발달로 많은 사람이 직접 촬영한 비디오를 업로드하고 시청할 수 있게 되며, 이러한 비정형 비디오들의 수집과 접근이 쉬워졌다. 이에 따라 비디오 이해를 위해 정형 비디오를 활용하는 행동 인식 연구뿐만 아니라 비정형 비디오에 대한 시간적 행동 검출 연구가 최근 활발히 이루어지고 있다. 시간적 행동 검출은 크게 오프라인 행동 검출과 온라인 행동 검출로 분류할 수 있으며, 지난 몇 년간 두 가지 분야 모두에서 많은 행동 검출 방법들이 제안되었다. 또한, 최근 딥러닝의 발달로 인해 그 성능 또한 눈에 띄게 향상되고 있다. 이러한 연구 동향에 따라 본 논문에서는 최근 주목받고 있는 딥러닝 기반의 시간적 행동 검출 방법들을 소개하고자 한다.

키워드 : 딥러닝, 시간적 행동 검출, 오프라인 행동 검출, 온라인 행동 검출
Key Words : deep learning, temporal action detection, offline action detection, online action detection

ABSTRACT

In the field of computer vision, action recognition for video understanding has been studied for a long time. However, the videos used in action recognition are trimmed videos processed by professionals for well representing predefined actions. In recent, many people have been able to upload and watch real-world videos from the development of many media platforms. These platforms also make it easier to collect and access such untrimmed videos. As a result, for video understanding, research on temporal action detection on untrimmed videos has been actively studied recently, as well as research on action recognition on trimmed videos. Temporal action detection can be categorized into offline and online action detection, and many temporal action detection methods have been proposed in both fields over the last few years. In addition, due to the recent promising results of deep learning in computer vision, the performance of temporal action detection approaches has been remarkably improved. In this paper, we introduce deep learning-based temporal action detection methods that have recently attracted attention.

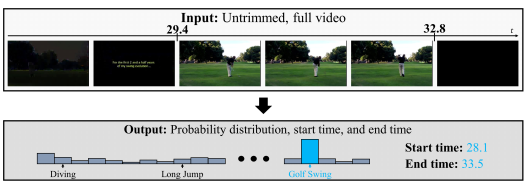
※ 이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.B0101-15-0266, 실시간 대규모 영상 데이터 이해·예측을 위한 고성능 비주얼 디스커버리 플랫폼 개발 및 No.2020-0-00004, 장기 시각 메모리 네트워크 기반의 예지형 시각지능 핵심기술 개발).
◆ First Author : School of Electrical Engineering, Korea Advanced Institute of Science and Technology, hj.eun@kaist.ac.kr, 학생회원
° Corresponding Author : Department of Electrical Engineering, Hanbat National University, peterjung@hanbat.ac.kr, 정회원
* Electronics and Telecommunications Research Institute (ETRI), {jymoon, jongyoul}@etri.re.kr
** School of Electrical Engineering, Korea Advanced Institute of Science and Technology, changick@kaist.ac.kr
논문번호 : 202004-079-A-RN, Received April 7, 2020; Revised May 9, 2020; Accepted May 15, 2020

Ⅰ. 서 론

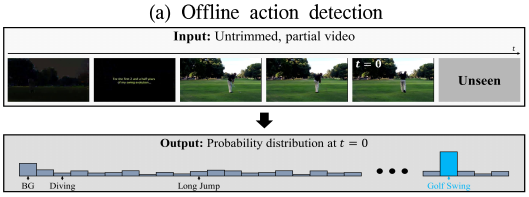
비디오 분석 연구 중 하나인 행동 인식은 오랫동안 연구되어 왔다[1,2]. 행동 인식 연구는 정형 비디오를 입력으로 하여 행동 분류를 수행한다. 여기서 정형 (trimmed) 비디오란 미리 정의된 행동만을 담도록 시간적으로 정제된 비디오를 의미한다 (그림 1). 최근 딥러닝의 발달로 행동 인식 연구는 높은 성능을 이루며, 더 어려운 비디오 분석 연구가 주목받기 시작하였다. 그중 하나가 비정형 (untrimmed) 비디오를 활용하는 시간적 행동 검출 연구이다. 그림 1에서와같이 비정형 비디오는 비디오의 시작과 끝이 행동의 시작과 끝으로 정의되지 않고 행동 외의 정보가 함께 포함되어있다. 또한, 비디오는 여러 개의 행동을 포함할 수 있다. 시간적 행동 검출 연구가 주목받기 시작한 또 하나의 이유는 최근 미디어 플랫폼의 발달이다. 많은 사람이 쉽게 비디오를 공유하고 접할 수 있게 되었으며, 이는 정제되지 않은 실제 비디오 분석을 필요하게 만들었다.

시간적 행동 검출 연구는 크게 오프라인 행동 검출과 온라인 행동 검출 두 가지로 나누어진다 (그림 2). 오프라인 행동 검출은 비정형 비디오 전체를 입력으로 하여 행동의 시작과 끝 시간을 찾고, 행동의 분류까지 수행한다. 비디오 내 여러 행동 존재, 행동의 시작과 끝 정의의 모호함, 행동 외 비관련 정보 존재 등은 오프라인 행동 검출을 어렵게 만드는 요소이다.

그림 1. 정형 비디오와 비정형 비디오의 비교
Fig. 1. Comparison between trimmed and untrimmed videos.



(a) Offline action detection



(b) Online action detection

그림 2. 오프라인 행동 검출과 온라인 행동 검출 비교
Fig. 2. Comparison between offline and online action detection.

온라인 행동 검출은 현재 프레임의 행동을 인식하는 작업으로 현재를 기준으로 미래 정보는 사용할 수 없는 스트리밍 비디오를 대상으로 한다. 이 때문에 사용할 수 있는 행동 정보가 부분적이며, 입력 비디오가 관심 있는 행동뿐만 아니라 그 외 정보도 포함할 수 있어 행동 검출에 많은 어려움이 있다.

본 논문의 구성은 다음과 같다. 2장에서는 최근 오프라인 행동 검출 방법에 대해, 3장에서는 최근 온라인 행동 검출 방법에 관해 설명한다. 4장에서는 시간적 행동 검출에 관한 향후 연구 방향과 전망을 살펴본다.

Ⅱ. 오프라인 행동 검출

오프라인 행동 검출은 보통 두 단계로 이루어진다. 첫 번째 단계는 행동의 시작과 끝으로 정의되는 행동 구간을 생성하며, 두 번째 단계는 행동 구간의 행동을 분류한다.

2.1 SCNN[3]

SCNN은 3D 컨볼루션 뉴럴 네트워크 (3D CNN: 3D Convolution Neural Network)을 기반의 방법으로 세 가지 3D CNN을 제안하며 오프라인 행동 검출을 수행한다 (그림 3). 행동 구간 제안 네트워크 (proposal network), 분류 네트워크 (classification network), 지역화 네트워크 (localization network)가 이 세 가지 네트워크이다.

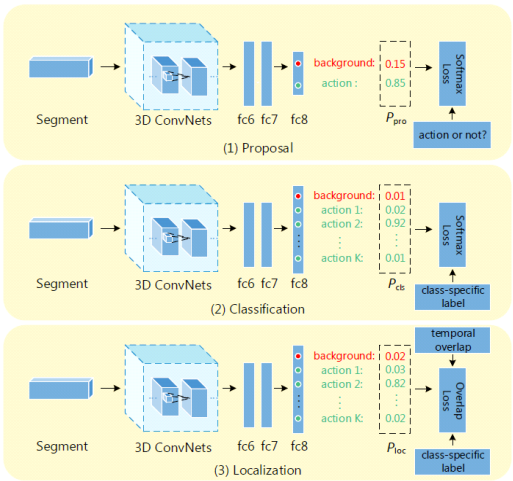


그림 3. SCNN의 세 가지 네트워크 구조[3]
Fig. 3. Three networks of SCNN[3].

SCNN은 먼저 비디오 $V=\{f\}_{t=1}^{T}$를 프레임 f로 정의하고, 다양한 길이의 프레임으로 이루어진 구간을 생성한다. 시간 축에 대해 슬라이딩 윈도우 방법을 사용하여 16, 32, 64, 128, 256, 512 프레임으로 이루어진 그룹이 생성되며 이 그룹은 75%의 겹침을 가진다. 각 그룹으로부터 균일한 샘플링을 통해 16 프레임을 추출하고 이를 하나의 구간으로 정의한다. 즉, 각 구간은 시작 시간과 끝 시간으로 정의되며 행동 구간 제안 네트워크의 입력으로 사용한다.

행동 구간 제안 네트워크는 구간의 행동 종류에 상관없이 행동인지 아닌지를 구분하는 역할을 한다. 이를 위해 정답 구간과 제안 구간 간의 시간적 교집합 영역 (tIoU: temporal Intersection-over-Union)을 이용한다. 네트워크 학습을 위한 구간들의 tIoU를 구하고, 그 값이 0.7보다 크면 1로 0.3보다 작으면 0으로 할당하여 행동과 배경을 구분하도록 네트워크를 학습한다.

다음 분류 네트워크는 행동 구간 제안 네트워크에서 구간의 행동 종류를 분류한다. 앞 행동 구간 제안 네트워크에서 배경으로 분류된 구간은 제외하고 행동으로 분류된 구간들만을 남긴다. 이를 분류 네트워크의 입력으로 사용하고, 구간의 행동 종류를 분류한다.

지역화 네트워크는 정답 구간과 높은 겹침을 가지는 구간을 찾기 위해 사용된다. 즉, 이 네트워크는 tIoU를 손실 함수로 사용하며, 각 구간의 tIoU를 추정한다.

세 가지 네트워크를 통과하고 남겨진 행동 구간들은 행동 종류에 따른 행동 길이 사전 지식을 지역화 점수에 곱하여 주고, 이 점수를 기반으로 논 맥시마 서프래션 (NMS: Non-Maxima Suppression)을 취해 중복되는 구간을 제거하며 최종 행동 객체 (action instance)를 정의한다.

SCNN은 여러 단계 및 여러 작업을 거치면서 더 나은 행동 검출 결과를 얻어낼 수 있지만, 여러 단계로 이루어진 방법은 앞 단계의 성능이 뒤 단계의 성능에 크게 영향을 미친다는 단점을 가진다. 또한, 슬라이딩 윈도우를 통해 구간을 정의하는 것은 행동의 경계를 정의하는 데 있어서 부정확하다는 문제점을 가진다.

2.2 R-C3D[4]

R-C3D는 객체 검출 네트워크 중 하나인 Faster R-CNN[13]을 기반으로 하여 시간적 행동 검출을 수행한 방법이다. 전체 구조는 그림 4에서 보여주고 있으며, 크게 행동 구간 제안 네트워크와 분류 네트워크로 이루어진다.

R-C3D는 먼저 비디오에 대해 특징을 얻기 위해 3D CNN을 이용한다. 3D CNN 중 하나인 C3D[14] 네트워크에 비디오를 입력으로 넣어 비디오에 대한 특징을 얻는다. 이 특징은 행동 구간 제안 네트워크와 분류 네트워크 모두에서 입력으로 활용된다.

행동 구간 제안 네트워크는 C3D 특징을 입력으로 행동 구간을 생성한다. Faster R-CNN과 동일한 메카니즘으로 이루어지며, 다양한 길이의 앵커 (anchor) 구간을 정의하여 tIoU를 기반으로 하여 행동 구간 여부를 예측한다. SCNN의 행동 구간 제안 네트워크와 유사하게 tIoU가 0.7 이상이면 행동 구간으로 0.3 이하이면 배경으로 분류한다.

다음 행동 구간 제안 네트워크에서 행동 구간으로 분류된 앵커 구간 중에서 중복되는 행동 구간을 제거하기 위해 NMS 수행한 후 남겨진 행동 구간은 분류 네트워크의 입력이 된다. 실제 네트워크의 입력은 행동 구간의 시작과 끝으로 정의되는 구간의 C3D 특징으로 3D RoI (Region of Interest) Pooling을 통해 동일한 크기의 행동 구간 특징이다. 분류 네트워크에서는 이 행동 구간 특징을 통해 행동 시작과 끝 시간의 회귀와 행동 종류에 대한 분류를 동시에 수행하여 최종 행동 객체를 정의한다.

R-C3D는 End-to-End 방법으로 세 단계로 이루어지는 SCNN보다 효율적이다. 하지만 슬라이딩 윈도우와 유사하게 사전에 정의된 앵커 구간을 사용하기 때문에 행동 경계가 부정확하다는 동일한 문제점을 가진다.

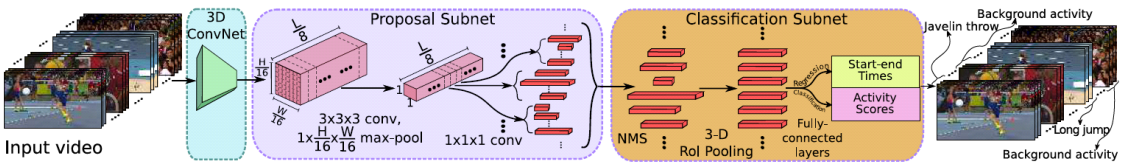


그림 4. R-C3D 모델 구조[4]
Fig. 4. R-C3D model architecture[4].

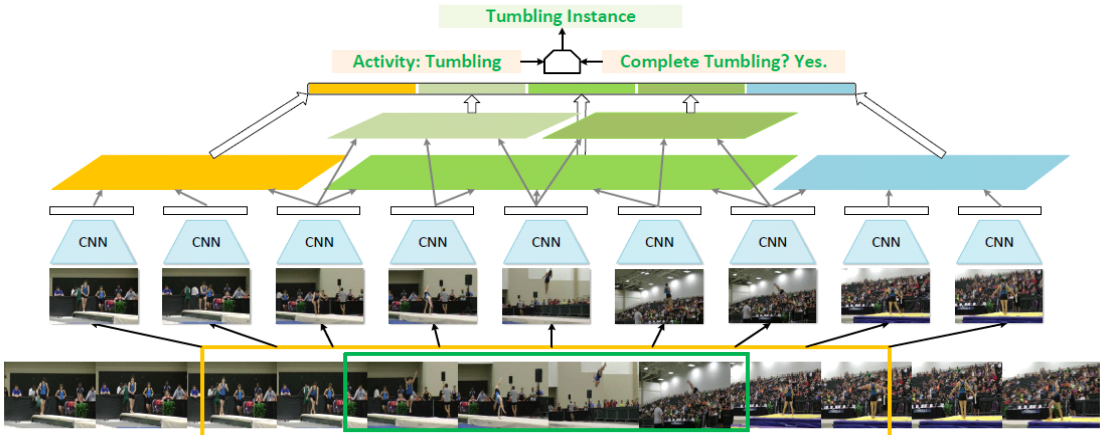


그림 5. SSN의 분류 네트워크 구조[5]
Fig. 5. Classification network architecture of SSN[5].

2.3 SSN[5]

SSN은 크게 두 단계로 이루어진다. 앞서 설명한 방법과 유사하게 첫 번째 단계에서는 행동 구간을 생성하며, 두 번째 단계에서는 행동 구간의 분류를 수행한다.

행동 구간 생성을 위해 SSN은 먼저 비디오 $V=\{s\}_{t=1}^{T}$를 단편 s로 정의하며, 여기서 단편은 연속된 소수의 프레임 집합을 나타낸다. 하나의 단편이 포함하는 프레임 수는 보통 6 프레임으로 정의한다. SSN에서는 SCNN, R-C3D에서 사용하는 슬라이딩 윈도우 방법과는 다르게 각 단편의 행동 점수를 추정한 후 높은 점수를 가지는 단편을 그룹화하여 행동 구간을 정의한다. 다시 말하면, 하나의 비디오에 대해 각 단편의 행동 점수를 추정하게 되면 하나의 1D 점수 신호를 얻을 수 있다. 이 1D 점수 신호에서 임계값을 정하여 임계 값보다 높은 점수를 가지는 단편을 그룹화한다. 그룹화 방법은 Watershed[6] 방법을 이용하여 다양한 행동 구간이 생성되도록 한다. 이렇게 단편의 점수를 기반으로 행동 구간을 생성하게 되면 슬라이딩 윈도우 방법과 비교하여 더 유연하고 정확한 행동 경계를 얻을 수 있다는 장점이 있다.

단편 기반의 행동 구간 생성 방법 외에 SSN은 독창적인 분류 네트워크 구조를 제안한다 (그림 4). 먼저 행동 구간을 시작, 중간, 끝 3개로 구분하고, 각 부분을 따로 인코딩 (encoding) 한다. 또한, 중간 부분을 두 개의 보조 부분으로 나누어 인코딩을 수행한다. 이렇게 총 5개의 부분을 인코딩한 후 각 부분의 특징을 연결하여 행동 구간에 대한 하나의 특징을 얻는다. SSN의 분류 네트워크의 경우 많이 사용되는 행동 구간의 행동 분류를 위한 손실 함수, 경계에 대해 회귀 손실 함수를 사용한다. 그 외에도 완전성 손실 함수를 함께 사용한다. 각 손실 함수는 아래와 같이 정의된다.

$$
\begin{aligned}
L_{classification} &= -\log P(c_i|p_i) \\
L_{boundary} &= \begin{cases} 0.5x_i^2, & \text{if}\,|x_i|<1 \\ |x_i|-0.5, & otherwise \end{cases} \\
L_{completeness} &= -\log P(b_i|c_i\,p_i)
\end{aligned}
\tag{1}
$$

$P(c_i|p_i)$는 조건부 확률로 c_i는 행동 구간 p_i의 행동 클래스로, 행동 구간 p_i의 각 행동 클래스일 확률을 나타내게 된다. $L_{classification}$은 교차 크로스 엔트로피 손실로 정의한다. $L_{boundary}$는 Smooth L1 손실로 정의하며, x_i는 실제 행동 구간의 중심 위치와 예측한 중심 위치의 차이, 행동 구간의 길이의 차이로 정의한다. 해당 손실 함수를 통해 실제 행동 구간의 중심 위치와 길이와 차이가 나는 경우 패널티를 줄 수 있다. $L_{completeness}$ 역시 교차 크로스 엔트로피 손실로 정의하며, 행동 구간 p_i의 완전성 여부 b_i에 대한 조건부 확률로 정의한다. 이때 고려되는 행동 구간 p_i는 행동 클래스 c_i가 배경이 아닌 행동으로 정의될 때이다. 완전성 손실 함수는 행동 구간의 많은 부분이 행동으로 구성되어 있지만, 실제 행동 객체와 tIoU를 측정하였을 때 그 값이 낮은 행동 구간을 제외하기 위해 사용한다. 예를 들어 행동 구간의 80%가 행동이지만 실제 정답 행동 구간과의 tIoU가 0.3 이하이면 불완전 행동 구간으로 정의한다.

2.4 TAL-Net[7]

TAL-Net은 R-C3D와 동일하게 Faster R-CNN 방법에 기반을 둔다. 하지만 R-C3D와는 다르게 세 가지 관점에서 Faster R-CNN을 행동 검출에 맞춰 변환시킨 방법을 제안한다 (그림 6).

첫 번째는 '어떻게 다양한 행동 길이를 효과적으로 고려할까?'이다. Faster R-CNN의 영상에서 객체 검출은 객체의 크기가 크게 변하지 않는다. 하지만 비디오에서 행동 검출은 행동 종류에 따라서도 길이 변화가 크지만 동일한 행동에서도 상대적으로 큰 변화를 가진다. 이는 직접적으로 학습에 어려움을 주며, RoI Pooling시에도 특징의 변형을 발생시킨다. 이를 해결하기 위하여 TAL-Net은 멀티-타워 구조와 시간적 팽창 컨볼루션 (dilated temporal convolution)을 사용한다. 구체적으로 설명하면 다양한 간격의 시간적 팽창 컨볼루션을 수행하여 다양한 크기의 수용 영역 (receptive field)를 가지는 특징을 멀티-타워 형태로 생성한다. 이를 통해서 다양한 길이를 가지는 행동을 효과적으로 다룰 수 있다.

두 번째는 '어떻게 시간적 문맥을 고려할까?'이다.

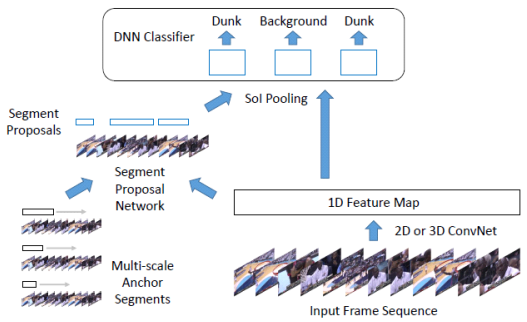


그림 6. TAL-Net 구조[7]
Fig. 6. TAL-Net architecture[7].

영상에서 객체의 경계는 쉽고 객관적으로 정의할 수 있다. 하지만 비디오에서 행동의 경계는 주관적이며 정의하기가 어렵다. 그 때문에 행동의 경계에서 문맥의 고려는 행동 검출의 중요한 요소이다. 이를 고려하기 위해 시작과 끝 영역을 확장하여 문맥 영역을 정의하고 앵커의 크기가 문맥 영역을 포함할 수 있도록 조정한다.

세 번째는 '멀티-스트림 (multi-stream) 특징 결합'이다. 최신 행동 인식 방법은 RGB와 광학 흐름 (optical flow) 특징을 결합하여 높은 성능을 이루었다. Faster R-CNN 기반 행동 검출에서 어떠한 특징 결합 방법이 효과적인지 분석이 필요하다. TAL-Net은 RGB와 광학 흐름 각 특징이 시간적 행동 검출에 중요하고, 다른 행동 인식 방법들과 유사하게 끝 단에서의 특징 결합이 더 효과적인 것을 보여준다.

2.5 PBRNet[17]

PBRNet은 시간적 행동 검출에서 행동 경계가 모호하다는 문제에 초점을 맞춘다. PBRNet은 3개의 검출 단계 구성되어 있으며, 각각은 개략 피라미드 검출 (coarse pyramidal detection), 정제 피라미드 검출 (refined pyramidal detection), 세밀 피라미드 검출 (fine-grained detection)이다 (그림 7).

개략 피라미드 검출에서는 기본적인 단일 단계 객체 검출 방법을 차용한다. 즉, 사전에 시간 축으로 앵커 구간을 정의하고 해당 구간에 대해 분류와 회귀를 수행하여 행동의 클래스와 경계를 예측한다.

다음 정제 피라미드 검출에서는 개략 피라미드 검출에서 사용된 특징을 가져와 함께 사용한다. 또한, 시간적 해상도를 단계적으로 향상해 더 정확한 행동 구간 경계를 찾고자 한다.

마지막 세밀 피라미드 검출에서는 마찬가지로 피라

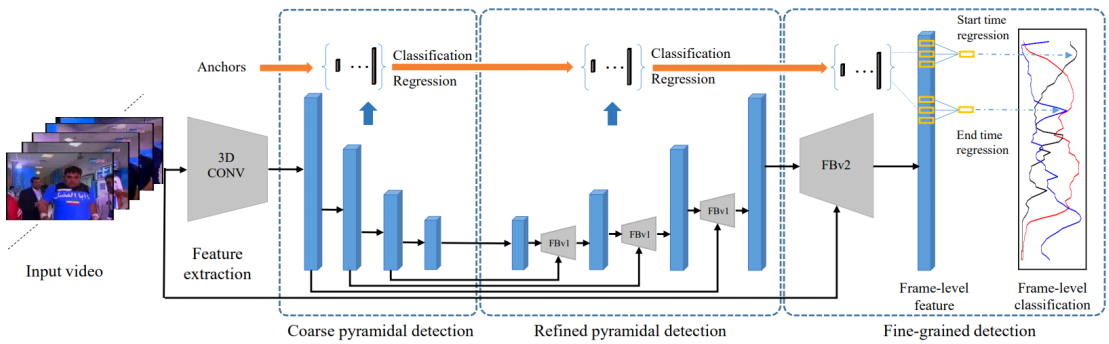


그림 7. PBRNet 구조[17]
Fig. 7. PBRNet architecture[17].

미드 구조를 사용하며, 행동 구간 단위의 예측이 아닌 프레임 단위의 예측을 통해 행동 구간의 시작과 끝 점수를 추정하여 개선된 행동 구간을 정의한다. 이러한 반복적인 행동 구간 경계의 개선을 통해 더욱 정확한 경계를 얻을 수 있다.

2.6 P-GCN[18]

그림 8과 같이 P-GCN은 그래프 컨볼루션 네트워크 (GCN: Graph Convolutional Network)[25]를 도입하여 행동 구간 간의 관계를 모델링 하는 네트워크이다. GCN은 비격자 구조에 대한 컨볼루션을 정의하기 위해 제안된 신경망의 종류 중 하나이다. P-GCN은 먼저 기존 행동 구간을 제안하는 다양한 방법을 통해 행동 클래스가 정의되지 않은 행동 구간을 얻고, 이를 입력으로 사용한다.

GCN에서 그래프는 $G(V,E)$로 정의하며, V는 노드의 집합, E는 노드와 노드를 연결하는 에지의 집합으로 정의한다. P-GCN에서는 V는 각 행동 구간으로 정의하며 E는 행동 구간 간의 관계로 정의할 수 있다. 또한, 행동 구간 간의 연결성을 나타내는 인접행렬 A를 정의한다. P-GCN은 그래프 구조를 통해 각 행동 구간이 인접한 행동 구간의 정보 취합을 통한 성능 향상을 기대한다.

P-GCN의 그래프를 정의하기 위해서는 에지 E를 정의가 필요하다. 해당 방법에서는 문맥 에지 (Contextual Edge)와 주변 에지 (Surrounding Edge)를 제안한다. 문맥 에지는 행동 구간 간의 겹침을 이용하여 일정 기준 이상 겹침이 있는 경우 에지를 생성한다. 즉, 겹쳐진 행동 구간일수록 서로 유용한 정보를 가진다는 것을 반영한다. 다음 주변 에지는 행동 구간 중심 위치 간의 시간적 거리를 계산하여 일정 기준 이하면 에지를 생성한다. 즉, 시간적으로 가까운 행동 구간이 연결되어야 함을 의미한다. 두 종류의 에지를 통해 인접행렬 A를 정의할 수 있으며, 이를 통해 P-GCN의 그래프를 최종 정의하게 된다. P-GCN의 학습은 일반적인 교차 크로스 엔트로피 손실 함수를 이용하여 행동 구간의 행동 분류를 학습하도록 하며, 정확한 경계를 가지는 행동 구간과 확장된 경계를 가지는 행동 구간 두 가지에 대해 각각 GCN을 학습하게 된다.

2.7 GTAN[19]

GTAN은 행동 구간의 길이가 너무나 다양하다는 문제점을 해결하기 위해, Gaussian 동작을 소개한다. Gaussian 동작은 동적으로 행동 구간의 시간적 크기를 최적화하는 역할을 한다.

GTAN의 전체 구조는 그림 9와 같으며, 전체 구조는 R-C3D와 유사하다. 먼저 3D CNN과 1D CNN을 통하여 입력 프레임에 대해 특징을 추출한다. 전체 네트워크는 총 7개의 max pooling을 수행하며 시간적으로 수용 영역을 확장한다. Gaussian 동작은 max pooling을 수행하기 전 특징에 매번 적용하게 되며, Gaussian 동작 후 행동 구간 예측을 수행하여 다양한 크기의 시간적 수용 영역을 기반으로 하는 결과를 얻을 수 있다.

Gaussian 동작은 Gaussian 컨볼루션, Gaussian 그룹화, Gaussinan pooling으로 이루어진다. Gaussian 컨볼루션은 1D 컨볼루션을 기반으로 하며, Gaussian 변수인 표준편차를 학습한다. 다음 Gaussian 그룹화에서는 Gaussian 컨볼루션에서 생성한 Gaussian 커널들을 겹친 정도에 기반을 두어 그룹화를 수행하게 된다. Gaussian pooling에서는 앞 두 단계를 통해 얻은

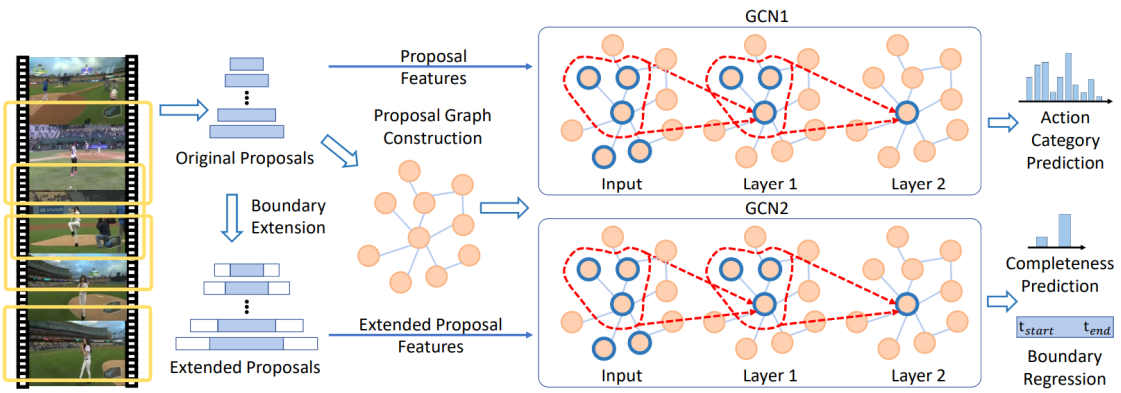


그림 8. P-GCN 구조[18]
Fig. 8. P-GCN architecture[18].

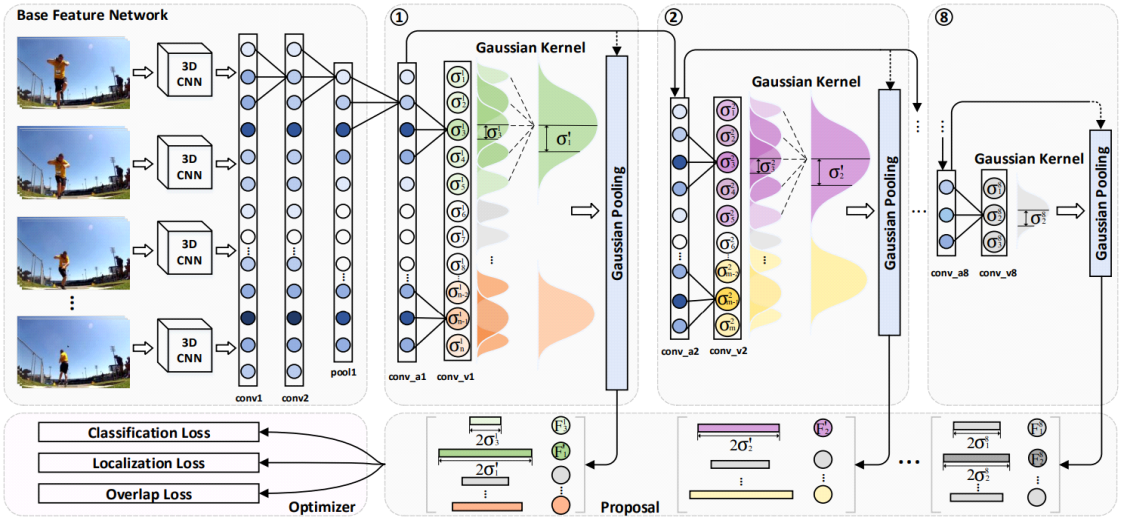


그림 9. GTAN 구조[19]
Fig. 9. GTAN architecture[19].

Gaussian 커널들을 이용해 동적인 시간적 앵커 구간을 취득한다. Gaussian 동작을 통해 얻은 앵커 구간은 기존 R-C3D의 고정된 크기의 앵커 구간보다 우수한 다양한 행동 구간 길이 모델링을 수행할 수 있다.

GTAN은 3가지 손실 함수를 사용하여 학습한다. 먼저 분류 손실 함수는 다른 방법과 동일하게 교차 크로스 엔트로피 손실을 사용하여 행동 구간의 분류를 학습한다. 다음 지역화 손실 함수도 역시 Smooth L1 손실을 사용하여 행동 구간의 중심과 길이에 대해 회귀를 수행하여 학습한다. 마지막으로 겹침 손실 함수는 평균 제곱 오류 (MSE: Mean Square Error)를 이용하여 실제 행동 구간과 예측한 행동 구간의 겹침을 학습하도록 한다.

2.8 MSCA-Net[20]

MSCA-Net 역시 다양한 길이의 행동 구간을 모델링하는 것이 오프라인 행동 검출에 중요한 요소라고 주장한다. 이를 위하여, MSCA-Net은 다양한 크기의 3D RoI pooling을 수행하며, 다양한 길이의 수용 영역에 기반 특징을 생성하는 것을 주요 목표로 한다.

그림 10은 MSCA-Net의 전체 구조이다. MSAC-Net은 먼저 R-C3D를 이용하여 후보 행동 구간을 생성하고, 이를 MSCA-Net의 입력으로 사용한다. 생성된 행동 구간에 대한 특징을 얻기 위해 3D

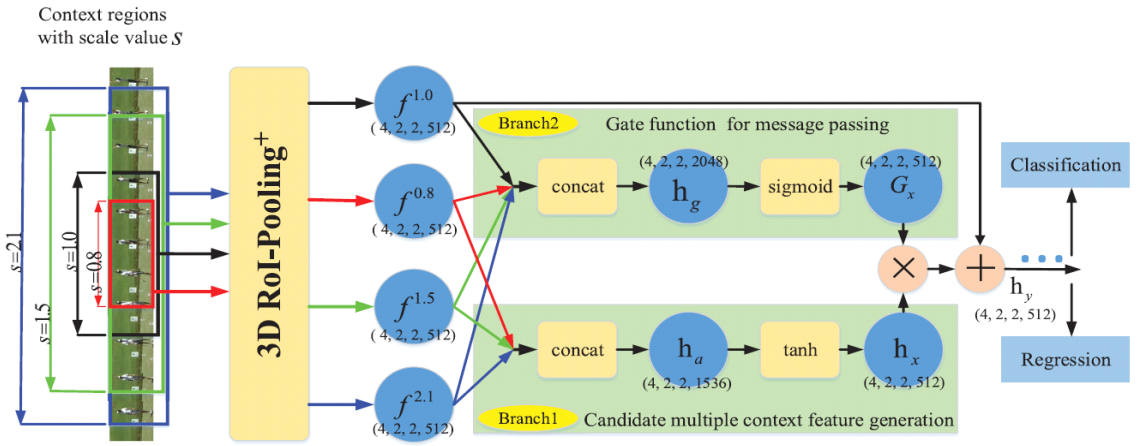


그림 10. MSCA-Net 구조[20]
Fig. 10. MSCA-Net architecture[20].

RoI pooling이 이용한다. 여기서 정확한 행동 구간에 대해서만 3D RoI pooling을 수행하는 것이 아니라, 행동 구간 길이의 0.8배, 1.5배, 2.1배의 길이에 대해서도 3D RoI pooling을 수행하여 총 4개 길이의 특징 $f^{0.8}, f^{1.0}, f^{1.5}, f^{2.1}$을 얻는다.

4개 길이의 특징은 병렬적으로 2가지 단계를 거치게 되는데 첫 번째는 특징의 주의 집중 (attention)을 생성하는 단계이다. 이 단계에서는 $f^{0.8}, f^{1.0}, f^{1.5}, f^{2.1}$를 연결한 후 3D 컨볼루션과 시그모이드 (sigmoid) 활성 함수를 이용하여 어떤 특징에 집중할 것인지를 나타내는 가중치 G_x를 생성한다.

두 번째 단계는 다양한 길이의 문맥을 포함하는 특징을 생성한다. 각 $f^{0.8}, f^{1.5}, f^{2.1}$는 3D 컨볼루션과 ReLU 활성 함수를 거치게 된다. 그 후 세 가지 특징을 연결하여 특징 h_x을 생성한다. 최종 특징 $h_y = f^{1.0} + \gamma h_x G_x$로 정의한다. 이로써 최종 특징은 다양한 길이의 문맥 특징을 적응적으로 활용할 수 있다.

MSCA-Net의 학습은 다른 방법과 동일하게 행동 구간의 클래스에 대해 교차 엔트로피 손실 함수를 정의하고 행동 구간의 경계를 학습하기 위해 Smooth L1 손실 함수를 이용한다.

2.9 비교

2.9.1 요약

표 1은 앞서 설명한 8개의 네트워크, SCNN[3], R-C3D[4], SSN[5], TAL-Net[7], PBRNet[17], P-GCN[18], GTAM[19], MSCA-Net[20]에 대해 요약한다. 각 네트워크에 따라 키 아이디어, 행동 구간 후보 생성 방법, 학습 시 사용한 손실 종류에 대해 정리하였다.

2.9.2 데이터셋

오프라인 행동 검출 연구에서는 두 개의 공식 데이터셋을 사용한다. 첫 번째 데이터셋인 THUMOS-14[8]

표 1. 오프라인 행동 검출 네트워크 요약 비교
Table 1. Summary on offline action detection networks.

Network	Key Idea	Proposal Generation	Losses
SCNN[3]	-Three networks (proposal generation, classification, localization) -Cascaded structure for three networks	sliding window	-action classification -boundary regression -action/non-action classification
R-C3D[4]	-Modifying Faster R-CNN[13] -Converting 2D bounding box detection to 1D temporal boundary detection -Change 2D images to 3D videos for input	temporal anchor	-action classification -boundary regression -action/non-action classification
SSN[5]	-Modeling action temporally by dividing a proposal into start, middle, middle-start, middle-end, and end -Learning action completeness	snippet actionness	-action classification -action completeness classification
TAL-Net[7]	-Modifying Faster R-CNN -Changing 2D image to 1D video features for input -Modeling various temporal length with multi-tower and dilated temporal convolution -Combining features based on a multi-stream structure	temporal anchor	-action classification -boundary regression -action/non-action classification
PBRNet[17]	-Cascaded structure for coarse pyramidal detection, refined pyramidal detection, and fine-grained detection -Pyramidal architecture	temporal anchor	-action classification -boundary regression
P-GCN[18]	-Using graph convolution networks -Modeling relation between temporal action proposals	pre-generated proposals	-action classification -boundary regression -action completeness classification
GTAN[19]	-Modifying Faster R-CNN -Obtaining anchors by using Gaussian operations (Gaussian convolution, Gaussian grouping, and Gaussinan pooling)	temporal anchor	-action classification -boundary regression -action overlap regression
MSCA-Net[20]	-Using various sizes of 3D RoI pooling -Attention module for the contextual information of various action length	pre-generated proposals	-action classification -boundary regression

는 2,584개의 비정형 비디오를 포함하고 있다. 그중에서 412개 비디오에 대해 20개 스포츠 액션에 대해 정답을 제공한다. 412개의 비디오 중 200개를 학습에 사용하고, 나머지 212개를 테스트에 사용한다. THUMOS-14 데이터셋의 각 비디오는 평균 15.8개의 행동 객체, 그리고 71% 배경을 포함하고 있다.

두 번째 데이터셋은 ActivityNet-1.3[9] 이다. 19,994개의 비디오로 이루어져 있으며, 200개의 행동에 대해 실제 행동 구간 정보를 제공한다. 학습셋 10,024개 비디오, 확인셋 4,926개 비디오, 테스트셋 5,044개 비디오로 이루어져 있다. 또한, THUMOS-14에 비해 전반적으로 행동 구간의 길이가 길다.

2.9.3 성능평가지표

오프라인 행동 검출의 성능평가지표는 tIoU에 따른 mAP (mean Average Precision)를 사용한다. 즉, tIoU를 변화시켜가며 검출한 행동 객체 구간이 실제 행동 객체 구간과 일치하는지를 판별한다.

2.9.4 성능 비교

표 2는 앞에서 설명한 방법들의 THUMOS-14 데이터셋에 대한 성능 비교 표이다. tIoU 0.1에서 0.4 구간에서는 P-GCN[17]이 가장 높은 성능을 보여주고 있으며 tIoU가 0.5인 경우에는 PBRNet[17]이 가장 높은 성능을 나타낸다. THUMOS-14 데이터셋은 복수의 행동 구간이 반복적으로 나타나는 특징을 가지고 있어, 복수 행동 구간의 관계를 모델링하는 P-GCN이 높은 성능을 나타내며, PBRNet의 경우 반복적으로 경계를 개선하기 때문에 높은 tIoU에서 좋은 성능을 보여준다고 할 수 있다. 표 3은 ActivityNet-1.3에 대한 성능 비교이며, PBRNet[17]과 GTAN[19]이 높은 성능을 보여준다. ActivitNey-1.3의 경우 길이가 길며 단일 행동 구간을 가지는 비디오가 높은 비중을 차지한다. 이 때문에 복수 행동 구간에 적합한 P-GCN보다는 행동 구간의 경계에 초점을 맞춘 PBRNet과 GTAN이 높은 성능을 보여준다.

표 2. THUMOS-14[8] 데이터셋에 대한 오프라인 행동 검출 성능 비교
Table 2. Performance comparison of offline action detection on THUMOS-14[8].

tIoU	0.1	0.2	0.3	0.4	0.5
SCNN[3]	47.7	43.5	36.3	28.7	19.0
R-C3D[4]	54.5	51.5	44.8	35.6	28.9
SSN[5]	66.0	59.4	51.9	41.0	29.8
TAL-Net[7]	59.8	57.1	53.2	48.5	42.8
PBRNet[17]	-	-	58.5	54.6	51.3
P-GCN[18]	69.5	67.8	63.6	57.8	49.1
GTAN[19]	69.1	63.7	57.8	47.2	38.8
MSCA-Net[20]	-	-	58.4	-	41.8

표 3. ActivityNet-1.3[9] 데이터셋에 대한 오프라인 행동 검출 성능 비교
Table 3. Performance comparison of offline action detection on ActivityNet-1.3[9].

tIoU	0.5	0.75	0.95	Avg.
R-C3D[4]	26.80	-	-	-
SSN[5]	43.26	28.70	5.63	28.23
TAL-Net[7]	38.23	18.30	1.30	20.22
PBRNet[17]	53.96	34.97	8.98	35.01
P-GCN[18]	48.26	33.16	3.27	31.11
GTAN[19]	52.61	34.14	8.91	35.54
MSCA-Net[20]	30.20	-	-	-

Ⅲ. 온라인 행동 검출

비디오 전체를 본 후 행동 구간을 찾고 종류를 분류하는 오프라인 행동 검출과 달리 온라인 행동 검출은 지금까지 들어온 프레임만을 입력으로 하여 현재 프레임의 행동을 인식하는 작업이다. 입력의 경우 비정형, 스트리밍 비디오이기 때문에 불필요한 정보를 구별하여야 하며 불완전한 정보를 다루어야 하여 더 어려운 작업이라고 말할 수 있다.

3.1 RED[11]

Gao 등[11]은 온라인 행동 검출을 위해 강화 인코더-디코더 (RED: Reinforced Encoder-Decoder) 네트워크를 제안하였다. 그림 11에 나타나듯이 RED는 LSTM (Long Short-Term Memory) 기반의 Sequence-to-Sequence 순환 신경망 네트워크 (RNN: Recurrent Neural Network)를 기반으로 한다. RED의 인코더 부분의 각 LSTM은 현재부터 일정 과거 시간까지의 정보를 입력으로 받고, 디코더 부분에서는 미래 시간의 행동을 예측한다. 인코더의 각 LSTM은 TSN[15]으로부터 생성한 각 프레임의 특징을 실제 입력으로 사용한다. LSTM 및 RNN의 구조 특성상 연속적인 데이터를 잘 학습할 수 있다는 장점이 있다. 디코더에서는 RED의 학습을 위한 장치가 구성되며, 로스는 각 미래 프레임의 행동 인식하기 위한 교차 엔트로피 손실 (cross entropy loss), 디코더의 학습을 돕기 위한 특징의 제곱 손실 (sequaed loss), 강화 모듈 관련 로스로 이루어진다.

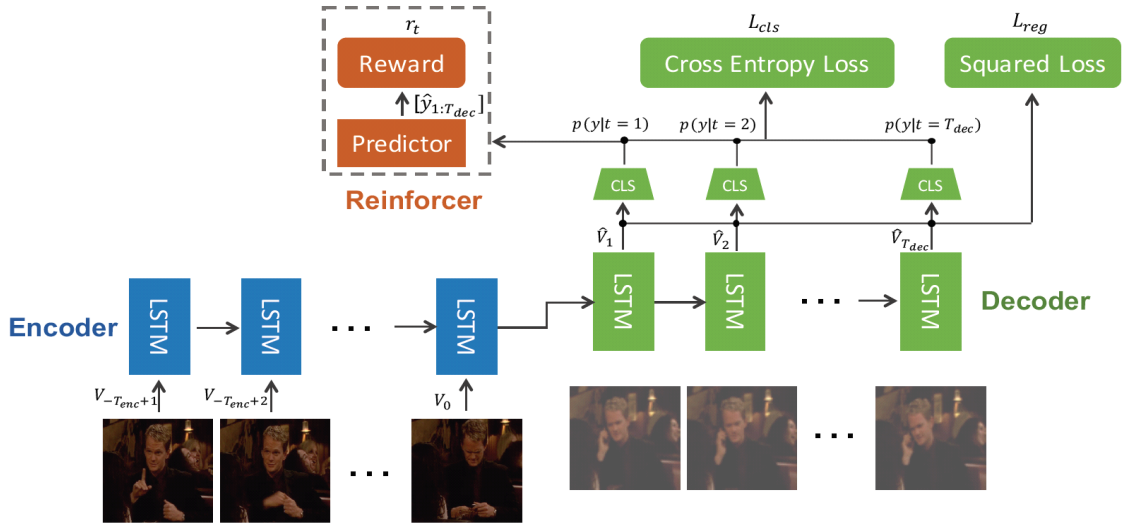


그림 11. RED 네트워크 구조[11]
Fig. 11. RED network architecture[11].

3가지 로스에 대해 조금 더 상세히 설명하도록 한다. 교차 엔트로피 로스는 행동 분류를 하기 위해 소프트맥스 (softmax) 함수를 기반으로 정의된다. 제곱 손실은 디코더의 각 LSTM에서 나오는 특징 V_t을 효과적으로 학습하기 위한 손실이다. 먼저 실제 미래 프레임의 특징 $\hat{V}_t$를 TSN으로부터 생성하고, 식 $L = \| \hat{V}_t - V_t \|$ 과 같이 L_2 Norm을 이용하여 두 특징의 차이를 학습한다. 세 번째 강화 모듈의 목적은 배경에서 행동으로 변화를 빠르게 검출하기 위함이다. 기존 교차 엔트로피 로스만을 사용한다면, 미래 프레임의 실제 행동 분류가 "011111"일 때, 예측한 행동 분류가 "000111"이거나 "001110"일 때 모두 동일한 손실을 가진다. 여기서 "0"은 배경, "1"은 행동을 의미한다. 하지만 두 결과 중 행동을 빠르게 검출하는 것이 더 나은 결과이기 때문에 이를 구분하고자 강화 손실을 정의한다. RED는 위 세 가지 손실을 합친 손실 함수 이용하여 학습하게 된다. 해당 논문에서는 강화 모듈을 제거한 네트워크인 ED와 RED를 비교하여 강화 모듈의 유효성을 입증한다.

3.2 TRN[12]

TRN은 온라인 행동 검출의 입력이 불완전 행동을 포함한다는 것에 초점을 맞추었다. RED와 동일하게 Sequence-to-Sequence RNN를 사용하지만, 미래 정보를 예측하여 현재 행동 인식에 과거 정보와 예측한 미래 정보를 함께 사용한다 (그림 12).

TRN의 각 셀 (cell)은 입력 프레임의 행동 분류를 위해 교차 엔트로피 손실을 이용하여 학습된다. 또한, 각 셀은 정해진 시간만큼의 미래를 예측하며, 각 미래 시간의 행동 분류를 위한 교차 엔트로피 로스도 함께 사용한다. 미래 시간의 행동을 예측하며 학습한 특징은 다시 현재 프레임에서 생성된 특징과 결합하여 현재 행동을 분류하는 데 이용하도록 네트워크가 구성된다.

TRN의 경우 현재까지의 정보를 이용해 미래 정보를 예측하고, 예측한 미래 정보와 현재 정보를 결합하여 현재의 행동을 검출하고자 하였다. 하지만 입력으로 약 4초 길이의 과거 정보를 사용하는 RED와는 비교되게 TRN은 4배인 16초 길이의 과거 정보를 사용하여 효율성이 떨어진다는 단점이 있다.

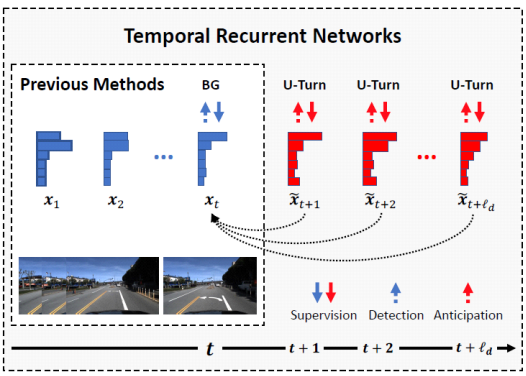


그림 12. TRN 네트워크 구조[12]
Fig. 12. TRN network architecture[12].

3.3 FV-SVM[10]

Geest 등[10]은 온라인 행동 검출 테스크를 처음 제안하며, 여러 기본 네트워크를 실험 비교하였다. 기본 네트워크로 일반적인 CNN과 LSTM 기반 RNN 외에도 Fisher vector (FV)[16]와 SVM을 기반으로 한 방법을 제안하였다. 해당 방법은 먼저 궤도, HOG, HOF, MBH의 특징을 계산한다. 그리고 이를 이용하여 FV를 계산하여 특징으로 사용한다. 다음 선형 one-vs-all SVM을 이용하여 계산한 특징을 분류하여 온라인 행동 검출을 수행한다. 여기서 하나의 SVM만을 사용하는 것이 아니라 20, 40, 60, 80 프레임에 대해서 각각 SVM을 학습시켜 행동 분류 점수를 얻고 max-pooling을 통해 최종 결과를 얻는다.

3.4 2S-FN[17]

2S-FN에서는 행동 간의 관계가 온라인 행동 검출에 중요하다고 주장하며, 행동 간의 긴 시간적 구간 의존성을 모델링 하고자 하였다. 이를 위하여 LSTM 기반 모델을 제안하였다. 하지만 기본 LSTM은 입력의 번역과 시간적 모델링이 어렵다고 가정하에 두 개 흐름을 기반으로 하는 LSTM을 설계한다. 그림 13과 같이 하나의 LSTM 흐름은 입력을 번역하는 데 초점을 맞춘다. 이후 행동 분류를 위해 임베딩 된 입력 프레임의 특징은 다른 하나의 LSTM의 입력으로 사용된다. 두 번째 LSTM의 목적은 이미 임베딩 된 특징 간의 시간적 모델링이다. 즉, 각 LSTM이 하나의 특화된 일을 할 수 있도록 구조를 설계하였으며, 이는 각 LSTM의 능력을 향상시킬 수 있다.

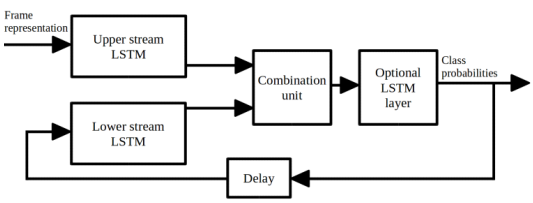


그림 13. 2S-FN 네트워크 구조[12]
Fig. 13. 2S-FN network architecture[12].

3.5 비교

3.5.1 요약

표 4는 앞서 설명한 4개의 방법, RED[11], TRN[12], FV-SVM[10], 2S-FN[17],에 대해 요약한다. 각 네트워크에 따라 키 아이디어, 학습 시 사용한 손실 종류에 대해 정리하였다.

3.5.2 데이터셋

온라인 행동 검출 연구에서는 두 개의 데이터셋을 사용한다. 첫 번째 데이터셋은 THUMOS-14[8]로 오프라인 행동 검출에서 설명하여 생략하도록 한다.

두 번째는 TVSeries[10] 데이터셋으로 미국 드라마 27개의 에피소드로 이루어져 있으며, 전체 16시간의 분량이다. 해당 데이터셋은 30개의 실제 행동에 대해 정답이 제공된다. 예를 들면, 물건 집기 (pick up), 문 열기 (open door), 운전 (drive), 마시기 (drink) 등이 있다. TVSeries 데이터셋은 다양한 배우들이 한 장면에 나타나고, 행동의 가림, 비디오 내의 많은 양의 배경 프레임, 제한되지 않은 시점 등으로 인해 어려운 데이터셋으로 평가받고 있다.

표 4. 온라인 행동 검출 방법 요약 비교
Table 4. Summary on online action detection methods.

Network	Key Idea	Losses
RED[11]	-Sequence-to-Sequence RNN -A reinforcement module for detecting action as soon as it happens	-action classification (current, future) -feature regression -reinforcement-based action sequence
TRN[12]	-Sequence-to-Sequence RNN -Using observed past and current information and predicted future information together	-action classification (current, future) -feature regression
FV-SVM[10]	-Using FV[16] and SVM -Computing FV using hand-crafted features (HOG, HOF, and MBH)	-action classification
2S-FN[17]	-LSTM-based two-stream structure -Embedding features from the first LSTM -Modeling temporal information from the second LSTM	-action classification

3.5.3 성능평가지표

온라인 행동 검출의 성능평가지표는 프레임 단위의 mAP를 사용한다. 프레임 단위의 mAP의 계산은 크게 두 단계로 이루어진다.

첫 번째는 각 행동 종류에 대해 AP를 계산한다. 행동 하나를 정하고, 모든 프레임의 해당 행동 점수를 내림차순으로 정렬한다. 그리고 프레임의 수를 하나씩 증가시켜가며 AP를 계산한다.

$$AP=\frac{\sum_n \mathrm{Prec}(n) I(n)}{P} \tag{1}$$

식 (1)에서 $\mathrm{Prec}(n)$은

$$\mathrm{Prec}(n)=\frac{TP(n)}{(TP(n)+FP(n))} \tag{2}$$

으로 계산한다. n은 계산 시 사용되는 프레임의 수이며, P는 실제 행동 프레임의 수이다. $I(n)$은 지시자로 프레임 n이 정탐이면 1로 그렇지 않으면 0으로 설정한다. AP를 모든 행동 종류에 대해 계산을 하며, 두 번째 단계에서는 모든 AP를 평균하여 mAP를 계산한다.

Geest 등[10]은 mAP가 실제 비행동 프레임과 실제 행동 프레임의 차이를 고려하지 못한다는 단점을 해결하고자 AP대신 cAP (calibrated Average Precision)을 제안한다. cAP는 조정 정확도 (calibrated precision)에 의해 계산되며 다음과 같이 계산한다.

$$cPrec(n)=\frac{wTP(n)}{(wTP(n)+FP(n))} \tag{3}$$

여기서 w는 비행동 프레임과 행동 프레임의 비율로 정의한다.

3.5.4 성능 비교

표 5는 앞에서 설명한 방법들 및 기초 방법들을 THUMOS-14 데이터셋에 대한 성능을 비교한다. RGB와 광학 흐름을 입력으로 사용하였으며, TRN이 가장 높은 성능을 보여준다. 이를 통해 예측을 통해 얻은 미래 정보와 과거 정보를 함께 활용하는 것이 성능 개선에 도움이 됨을 확인할 수 있다. 표 6은 TVSeries 데이터셋에 대한 성능 비교이다. 다양한 입력에 대해 성능 비교를 하고 있으며, 모든 입력 종류에서 TRN이 가장 높은 성능을 보여주고 있다. 이를 통해 THUMOS-14에 대한 실험과 동일하게 미래 정보의 활용이 온라인 행동 검출에 중요하다는 것 확인할 수 있다. 또한, 광학 흐름만을 사용하는 FV-SVM의 성능이 RGB만을 사용하는 방법보다 전반적으로 높은 것을 보아 행동 검출에 광학 흐름이 중요한 특징이라고 말할 수 있다.

표 5. THUMOS-14[8] 데이터셋에 대한 온라인 행동 검출 성능 비교
Table 5. Performance comparison of online action detection on THUMOS-14[8].

Input	Method	mAP
RGB+Flow	ED[11]	43.7
	RED[11]	45.3
	TRN[12]	47.2

표 6. TVSeries[10] 데이터셋에 대한 온라인 행동 검출 성능 비교
Table 6. Performance comparison of online action detection on TVSeries[10].

Input	Method	mcAP
RGB	ED[11]	71.0
	RED[11]	71.2
	2S-FN[17]	72.4
	TRN[12]	75.4
Flow	FV-SVM[10]	74.3
RGB+Flow	ED[11]	78.5
	RED[11]	79.2
	TRN[12]	83.7

Ⅳ. 결 론

본 논문에서는 다양한 딥러닝 기반의 시간적 행동 검출 방법을 소개하였다. 최근 딥러닝의 발달은 정형 비디오를 분석하는 행동 인식의 성능을 향상 시켰을 뿐 아니라, 시간적 행동 검출과 같은 비정형 비디오에 대한 분석도 실현 가능하게 만들었다.

비정형 비디오 전체를 본 후 행동의 시작과 끝, 종류를 찾는 오프라인 행동 검출은 많은 실제 비디오를 쉽게 접할 수 있는 현대에 필요하고 주목받을 연구라고 생각한다. 또한, 비정형 스트리밍 비디오에 대해 현재의 행동을 인식하는 온라인 행동 검출 연구 역시 자율 주행 등과 같은 최신 연구 분야에 필요한 연구이다. 아직 실생활에 사용될 정도의 성능은 이루지 못하였지만 앞으로 시간적 행동 검출 연구 및 비정형 비디오 분석 연구는 활발하게 이루어질 것으로 예측된다.

References

[1] D. Tran, H. Wang, L. Torresani, and M. Feiszli, "Video classification with channel-separated convolutional networks," in *Proc. IEEE ICCV*, Oct. 2019.

[2] J.-H. Hong, B.-C. Ko, and J.-Y. Nam, "Human action recognition in still image using weighted bag-of-features and ensemble decision trees," *J. KICS*, vol. 39, no. 1, pp. 1-9, Jan. 2013.

[3] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *Proc. IEEE Conf. CVPR*, pp. 1049-1058, Jun. 2016.

[4] H. Xu, A. Das, and K. Saenko, "R-c3d: Region convolutional 3d network for temporal activity detection," in *Proc. IEEE ICCV*, pp. 5783-5792, Oct. 2017.

[5] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *Proc. IEEE ICCV*, pp. 2914-2923, Oct. 2017.

[6] J. B. Roerdink and A. Meijster, "The watershed transform: Definitions, algorithms and parallelization strategies," *Fundamenta Informaticae*, vol. 41, pp. 187-228, Apr. 2000.

[7] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, "Rethinking the faster r-cnn architecture for temporal action localization," in *Proc. IEEE Conf. CVPR*, pp. 1130-1139, Jun. 2018.

[8] Y. G. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, "*Thumos challenge: Action recognition with a large number of classes*," http://crcv.ucf.edu/THUMOS14/, 2014.

[9] F. C. Heilbron, B. G. V. Escorcia, and J. C. Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE Conf. CVPR*, pp. 961-970, Jun. 2015.

[10] R. De Geest, E. Gavves, A. Ghodrati, Z. Li, G. Snoek, and T. Tuytelaars, "Online action detection," in *Proc. ECCV*, pp. 269-285, Oct. 2016.

[11] J. Gao, Z. Yang, and R. Nevatia, "Red: Reinforced encoder-decoder networks for action anticipation," in *Proc. BMVC*, pp. 92.1-92.11, Sep. 2017.

[12] M. Xu, M. Gao, Y.-T. Chen, L. S. Davis, and D. J. Crandall, "Temporal recurrent networks for online action detection," in *Proc. IEEE ICCV*, pp. 5532-5541, Oct. 2019.

[13] S. Ren, K. He, R. Girchick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proc. Advances NeurIPS*, pp. 91-99, Dec. 2015.

[14] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proc. IEEE ICC*, pp. 4489-4497, Dec. 2015.

[15] L. Wang, Y. Xiong, Z. Wang, Y. Q. D. Lin, X. Tang, and L. van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. ECCV*, pp. 20-36, Oct. 2016.

[16] F. Perronnin, J. Sanchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proc. ECCV*, pp. 143-256, Sep. 2010.

[17] Q. Liu and Z. Wang, "Progressive boundary refinement network for temporal action detection," in *Proc. AAAI*, Feb. 2020.

[18] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, "Graph convolutional networks for temporal action localization," in *Proc. IEEE ICCV*, pp. 7094-7103, Oct. 2019.

[19] F. Long, T. Yao, Z. Qiu, X. Tian, J. Luo, and T. Mei, "Gaussian temporal awareness networks for action localization," in *Proc. IEEE Int. CVPR*, pp. 344-353, Jun. 2019.

[20] H. Liu, S. Wang, W. Wang, and J. Cheng, "Multi-scale based context-aware net for action detection," *IEEE Trans. Multimedia (TMM)*, vol. 22, no. 2, pp. 337-348, Jul. 2019.

[21] R. De Geest and T. Tuytelaars, "Modeling temporal structure with lstm for online action

detection," in *Proc. IEEE WACV*, pp. 1549-1557, Mar. 2018.

[22] S. Narayan, H. Cholakkal, F. S. Khan, and L. Shao, "3C-Net: Category count and center loss for weakly-supervised action localization," in *Proc. IEEE ICCV*, pp. 8679-8687, Oct. 2019.

[23] T. Yu, Z. Ren, Y. Li, E. Yan, N. Yu, and J. Yuan, "Temporal structure mining for weakly supervised action detection," in *Proc. IEEE ICCV*, pp. 5522-5531, Oct. 2019.

[24] D. Liu, T. Jiang, and Y. Wang, "Completeness modeling and context separation for weakly supervised temporal action localization," in *Proc. IEEE Int. Conf. CVPR*, pp. 1298-1307, Jun. 2019.

은 현 준 (Hyunjun Eun)

2013년 8월 : 경북대학교 전자공학부 졸업
2015년 8월 : 한국과학기술원 전기및전자공학부 석사
2020년 2월 : 한국과학기술원 전기및전자공학부 박사
2020년 2월~현재 : SK Telecom AI Researcher

<관심분야> 컴퓨터비전, 딥러닝, 비디오이해
[ORCID:0000-0001-7794-5377]

문 진 영 (Jinyoung Moon)

2000년 2월 : 경북대학교 컴퓨터공학부 졸업
2002년 2월 : 한국과학기술원 컴퓨터공학부 석사
2018년 8월 : 한국과학기술원 산업공학과 박사
2002년 2월~현재 : ETRI 책임연구원

<관심분야> 영상인식, 행동탐지, 비디오이해, 딥러닝
[ORCID:0000-0002-6616-824X]

박 종 열 (Jongyoul Park)

1996년 8월 : 충남대학교 컴퓨터공학과 졸업
1999년 2월 : 광주과학기술원 정보통신공학부 석사
2004년 8월 : 광주과학기술원 정보통신공학부 박사
2004년 7월~현재 : ETRI 책임연구원(실장)

<관심분야> 영상인식, 포즈인식, 행동인식, 딥러닝
[ORCID:0000-0002-4878-4129]

정 찬 호 (Chanho Jung)

2004년 2월 : 서강대학교 전자공학과 졸업
2006년 2월 : 서강대학교 전자공학과 석사
2013년 2월 : 한국과학기술원 전기및전자공학부 박사
2016년 9월~현재 : 한밭대학교 전기공학과 부교수

<관심분야> 전자공학, 영상인식, 딥러닝
[ORCID:0000-0003-3145-6732]

김 창 익 (Changick Kim)

1989년 : 연세대학교 전기공학과 학사 졸업.
1991년 : 포항공과대학교 전기전자 공학과 석사 졸업.
2000년 : 워싱턴주립대학교 전기전자 공학과 박사 졸업.
1991년~1997년 : ㈜ KC 중앙연구소 선임연구원
2000년~2005년 : Epson Palo Alto Lab. 책임연구원
2005년~2009년 : 한국정보통신대학교 조교수, 부교수
2009년~2010년 : HP Labs, Palo Alto 방문연구원
2009년~2014년 : 한국과학기술원 부교수
2015년~2016년 : UC Berkeley 방문교수
2014년~현재 : 한국과학기술원 교수

<관심분야> 영상처리, 영상이해, 컴퓨터비전, 패턴 인식
[ORCID:0000-0001-9323-8488]