일반논문 (Regular Paper)
방송공학회논문지 제27권 제4호, 2022년 7월 (JBE Vol.27, No.4, July 2022)
https://doi.org/10.5909/JBE.2022.27.4.561
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

동영상 물체 분할을 위한 효율적인 메모리 업데이트 모듈

조 준 호[a)], 조 남 익[a)‡]

Efficient Memory Update Module for Video Object Segmentation

Junho Jo[a)] and Nam Ik Cho[a)‡]

요 약

최근 대부분의 딥러닝 기반 동영상 물체 분할 방법들에서는 외부 메모리에 과거 예측 정보를 저장한 상태에서 알고리즘 수행을 하며, 일반적으로 메모리에 많은 과거 정보를 저장할수록 관심 물체의 다양한 변화에 대한 근거들이 축적되어 좋은 결과를 얻을 수 있다. 하지만 하드웨어의 제한으로 인해 메모리에 모든 정보를 저장할 수 없어 이에 따른 성능 하락이 발생한다. 본 논문에서는 저장되지 않는 정보들을 기존의 메모리에 추가적인 메모리 할당 없이 저장하는 방법을 제안한다. 구체적으로, 기존 메모리와 새로 저장할 정보들과의 어텐션 점수를 계산한 후에, 각 점수에 따라 해당 메모리에 새 정보를 더한다. 이 방법으로 물체 형체의 변화에 대한 정보가 반영되어 물체 변화에 대한 강인성이 높아져서 분할 성능이 유지됨을 확인할 수 있었다. 또한, 메모리의 누적 매칭 횟수에 따라 적응적으로 업데이트 비율을 결정하여, 업데이트가 많이 되는 샘플들은 과거의 정보를 더 기억하여 신뢰성 있는 정보를 유지할 수 있게 하였다.

Abstract

Most deep learning-based video object segmentation methods perform the segmentation with past prediction information stored in external memory. In general, the more past information is stored in the memory, the better results can be obtained by accumulating evidence for various changes in the objects of interest. However, all information cannot be stored in the memory due to hardware limitations, resulting in performance degradation. In this paper, we propose a method of storing new information in the external memory without additional memory allocation. Specifically, after calculating the attention score between the existing memory and the information to be newly stored, new information is added to the corresponding memory according to each score. In this way, the method works robustly because the attention mechanism reflects the object changes well without using additional memory. In addition, the update rate is adaptively determined according to the accumulated number of matches in the memory so that the frequently updated samples store more information to maintain reliable information.

Keyword : Video Object Segmentation, Memory, Update, Adaptive

a) 서울대학교 전기정보공학부 (Department of ECE, INMC, Seoul National University)
‡ Corresponding Author : 조남익(Nam Ik Cho)
E-mail: nicho@snu.ac.kr
Tel: +82-2-880-8480
ORCID: https://orcid.org/0000-0001-5297-4649
※ This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (2021R1A2C2007220).
· Manuscript May 20, 2022; Revised July 20, 2022; Accepted July 20, 2022.

Ⅰ. 서 론

최근 동영상 편집에 대한 수요가 늘어나면서, 편집 작업에서 가장 기본적인 동영상 물체 분할(Video Object Segmentation)에 대한 연구가 활발히 진행 중이다[1-8]. 이 중 준지도적 동영상 물체 분할(Semi-Supervised Video Object Segmentation)은 사용자가 관심 있는 물체에 대해 첫 번째 프레임(frame)에 마스크(mask)를 제공하면, 나머지 동영상 전체 프레임에 주어진 마스크를 전파하는 작업을 의미한다. 이는 동영상의 첫 번째 프레임에 대해서만 수작업을 요구한다는 점에서 매우 사용자 친화적인 작업이지만, 첫 프레임 이후 모든 프레임의 자동적인 분할을 제대로 하기에는 매우 적은 정보만을 제공하는 것이므로 프레임이 진행될수록 제대로 물체 영역을 추론하기 어려워진다. 구체적으로, 프레임이 진행되면서 관심 물체의 외형, 카메라 포즈, 배경 등에 다양한 변화들이 일어나는 어려운 경우들이 있고, 극단적으로 관심 물체가 사라지거나 다른 물체에 가려지는 상황과 같이 매우 어려운 경우들이 발생한다.

위에 언급한 문제들을 해결하기 위하여, 최근에는 동영상 물체 분할 작업에도 딥러닝(deep learning)을 도입하여 많은 성능 향상을 이룩하였다. 딥러닝 기반 물체 분할 방법들에도 다양한 접근법이 있지만, 본 논문에서는 매칭(matching) 기반의 방법들[1-3, 6-8]을 중점적으로 다루고자 한다. 매칭 기반의 방법들은 첫 번째 주어진 프레임과 추론하고자 하는 두 번째부터의 타겟 프레임 간의 픽셀(pixel) 혹은 패치(patch) 단위의 매칭을 통해 가장 비슷한 연결 관계를 찾고, 그 연결 관계에 따라 정답 마스크를 전파하는 방법이다. 이들 중 STM[1]이 가장 널리 사용되는 매칭 기반 방법이며, 외부 메모리(memory)를 이용하면서 많은 성능 향상을 이뤄냈다. STM은 과거 예측된 정보들을 외부 메모리에 저장을 한 뒤, 이를 현재 프레임의 정답을 도출할 때 활용한다. 외부 메모리 사용의 강점은 과거 프레임과 예측된 마스크에 대한 정보를 저장할 수 있다는 점이다. 즉, 관심 물체의 다양한 변화에 대한 정보들을 담을 수 있기 때문에, 입력으로 주어진 정답 프레임에 비해 급격한 변화가 있는 프레임을 추론하는 경우에도 메모리에 저장된 비슷한 형태의 정보들을 근거로 정답 마스크를 도출하는 것이다. 따라서 이 방법에서는 일반적으로 메모리의 크기와 비례한 성능이 나온다[1]. 하지만, 실제로는 하드웨어의 제약에 의해 동영상 내의 모든 프레임의 정보들을 다 저장하여 사용할 수 없으며 이에 따른 성능 저하가 일어난다.

최근 위에서 언급한 메모리 크기의 제약에 따른 성능 하락의 문제를 해결하기 위한 연구들[1,2,3]이 진행되고 있다. GCNet[2]은 한 장의 프레임만을 메모리에 저장하여 사용하는 전역 맥락 메모리(Global Contextual Memory)를 제안하였다. 모든 프레임에 대한 정보를 시간 축으로 이동 평균하며 단 1장의 메모리만을 사용하여 빠른 동영상 물체 분할 모델을 제안하였다. Adaptive Feature Bank Network (AFBNet)[3]은 기존의 이미지 형태로 저장되던 형식을 픽셀 단위의 저장으로 바꾸었다. 이를 위해 기존의 메모리와 새로 저장할 정보들과의 유사성을 계산한 후, 어텐션(attention) 점수에 따라 가산하여 각 메모리 픽셀에 저장하는 방법을 제안하였다. 본 논문에서도 위에 설명한 방법들 중[2,3]과 마찬가지로, 메모리의 제약에 의한 성능 하락을 방지하는 방법을 제안한다. 구체적으로, 메모리에 일정한 간격을 가지고 프레임들을 저장하는 상황에서, 저장되지 않고 버려지는 프레임들에 대한 정보들을 기존에 저장된 메모리에 어텐션 점수에 따라 가산하여 정보를 저장한다. 이에 따라 설계된 메모리 업데이트 모듈에서는 추가적인 메모리의 할당 없이 물체의 다양한 변화에 강인한 정보를 제공함으로써 물체 분할 성능 하락을 저지할 수 있다. AFBNet에서는 처음부터 끝까지 고정된 값을 가지고 기존의 메모리에 새로운 정보들을 업데이트(update) 하였다면, 본 논문에서 제안하는 방법에서는 업데이트 된 빈도에 따라 픽셀 단위로 적응적인 업데이트 비율을 결정하며 진행한다. 이를 통해 어느 정도 업데이트가 된 메모리는 업데이트를 더 하지 않아 과거의 신뢰성 있는 정보들을 보존할 수 있게 되어, 긴 간격을 두고 저장을 하여도 성능 하락의 폭이 줄어드는 것을 확인할 수 있다.

Ⅱ. 방법론

본 논문의 목적은 동영상에서 원하는 물체 분할을 수행하는 딥러닝 네트워크를 설계하고 구현하는 것이다. 크기

가 $H\times W$인 컬러 영상 프레임 $Q_i\in\mathbb{R}^{H\times W\times 3}$로 이루어져 있는 비디오 $V=\{Q_1, Q_2,..., Q_r\}$와 첫번째 프레임의 정답 마스크 $Y_1\in\mathbb{R}^{H\times W\times C}$가 주어졌다고 할 때, 동영상 물체 분할 네트워크 $f(\cdot,\cdot)$, 외부 메모리 $M_t\in\mathbb{R}^{N\times D}$로 구성된 네트워크가 Q_t의 정답 마스크를 도출하는 과정을 다음과 같이 표현할 수 있다.

$$\hat{Y}_t = f(Q_t, M_{t-1}). \tag{1}$$

최초의 메모리인 M_0은 동영상의 첫 번째 프레임의 마스크에 대한 정보로서 제공되는 Y_1으로 초기화가 되어있으며, 다음으로 메모리 업데이트 모듈 $U(\cdot,\cdot,\cdot)$을 통해 예측된 정보들을 메모리에 저장하는 과정은 아래와 같은 식으로 표현된다.

$$M_t = U(M_{t-1}, Q_t, \hat{Y}_t). \tag{2}$$

2.1절에서는 동영상 물체 분할 네트워크 $f(\cdot,\cdot)$ 그리고 2.2절에서는 메모리 업데이트 모듈 $U(\cdot,\cdot,\cdot)$에 대해 자세히 설명한다.

1. 동영상 물체 분할 네트워크

타겟 프레임 $Q_t\in\mathbb{R}^{H\times W\times 3}$에 대한 정답 마스크를 도출하기 위한 네트워크는 식(1) 로 표현된다. 이 때 메모리 M_{t-1}의 정보를 Q_t와의 유사도에 따라 정보들을 취합하여 정답을 도출한다. 이 유사도를 측정하기 위해 합성곱 층으로 이루어진 인코더(encoder)[9]를 통해 $k^{Q_t}\in\mathbb{R}^{n\times D_{key}}$와 $v^{Q_t}\in\mathbb{R}^{n\times D_{val}}$의 정보를 추출한다. 마찬가지로, 메모리에 저장된 M_{t-1}또한 투영을 통해 $k^{M_{t-1}}\in\mathbb{R}^{N\times D_{key}}$와 $v^{M_{t-1}}\in\mathbb{R}^{N\times D_{val}}$를 추출한다. 여기서 $n=\frac{H}{16}\times\frac{W}{16}$을 의미한다. 이 때 k는 디스크립터(descriptor)로서 픽셀들 간의 유사도를 측정하기 위해 사용되고, v는 해당 픽셀의 마스크 정보들을 지니고 있어 동영상 물체 분할을 하기 위해 직접적으로 사용되는 정보이다. 먼저 k^{Q^t}와 $k^{M_{t-1}}$과의 유사도를 기반한 연결 관계 A_{ij}는 아래와 같은 식으로 계산할 수 있다.

$$A_{ij} = \frac{exp(S(k_i^{Q_t}, k_j^{M_{t-1}})/\tau)}{\sum_{l=1}^{N} exp(S(k_i^{Q_t}, k_l^{M_{t-1}})/\tau)} \tag{3}$$

여기서 $i\in\{1,...n\}$, $j\in\{1,...,N\}$는 각 피쳐(feature)들의 픽셀 인덱스(index)를 의미하며, $S(\cdot,\cdot)$은 두 인자 간의 유사도를 측정하는 식으로서 내적, 코사인 유사도 등을 사용할 수 있다[1]. 본 논문에서는 메모리 업데이트 모듈에서 업데이트할 때 피쳐의 크기와 무관한 매칭을 위하여 코사인 유사도를 사용하였다. 그리고, τ는 유사도의 강도를 조절하는 파라미터이다. 디코더(decoder)로 넘겨주는 피쳐 $V\in\mathbb{R}^{n\times 2\times D_{val}}$를 구하기 위해 식(3)을 통해 얻어진 어텐션 점수에 따라 메모리에 저장되어 있는 $v^{M_{t-1}}$의 정보의 가중치 합을 구한 다음 현재 프레임의 v^{Q_t}과 다음과 같이 병합한다.

$$V_{i,1} = \left[v_i^{Q_t}; \textstyle\sum_{l=1}^{N} A_{i,l} \bullet v_l^{M_{t-1}}\right]. \tag{4}$$

이와 같이 병합된 피쳐들을 합성곱층을 지나면서 물체를 분할하는 디코더[10]에 넘겨주면, 네트워크는 과거의 정보들을 근거로 하여 현재 프레임에 대한 정답 마스크 $\hat{Y}_t$ 추론을 한다. 전체적인 네트워크 구조는 STM[1]을 기본으로 실험하였다.

2. 메모리 업데이트 모듈

앞 절에서 설명한 방식으로 현재 프레임에 대한 정답 마스크 $\hat{Y}_t$를 도출한 뒤, 예측한 정보를 바탕으로 새로운 정보들을 메모리에 저장하는 업데이트 과정(식-(2))을 진행한다. 앞 절에서와 마찬가지로 어텐션에 따른 가산합을 진행하기 위하여 새로운 프레임 Q_t와 예측한 마스크 $\hat{Y}_t$를 인코더에 넣어 $k^{new}\in\mathbb{R}^{N\times D_{key}}$와 $v^{new}\in\mathbb{R}^{n\times D_{val}}$를 추출한다. 이를 이용하여 아래와 같은 식으로 기존의 메모리와 새로 저장할 메모리 정보 간의 어텐션 점수를 계산한다.

$$B_{ij}^{t} = \frac{exp(S(k_j^{M_{t-1}}, k_i^{new},)/\tau)}{\sum_{l=1}^{N} exp(S(k_i^{M_{t-1}}, k_l^{new})/\tau)}. \tag{5}$$

위의 점수를 근거로 새로 저장할 정보들을 기존의 메모리에 아래와 같은 식으로 합산하여 기존의 메모리를 업데이트한다.

$$k^{M_t} = \lambda(t) \cdot k^{M_{t-1}} + (1-\lambda(t)) \cdot B * k^{new}, \quad (6)$$

$$\lambda(t) = \frac{u(t-1)}{u(t)}, \quad (7)$$

$$u(t) = u(t-1) + \Sigma_{j=0}^{n} \mathrm{B}_{ij}^{t} \quad (8)$$

여기서 $u(t) \in \mathbb{R}^N$는 기존의 메모리 픽셀들의 새로 들어온 정보들과 매칭된 총 누적 양을 의미한다. 이는 업데이트 할수록 높은 값을 갖게 되어 $\lambda(t) \in \mathbb{R}^N$ 값이 점점 1로 수렴한다. 따라서 빈번히 업데이트되면 될수록 적응적으로 기존의 메모리에 대한 비중이 높아져 신뢰성 있는 과거 정보를 지켜간다고 생각할 수 있다. 최종적으로 메모리에 저장되는 정보는 피쳐 공간상에서 매칭된 모든 픽셀의 평균값이 된다. 위와 같은 메모리 업데이트 모듈을 통해 업데이트한 메모리를 사용하여 다음 프레임의 정답을 도출하는 식으로 동작한다.

위에서 언급한 식-(5), (6)처럼 기존의 메모리와 새로 저장할 정보 간의 어텐션 점수에 따라 업데이트를 진행하였을 때, 높은 상관관계가 있는 정보만을 가지고 업데이트하기 위하여 새로운 정보와의 어텐션 점수에 top-K 매칭 알고리즘[7]을 적용하여, 상관도가 높은 정보 K개만 이용하여 진행하였다. 전체적인 메모리 업데이트 구조도는 그림 1에서 확인 할 수 있다.

Ⅲ. 실험 결과 및 분석

제안하는 동영상 물체 분할의 메모리 업데이트 모듈의 성능을 측정하기 위하여, 공개되어 있는 데이터 셋인 DAVIS 2017[4], YouTube-VOS 2018[5]을 사용하였다. 성능 측정으로는 분할 영역의 IoU 성능 (Jaccard: J) 과 분할 영역의 경계선의 IoU 성능 (F-measure: F)의 평균값($J\&F$)을 사용하였다.

1. 메모리 사용량에 따른 성능 비교

그림 2는 제안한 모듈의 업데이트 부분을 다른 참조 논문들[1,2,3]에서 사용한 방법들과 비교한 결과이다. 우선 메모리 저장 간격에 따른 성능을 보면, 전체적으로 간격이 늘어날수록 $J\&F$ 성능이 떨어지는 것을 볼 수 있다. 특히 기본 모델인 STM[1]의 경우 4번째 프레임마다 저장하는 것에서 128 프레임마다 저장하는 상황으로 메모리 사용량을 낮추면 성능 하락이 2.52가 발생하는 것을 볼 수 있었다. 이는

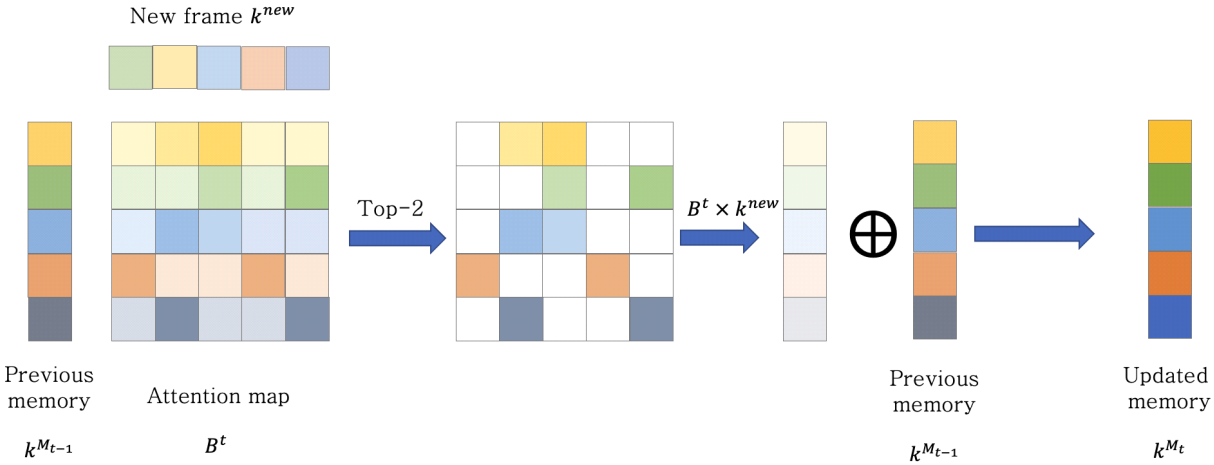

그림 1. 메모리 업데이트 모듈 구조도. 그림 내의 ⊕는 식(6)을 의미한다
Fig. 1. Diagram of the memory update module. In the figure, ⊕ denotes Equation-(6)

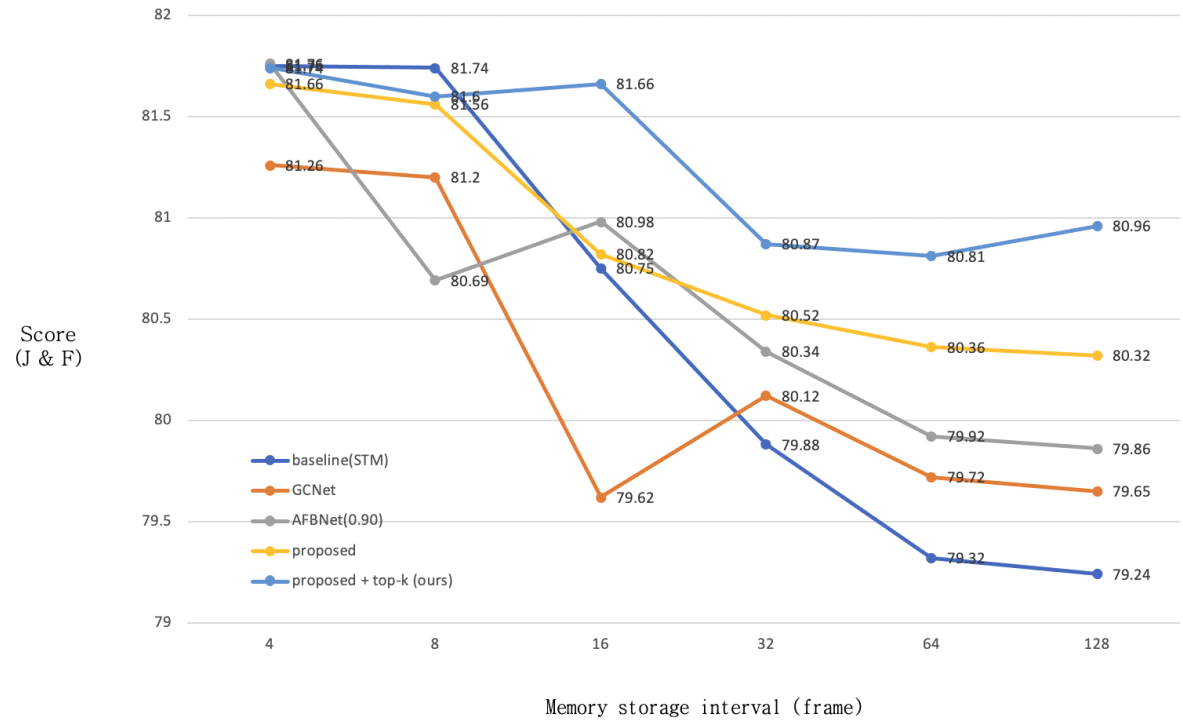

그림 2. 메모리 저장 간격에 따른 DAVIS 2017 validation 성능 그래프
Fig. 2. Performance graph according to memory storage interval for DAVIS 2017 validation set

메모리에 저장되는 양이 작아짐에 따라 정답을 도출할 때 사용되는 근거들이 적어져 발생하는 성능 하락으로 볼 수 있다. 저장 간격이 128 프레임인 경우 DAVIS 2017 validation set의 모든 비디오에서 첫 번째 프레임만 저장을 하는 상황이다. 정보를 저장하는 방법의 공정한 비교를 하기 위하여, 메모리와 매칭된 정보를 가산할 때의 단계 이외에는 모두 같은 상황으로 실험을 진행하였다. GCNet[2]의 경우는 정보를 합칠 때 시간 축으로 이동 평균을 진행하는 방법이며, AFBNet[3]은 고정된 업데이트 비율을 가지고 가산을 하는 방식이다. 이때 다른 참조 논문들에 비해 제안하는 방법이 저장 간격에 따른 성능 하락 폭이 가장 작은 것을 확인할 수 있으며, 이는 같은 메모리 크기일 때, 정보 저장을 효율적으로 하여 발생한 성능 향상이라고 볼 수 있다. 또한 추가적으로 top-k 알고리즘을 적용하여, 상관도가 낮은 정보들에 대한 배제가 성능향상을 이끈 것을 확인 할 수 있었다.

2. YouTube-VOS 성능 비교

표 1은 YouTube-VOS 2018 validation set으로 메모리의 제한 없이 측정한 성능과 단 2장만을 사용한 세팅에서의 실험 결과를 기록한 것이다. 학습 데이터에 존재하는 클래스에 대한 결과는 seen, 존재하지 않는 경우의 결과는 unseen으로 분류되어 표기하며, Overall은 J와 F 성능의 평균

표 1. YouTube-VOS 2018 validation set 성능 비교
Table 1. Performance comparison for YouTube-VOS 2018 validation set

Method	Memory constraint	Overall	J		F	
			seen	unseen	Seen	unseen
STM[6]	X	79.4	79.7	72.8	84.2	80.9
AFB-Net[4]	X	79.6	78.8	74.1	83.1	82.6
STM-FL	O	77.2	77.4	71.0	81.5	78.7
GCNET[5]	O	73.2	72.6	68.9	75.6	75.7
AFB-Net	O	77.7	77.2	72.1	81.4	80.2
ours	O	78.5	78.8	72.2	83.2	79.6

치를 의미한다. 기본 모델인 STM[1]의 경우 메모리의 제한 없이 5장마다 저장하는 식으로 했을 경우 성능이 79.4인 반면, 메모리 제한이 들어가 2장만 사용한 경우(STM-FL)에는 77.2로 성능 하락이 있는 것을 볼 수 있었다. 이때 제안하는 방법을 추가한 경우(STM + MUM) 78.5로 성능이 오른 것을 확인할 수 있다. GCNet[2]의 경우에는 많은 하락이 있는 것을 확인 할 수 있는데, 이는 메모리와 새로운 정보 간의 매칭 없이 시간 축으로의 이동 평균만을 진행하기에, 의미 없는 정보들이 되어버려 성능이 하락한 것으로 볼 수 있다. 제안하는 방법의 경우에는 메모리 사용량은 똑같이 유지하지만, 성능이 오른 것을 보아 제안하는 메모리 업데이트 방식에 의한 저장이 효과적이라고 생각할 수 있다.

3. DAVIS 2017 validation 시각화 성능 비교

그림 3과 그림 4는 DAVIS 2017 validation set 중 “bmx-trees,” “India”라는 비디오에 대한 결과를 시각화한 것이다. 앞 프레임까지의 메모리 업데이트를 하는 네트워크들 (AFB[3],

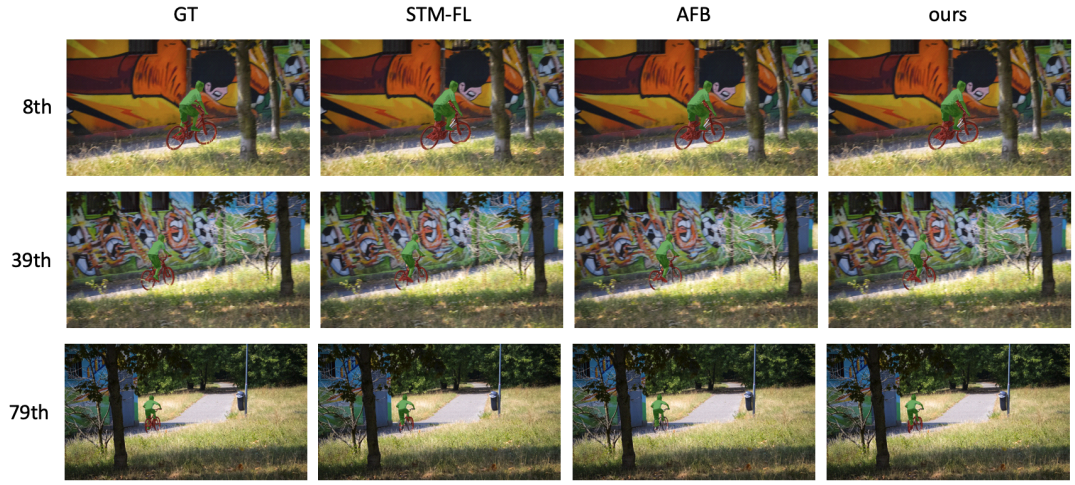

그림 3. DAVIS 2017 validation set 중 “bmx-trees” 결과
Fig. 3. Results for “bmx-trees” video of DAVIS 2017 validation set

그림 4. DAVIS 2017 validation set 중 “India” 결과
Fig. 4. Results for “India” video of DAVIS 2017 validation set

ours)의 성능은 베이스라인인 첫 번째와 마지막 프레임에 대한 정보를 사용하는 STM-FL과의 차이는 거의 나지 않는 것을 확인할 수 있다. 하지만 비디오의 마지막에 있는 프레임들에 대한 결과를 보면, 업데이트하는 네트워크에 비해 성능이 좋게 나오지 않는 것을 확인할 수 있다. 이는 STM-FL 같은 경우 첫 번째 주어지는 정보만을 업데이트하지 않고 사용하기 때문에, 많은 변화가 일어나는 뒤쪽 프레임일수록 성능 저하가 많이 일어나는 것이라고 생각할 수 있다. 반대로, 본 논문에서 제안하는 방법의 경우, 처음 주어진 정보를 다음 프레임을 사용하여 업데이트하기 때문에, 변화에 강인한 결과를 보여준다.

Ⅳ. 결 론

본 논문에서는 메모리 제약에 따른 동영상 물체 분할 네트워크의 성능 하락을 방지하는 메모리 업데이트 모듈을 제안하였다. 제안 방법에서는 기존의 메모리 정보와 새로 들어온 정보를 매칭한 후 업데이트된 정도에 따라 이들을 적응적으로 가산하여 메모리를 업데이트한다. 메모리 업데이트 가산 방법들과 DAIVS 2017 validation set의 성능 비교 측정에서, 기존의 STM 모델의 경우 전체 메모리 사용량을 1장으로 줄였을 때, 동영상 물체 분할 성능인 $J\&F$ 측정치가 2.52 만큼 떨어지는 반면 제안하는 모듈을 통해 메모리를 업데이트 한 결과 0.79 만의 성능 하락을 보였다. 이를 통해 제안한 방법이 메모리의 사용량 증가 없이 바뀌는 정보들을 잘 저장하여 우수한 결과를 가져옴을 알 수 있다.

참 고 문 헌 (References)

[1] Oh, S. W., Lee, J. Y., Xu, N., & Kim, S. J., "Video object segmentation using space-time memory networks." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.
doi: https://doi.org/10.1109/ICCV.2019.00932

[2] Li, Yu, Zhuoran Shen, and Ying Shan., "Fast video object segmentation using the global context module." European Conference on Computer Vision. Springer, Cham, 2020.
doi: https://doi.org/10.1007/978-3-030-58607-2_43

[3] Liang, Y., Li, X., Jafari, N., & Chen, J., "Video object segmentation with adaptive feature bank and uncertain-region refinement." Advances in Neural Information Processing Systems 33: 3430-3441., 2020.

[4] Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., & Van Gool, L, "The 2017 davis challenge on video object segmentation." arXiv preprint arXiv:1704.00675, 2017.

[5] Ning Xu, Linjie Yang, Dingcheng Yue, Jianchao Yang, Brian Price, Jimei Yang, Scott Cohen, Yuchen Fan, Yuchen Liang, and Thomas Huang., "Youtube-vos: Sequence-to-sequence video object segmentation." In European Conference on Computer Vision (ECCV), 2018.
doi: https://doi.org/10.1007/978-3-030-01228-1_36

[6] Yao, R., Lin, G., Xia, S., Zhao, J., & Zhou, Y., "Video object segmentation and tracking: A survey." ACM Transactions on Intelligent Systems and Technology (TIST) 11.4 ,1-47p, 2020.
doi: http://dx.doi.org/10.1145/3391743

[7] Wang, H., Jiang, X., Ren, H., Hu, Y., & Bai, S., "Swiftnet: Real-time video object segmentation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.
doi: https://doi.org/10.1109/CVPR46437.2021.00135

[8] Hu, Yuan-Ting, Jia-Bin Huang, and Alexander G. Schwing. "Videomatch: Matching based video object segmentation." Proceedings of the European conference on computer vision (ECCV). 2018.
doi: https://doi.org/10.1007/978-3-030-01237-3_4

[9] He, K., Zhang, X., Ren, S., & Sun, J., "Deep residual learning for image recognition.", Proceedings of the IEEE conference on computer vision and pattern recognition, p. 770-778, 2016.
doi: https://doi.org/10.1109/CVPR.2016.90

[10] T. -Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection.", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 936-944, 2017.
doi: https://doi.org/10.1109/CVPR.2017.106

저 자 소 개

조 준 호

- 서울대학교 전기정보공학부 석박통합과정
- ORCID : https://orcid.org/0000-0002-3546-1574
- 주관심분야 : 동영상 물체 분할, 트랜스포머, 딥러닝

조 남 익

- 서울대학교 전기정보공학부 교수
- ORCID : https://orcid.org/0000-0001-5297-4649
- 주관심분야 : 디지털 신호처리, 영상처리, 컴퓨터 비전