비디오 객체 분할(video instance segmentation) 기술 동향

2022. 01. 24.

시각지능연구실
지능정보연구본부
인공지능연구소

ETRI 한국전자통신연구원
Electronics and Telecommunications Research Institute

목 차

1 개요

1.1 목적

본 문서는 컴퓨터 비전 분야에서 최근에 제안된 비디오 객체 분할(video instance segmentation) 기술에 대한 최신 연구 동향과 각 연구에서 제시하는 비디오 객체 분할 모델에 대한 설명을 기술한다.

1.2 범위

본 문서는 비디오 객체 분할 문제에 대한 정의와 최신 연구에서 제시한 비디오 객체 분할 모델에 대한 설명을 포함한다.

2 비디오 객체 분할 문제 정의

Hariharan *et al.* [1]이 처음 제시한 이미지에서의 객체 분할(instance segmentation)은 개별 객체의 탐지(detection)과 분할(segmentation)을 동시에 수행하는 것을 목표로 한다. 객체 분할 기술이 등장한 이후 그 중요성을 인정받아 컴퓨터 비전 분야에서 많은 관심을 받고 있다. 최근에는 객체 분할 문제를 이미지에서 비디오로 가져온 비디오 객체 분할(video instance segmentation) 문제가 제안되었다 [2][1]. 비디오 객체 분할은 비디오에서의 개별 객체의 탐지, 분할, 그리고 개별 객체의 추적(tracking)까지 동시에 수행하는 새로운 문제이다. 비디오 객체 분할 기술은 향후 비디오 편집, 자율 주행, 증강 현실 등 비디오에서의 객체 마스크(mask)가 필요한 영역에서 활용될 수 있을 것으로 기대되고 있다.

그림 1은 비디오 객체 분할의 예시를 보여준다. 그림 1의 첫 번째 행은 비디오 원본 영상의 예시이다. 두 번째 행은 비디오 원본 영상에서 나타난 각 객체들의 탐지, 분할, 추적 결과를 나타낸다. 비디오 안에서 각 객체는 여러 프레임(frame)에 걸쳐서 나올 수 있는데, 비디오 객체 분할 문제는 각 프레임에 등장한 객체가 하나의 동일한 객체였음을 인식할 수 있어야 한다. 두 번째 행의 첫 번째 프레임 결과를 보면 각기 다른 4명의 사람이 등장한다는 것을 알 수 있다. 그리고 두 번째, 세 번째 프레임에서는 사람들이 조금씩 움직이게 되는데, 객체 분할 결과를 보게 되면 사람들의 위치 변화를 파악하여 같은 객체을 지속적으로 추적하고 있음을 알 수 있다.

2.1 기존의 다른 문제와의 차이점

이미지의 객체 분할과 비교할 때, 비디오 객체 분할은 단순히 개별 프레임의 객체 분할을 수행하는 것뿐만 아니라 각 프레임에 등장한 객체를 하나의 동일한 객체였음을 이어주는 객체 추적이 필요하다는 점에서 더 어려운 문제라 할 수 있다. 대신, 비디오는 서로 다른 객체의 움직임 패턴, 시간적 일관성 등 단일 이미지에서 포함한 것보다 더 많은 정보들을 포함하기 때문에 객체 인식, 분할 시에 더 많은 단서를 제공할 수 있다.

기존의 다른 문제 중 비디오 객체 분할과 유사한 문제로는 비디오 물체 분할(video object segmentation)

[1]비디오 객체 분할 문제는 ICCV 2019에서 제안되었으나, 그보다 조금 앞서 다중 객체 추적 및 분할(multi-object tracking and segmentation) 문제가 CVPR 2019에서 제안되었다 [3]. 두 문제는 접근하는 관점의 차이가 있으나, 사실 같은 문제이다.

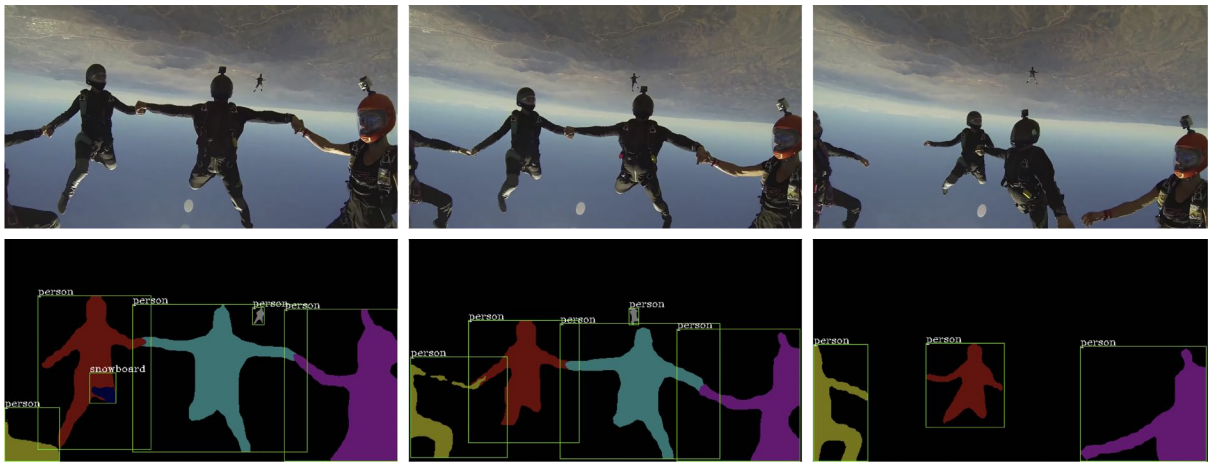

그림 1: 비디오 객체 분할 예시 그림 [2]. 위에서부터 첫 번째 행은 비디오의 이미지 프레임을, 두 번째 행은 MaskTrack R-CNN의 비디오 객체 분할 결과를 보여준다.

문제가 있다. 비디오 물체 분할 [4] 문제는 비디오에서 대상이 되는 물체를 분할, 추적하는 것을 목표로 하고 있다. 보통 첫 번째 프레임에서 사람이 직접 대상이 되는 물체를 선정하여 모델의 입력으로 전달하면, 모델은 대상 물체의 마스크를 학습하여 그 다음 프레임부터 동일한 물체을 추적하고 마스크를 분할한다. 이때, 대상이 되는 물체의 종류는 비디오 물체 분할 문제에서 요구되지 않는다. 반면, 비디오 객체 분할 문제는 인식할 수 있는 물체의 종류가 한정되나, 비디오에 등장하는 등장하는 모든 객체들을 인식, 탐지, 분할, 추적할 수 있어야 한다. 특히, 객체가 비디오 중간에 등장하더라도 새로운 객체로 인식하여 결과를 출력할 수 있어야 한다.

3 MaskTrack R-CNN

MaskTrack R-CNN [2]은 비디오 객체 분할 문제를 해결한 첫 번째 기술로, 대표적인 이미지 객체 분할 모델 중 하나인 Mask R-CNN [5]을 확장하여 만들어졌다. MaskTrack R-CNN은 Mask R-CNN의 객체 분류(object classification), 경계 상자 회귀(bounding box regression), 마스크(mask) 생성을 위한 기존의 세 가지 브랜치(branch)에, 비디오 전체에서 개별 객체를 추적하기 위한 추적 브랜치와 관측된 객체 정보를 저장하는 외부 메모리(external memory)를 추가함으로써 비디오 객체 분할 문제를 해결한다. 본 장에서는 선행 기술의 문제점, MaskTrack R-CNN의 전체적인 구조과 주요 요소, 선행 기술과 비교했을 때의 장점을 알아본다.

3.1 선행 기술의 문제점

MaskTrack R-CNN 이전에는 비디오 객체 분할 문제를 해결하는 선행 기술이 존재하지 않는다. 하지만 유사한 다른 분야의 선행 기술들을 조합하여 비디오 객체 분할 문제에 적용하는 방안을 생각해볼 수 있다. 첫 번째는 비디오의 첫 번째 프레임에서 나타난 객체의 마스크를 구한 다음, 이를 비디오 물체 분할 알고리듬의 입력으로 넣어 초기 마스크를 전파시키는(mask propagation) 방법을 생각할 수 있다. 하지만

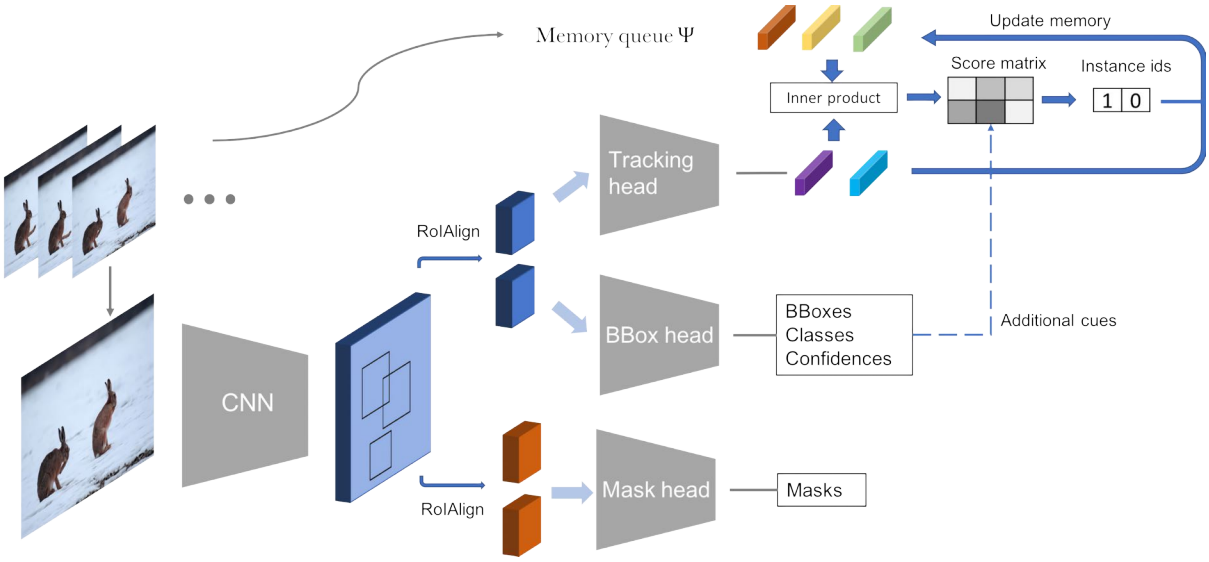

그림 2: MaskTrack R-CNN 구조도

이 방법은 비디오 중간에 등장하는 객체를 처리할 수 없고, 첫 번째 프레임의 마스크 품질 저하가 전체 비디오의 객체 마스크 품질과 직결된다는 문제점을 안고 있다. 두 번째는 tracking-by-detection 방법을 차용하는 것으로, 각 프레임에서 나타나는 객체 탐지 결과를 추적 알고리듬으로 연결하는 것이다. 이 방법의 경우, 객체 탐지와 추적을 독립적으로 수행하기 때문에 두 하위 작업(sub-task)이 정보를 공유하지 않아 전체적인 성능이 저하된다는 단점이 있다. 또, 어떤 추적 방법을 선택하는 지에 따른 한계점(시각 정보(visual information)를 사용하지 않거나, 오프라인으로만 동작하여 전체 비디오를 한번에 처리해야만 하는 등)이 그대로 나타날 수 있다.

3.2 MaskTrack R-CNN의 구조

그림 2는 MaskTrack R-CNN의 구조도이다. MaskTrack R-CNN은 이미지 객체 분할 모델 중 2단계 절차(two-stage) 모델인 Mask R-CNN을 확장하여 만들어졌다. MaskTrack R-CNN의 두 단계는 다음과 같이 구성된다. 첫 번째 단계에서는 region proposal network (RPN) [6]가 각 프레임에서 등장하는 객체의 후보 경계 상자들을 찾아낸다. 두 번째 단계에서는 각 후보 경계 상자에 대해 RoIAlign 연산으로 특징 벡터를 추출하고, 이 특징 벡터들은 Mask R-CNN의 세 개의 브랜치와 병렬로 연결된 새로운 추적 브랜치에서 사용되어 각각 분류, 경계 상자 회귀, 마스크 생성, 객체 추적을 수행한다. 이때, 추적 브랜치에는 이미 관측된 객체 정보를 저장하는 외부 메모리가 추가되어 객체 추적을 용이하게 한다.

3.2.1 추적 브랜치

MaskTrack R-CNN에서 제안하는 추적 브랜치는 주로 외관 유사성(appearance similarity)을 활용하여, 첫 번째 단계에서 얻은 후보 경계 상자에 레이블(label)을 할당할 확률을 계산한다. 이전에 관측된 객체가 N개 있다고 가정하면, 새 후보 상자는 이전 객체 N개 중 하나에 해당하거나, 새롭게 등장한 객체일 것이다.

새로운 객체인 경우 레이블 0을 할당한다고 하면, 후보 상자 i에 레이블 n을 할당하는 확률은 다음과 같이 정의한다.

$$p_i(n) = \begin{cases} \dfrac{\exp(f_i^T g_n)}{1 + \sum_{j=1}^{N} \exp(f_i^T g_j)}, & n = 1, \cdots, N, \\ \dfrac{1}{1 + \sum_{j=1}^{N} \exp(f_i^T g_j)}, & n = 0. \end{cases} \tag{1}$$

여기서 f_i과 g_j $(j = 1, \cdots, N)$는 각각 후보 상자 i와 N개의 관측된 객체로부터 추적 브랜치의 두 개의 fully connected layer가 RoIAlign에서 추출된 특징맵을 투영하여(projection) 만들어낸 특징 벡터를 의미한다[2]. 추적 브랜치를 학습할 때 추적 손실 $\mathcal{L}_{track}$은 cross entropy loss를 사용한다. 기존의 브랜치 손실(loss)까지 결합하여, 전체 네트워크의 손실은 $\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{box} + \mathcal{L}_{mask} + \mathcal{L}_{track}$으로 정의하며, 이를 통해 전체 네트워크를 end-to-end로 학습하게 된다.

위의 계산에서 연산의 효율성을 위해 이전에 관측된 객체의 특징 벡터는 미리 계산되어 외부 메모리에 저장된다. 또, 후보 상자에 레이블이 할당되면 외부 메모리를 동적으로 업데이트한다. 만약 후보 상자가 이전에 존재하던 N개의 객체 중 하나의 레이블을 할당받는다면 외부 메모리에 저장된 해당 객체의 특징을 새로운 후보의 특징으로 업데이트하여 해당 객체의 가장 최신 상태(state)를 유지한다. 만약 후보 상자가 새로운 객체에 해당한다면, 즉 레이블 0을 할당받는다면, 외부 메모리에 후보 객체 특징을 추가하고 관측된 객체 수에 1을 더한다.

3.2.2 추적 성능 향상 방법

추적 브랜치에서 후보 상자의 레이블을 할당할 때 외관 유사성 이외에 의미 일관성(semantic consistency), 공간 상관관계(spatial correlation), 탐지 신뢰성(detection confidence) 등 추가적인 정보를 결합한다면 추적 정확도를 높일 수 있다. 구체적으로 후보 상자 i에 대해 네트워크의 경계 상자 브랜치와 분류 브랜치로부터 얻을 수 있는 경계 상자 예측값, 분류 레이블, 탐지 점수를 각각 b_i, c_i, s_i라 정의한다. 마찬가지로 레이블 n에 해당하는 객체의 경우 외부 메모리에 저장된 특징 벡터에 대한 경계 상자 예측값과 범주 레이블을 각각 b_n, c_n으로 정의한다. 그러면 후보 상자 i에 레이블 n을 할당하기 위한 점수는 다음과 같이 정의한다.

$$v_i(n) = \log p_i(n) + \alpha \log s_i + \beta \, \mathrm{IoU}(b_i, b_n) + \gamma 1_{\{c_i = c_n\}}. \tag{2}$$

여기서 $p_i(n)$은 수식 (1)과 같이 계산되며, $\mathrm{IoU}(b_i, b_n)$은 두 경계 상자 b_i, b_n 간의 IoU를 의미한다. 이처럼 각 레이블 n에 대해 계산한 점수 $v_i(n)$을 기반으로, 후보 상자 i에게 가장 높은 점수에 해당하는 레이블을 할당한다. 참고로 수식 (2)은 네트워크 추론(inference) 단계에서만 사용되며, 학습 시에는 사용되지 않는다.

[2]MaskTrack R-CNN 논문 [2]에서는 g_j 대신 f_j로 표기하고 있으나, f_i가 현재 프레임에서 새로 관측한 객체의 특징 벡터이고 g_j는 이전 프레임에서 이미 관측된 객체들의 특징 벡터이므로 f_i, f_j처럼 동일한 f로 표기하면 오해의 소지가 발생할 수 있다. 따라서 여기서는 다른 표기를 사용한다.

3.2.3 추론 방법

새로운 비디오를 대상으로 객체 분할을 수행한다고 하면, 먼저 외부 메모리는 아무 것도 저장되어 있지 않으며, 관측된 객체의 수를 0으로 놓는다. 추론 방법은 온라인(online) 방식으로, 매 프레임 입력으로 들어올 때마다 객체 분할을 수행한다. 각 프레임별로 객체 분할을 수행한 후, 나머지 결과를 대상으로 수식 (2)을 이용하여 이미 관측된 객체의 레이블을 할당하여 추적을 수행한다. 단, 비디오의 첫 번째 프레임에서는 객체 분할 결과를 전부 새로운 객체로 인식하고 외부 메모리에 저장한다.

모든 프레임에 대해 객체 분할을 수행하고 나면 결과물로 비디오에서 관측된 각 객체에게 할당된 레이블과, 프레임별 객체 클래스 정보, 마스크, 탐지 신뢰도(confidence)의 시퀀스(sequence)를 얻게 된다. 여기서는 전체 탐지 신뢰도의 평균을 전체 시퀀스의 신뢰도로 사용하고, 객체 클래스 결과의 다수(majority)에 해당하는 클래스를 해당 객체의 최종 클래스로 사용한다.

3.3 선행 기술 대비 장점

MaskTrack R-CNN을 앞서 언급한 두 가지 선행 기술 활용 방안과 비교하면 다음과 같은 장점을 갖고 있다. 마스크 전파 방법과 비교할 때, 선행 기술로는 비디오 중간에 등장하는 객체를 처리할 수 없었지만, MaskTrack R-CNN은 이를 새로운 객체로 인식하고 외부 메모리에 저장하여 처리할 수 있다. 또한 tracking-by-detection 방법과 달리, MaskTrack R-CNN은 추적 브랜치도 다른 브랜치들과 마찬가지로 end-to-end로 학습이 가능하여 추적 브랜치의 손실이 다른 브랜치로도 역전파(back propagation)되기 때문에 전체적인 성능 향상을 이끌어낸다. 이외에도 MaskTrack R-CNN은 여러 프레임에 걸쳐 이미 관측된 객체가 사라졌다가 다시 등장해도 여전히 추적 가능하다.

종합하면, MaskTrack R-CNN은 비디오 객체 분할 분야의 포문을 연 첫 번째 모델이자, 앞선 두 가지 유형의 선행 기술 활용 방법들보다 높은 성능을 보이면서 향후 연구자들에게 비디오 이해 분야의 새로운 아이디어와 연구 방향을 떠올리게 만든 기술이라 할 수 있다.

4 SG-Net

MaskTrack R-CNN 이후 비디오 객체 분할 문제를 해결하는 많은 방법들은 대부분 2단계 절차 모델을 기반으로 하고 있다. SG-Net [7]은 이러한 관점을 바꿔 세개의 하위 작업(sub-task)인 탐지, 분할, 추적을 상호 연결된 문제로 간주하는 1단계 절차(one-stage) 모델 기반의 공간 세분화 네트워크(spatial granularity network)를 제시한다. 본 장에서는 선행 기술의 문제점, SG-Net의 전체적인 구조와 주요 요소, 선행 기술과 비교했을 때의 장점을 알아본다.

4.1 선행 기술의 문제점

비디오 객체 분할 문제에서 기존의 Mask R-CNN 기반의 2단계 절차를 따르는 모델인 MaskTrack R-CNN은 RPN에서 각 프레임에 등장하는 객체의 후보 경계 상자를 선정한 다음, 각각의 후보 경계 상자에 대해 RoI 특징 정보를 추출하여 각 브랜치의 입력으로 넣어 객체 분류, 경계 상자 계산, 마스크 생성, 객체 추적을 수행한다. 이러한 2단계 절차 모델은 몇 가지 문제점이 존재한다. 첫 번째는 각 하위 작업 브랜치

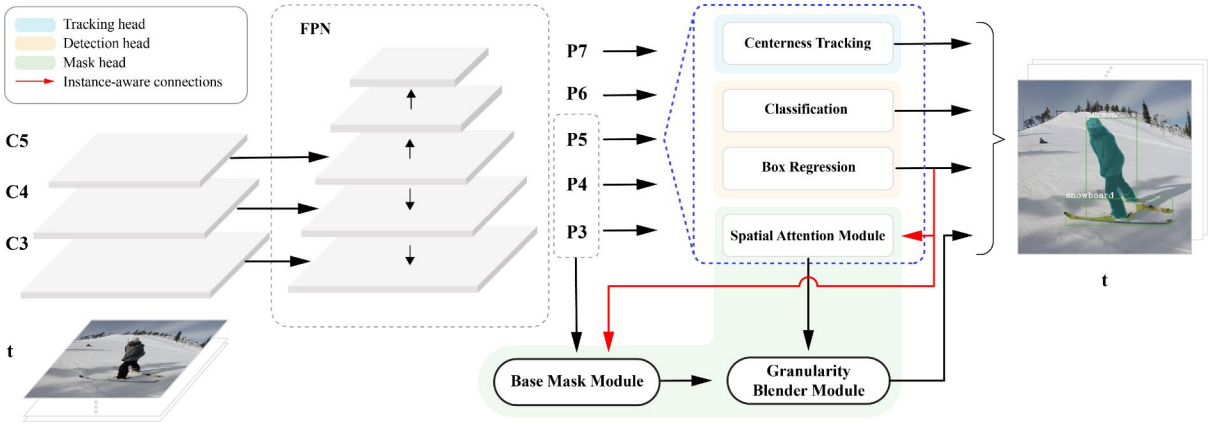

그림 3: SG-Net 구조도

간 특징 정보의 공유가 어려워 모델의 전체적인 최적화에 어려움이 발생한다. 두 번째는 RoI 특징 정보가 균일한 크기로 정규화되어 객체 마스크의 출력 해상도가 제한된다. 세 번째는 객체와 배경(background)를 구별해내기 위해 이미지의 충분한 맥락(context) 정보를 이해하는 과정에서 비교적 큰 receptive field를 사용하게 되어 추론 시에 관측된 객체의 수에 따라 계산 시간이 크게 바뀌게 된다.

최근 이미지 객체 분할 문제에서, BlendMask [4]와 CondInst [8]처럼 기존의 Mask R-CNN 기반의 2단계 절차 모델 대신 FCOS [9]처럼 1단계 절차(one-stage) 모델을 기반으로 한 접근 방법이 연구되었다. 이 방법들은 RoI 연산을 사용하지 않고 오직 fully convolutional network (FCN) 구조를 갖고 있어 특징맵의 해상도와 마스크 경계의 세부사항을 보존하여 성능을 크게 개선하였다. 하지만 이런 방법들은 객체 수준에서 마스크를 계산하기 때문에 세밀한 마스크를 구하기 어렵다는 문제점이 있다.

4.2 SG-Net의 구조

그림 3은 SG-Net의 구조도이다. SG-Net은 이미지 객체 탐지 모델 중 1단계 절차 모델인 FCOS를 기반으로 하여 만들어졌다. SG-Net은 다음과 같이 구성된다. FCOS와 마찬가지로 ResNet [10] 백본 네트워크와 feature pyramid network (FPN) [11]를 갖고 있으며, FPN의 P3, P4, P5, P6, P7에서 추출한 특징맵을 사용하여 각각의 하위 작업을 수행한다. 객체 탐지 헤드(head)는 FCOS와 동일하게 객체 분류, 경계 상자 회귀, 중심성(centerness) 브랜치를 포함한다. SG-Net은 추가적으로 비디오 객체 분할을 위해 마스크 헤드와 추적 헤드를 사용한다.

4.2.1 마스크 헤드

SG-Net의 마스크 헤드는 그림 3의 초록색 부분과 같이 공간 주의 모듈(spatial attention module), 기본 마스크 모듈(base mask module), 세분화 혼합 모듈(granularity blender module)로 이루어져 있다. 각각의 모듈은 최종 마스크를 계산하기 위해 유기적으로 연결되어 작동한다.

공간 주의 모듈은 객체 탐지 결과로부터 시작하여, 개별 객체의 경계 상자를 여러 개의 하위 영역으로 나누고 각 하위 영역의 attention score를 계산한다. 객체의 경계 상자의 크기가 $w \times h$로 주어졌을 때,

해당 객체를 $r_1 \times r_2$의 하위 영역으로 나누게 된다. 여기서 r_1과 r_2는 다음과 같이 계산한다.

$$r_1 = \min\left(6, \left\lceil \frac{w}{50} \right\rceil\right), \quad r_2 = \min\left(6, \left\lceil \frac{h}{50} \right\rceil\right). \tag{3}$$

이후 FPN의 P3-P7에 두 개의 3×3 컨볼류션 레이어를 추가하여 모든 하위 영역에 대한 attention score를 계산하고, 이를 a^j $(j = 1, \cdots, r_1 \times r_2)$로 표시한다.

기본 마스크 모듈은 FPN의 P3-P5에서 특징맵을 이용하여 각 객체의 기본 마스크를 계산한다. 각 객체의 경계 상자 바깥쪽의 특징 값을 0으로 하여 객체별 특징맵을 구한 후, 1×1 컨볼루션을 적용하여 기본 마스크를 동적으로 생성한다. 각 객체마다 생성하는 기본 마스크의 수는 공간 주의 모듈의 하위 영역의 개수로 정해진다. 따라서 각 객체마다 기본 마스크 $B^j \in \mathbb{R}^{(H/2)\times(W/2)}$ $(j = 1, \cdots, r_1 \times r_2)$가 만들어진다[3].

세분화 혼합 모듈은 앞서 계산한 attention score와 기본 마스크를 바탕으로 최종 마스크를 계산한다. 관측된 객체 i에 대하여, 다음과 같이 attention score와 기본 마스크를 성분 단위로 곱한 결과를 합산하여 마스크 M^i를 계산한다.

$$M^i = \sum_{j=1}^{r_1 \times r_2} \sigma(a^j \odot B^j). \tag{4}$$

여기서 $\sigma(\cdot)$은 sigmoid 함수를 의미한다. 해당 프레임에서 n개의 객체가 관측되었다면, 해당 프레임의 객체 마스크는 다음과 같이 구하게 된다.

$$M = \sum_{i=1}^{n} M^i. \tag{5}$$

SG-Net의 마스크 헤드 부분은 얼핏 보기에 대량의 객체를 포함한 프레임을 처리할 때 매우 많은 네트워크 변수(parameter)가 필요해보이지만, 2단계 절차 모델의 비효율적인 후보 경계 상자 생성과 RoI 연산에 비하면 훨씬 빠르고 더 좋은 품질의 마스크를 생성하게 된다.

4.2.2 추적 헤드

SG-Net의 추적 방법은 FCOS으로부터 구한 객체의 중심성 위치(centerness location)을 추적하여, 객체의 외관 변화에 강인하게 이루어진다[4]. SG-Net의 추적 헤드는 시간에 따른 중심성 위치의 변화를 나타내는 움직임 맵 $D \in \mathbb{R}^{\widetilde{H}\times\widetilde{W}\times 2}$를 계산한다[5]. 시간 t에 대해 관측된 객체 i의 중심성 위치를 $o_t^i \in \mathbb{R}^2$라 하면, D_t^i는 시간 $t-1$에서 t로의 중심성 위치의 변화 $o_t^i - o_{t-1}^i$를 나타낸다. 따라서 추적 손실 $\mathcal{L}_{track}$은 다음과 같이 정의한다.

$$\mathcal{L}_{track} = \frac{1}{N} \sum_{1 \le t \le T} \sum_{1 \le i \le N} \left| D_t^i - (o_t^i - o_{t-1}^i) \right|. \tag{6}$$

여기서 N은 비디오에서 관측된 모든 객체의 수를, T는 비디오의 전체 프레임을 나타낸다.

[3]여기서 W와 H는 프레임의 크기로 판단된다.

[4]이 논문에서 말하는 중심성 위치는 특징맵에서 해당 객체의 경계 상자를 그려낸 위치 정보를 의미하는 것으로 판단되며, FCOS [9]에서 언급한 중심성(centerness)과는 다른 것으로 보인다.

[5]여기서 $\widetilde{W}$와 $\widetilde{H}$는 특징맵의 크기로 판단된다.

추론 단계에서는 간단한 greedy matching 방법으로 객체를 추적한다. 시간 t에 대해 관측된 객체 i의 위치가 o_t^i라 할 때, $o_t^i - D_t^i$를 계산하여 가장 가까운 관측 객체의 레이블을 할당한다. 만약 주어진 반경 r 이내에 마땅한 후보 객체가 존재하지 않는다면 새로운 추적 경로(tracklet)을 생성한다[6].

4.2.3 경계 상자 손실

SG-Net은 객체 탐지 결과로부터 경계 상자 위치를 이용하여 객체 분할 결과를 계산하는 top-down 방식을 취하고 있다. 따라서 향상된 경계 상자 회귀 결과가 객체 분할 결과에 좋은 영향을 준다고 볼 수 있다. 이런 관점에서, 다음과 같은 경계 상자 손실을 사용함으로써 작은 generalized IoU (GIoU) [12]를 보이는 hard sample 학습을 용이하게 한다.

$$\mathcal{L}_{box} = -\log \frac{1 + GIoU}{2} \tag{7}$$

따라서 전체 네트워크의 손실은 $\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{box} + \mathcal{L}_{cent} + \mathcal{L}_{mask} + \mathcal{L}_{track}$으로 정의하며, 이를 통해 전체 네트워크를 end-to-end로 학습하게 된다. 여기서 $\mathcal{L}_{cls}$와 $\mathcal{L}_{cent}$는 FCOS의 손실과 동일하며, $\mathcal{L}_{box}$는 수식 (7)과 같이 계산되며, $\mathcal{L}_{mask}$는 Mask R-CNN [5]의 마스크 손실, $\mathcal{L}_{track}$은 수식 (6)과 같이 계산된다.

4.3 선행 기술 대비 장점

SG-Net은 MaskTrack R-CNN과 같은 2단계 절차 모델들과 달리, 1단계 절차 모델인 FCOS를 기반으로 하여 각 하위 작업 간 특징 정보를 효과적으로 공유하여 모델의 전체적인 최적화가 될 수 있도록 만들어졌다. 또, 마스크 계산 과정이 객체의 하위 영역별로 이루어지면서 보다 세밀한 마스크를 얻을 수 있다. SG-Net의 중심성 기반의 추적은 FCOS와 매끄럽게 결합되는 동시에, 외관 정보의 변화에 강인하게 동작하여 두 객체가 상당 부분 겹치는 어려운 상황에서도 준수한 성능을 보여준다.

종합하면, SG-Net은 이미지 객체 탐지의 1단계 절차 모델의 장점을 잘 살리는 방향으로 비디오 객체 분할 문제를 해결한 모델이라 할 수 있다. 특이한 점은, SG-Net 자체는 비디오 객체 분할 문제에 초점을 맞추고 있지만 주요 기여점은 마스크 헤드 부분으로 기존의 이미지 객체 분할 모델보다 더 좋은 성능을 보여준다고 주장한 점이다. 이는 SG-Net 연구진은 비디오의 개별 프레임의 객체 분할 성능을 끌어올리는 것이 전체 비디오 객체 분할 성능을 높이는 방향이라고 판단한 것으로 보인다. 즉, SG-Net은 전반적인 효율성와 마스크 품질 향상을 이끌어냈으나, 아직 비디오에서 얻을 수 있는 여러 정보들을 제대로 활용한 모델로 보기는 어렵다고 판단할 수 있다.

5 CrossVIS

이미지 객체 분할 문제와 달리, 비디오 객체 분할 문제에서는 프레임 간 동일 객체의 시간적 일관성이 중요한 역할을 할 수 있다. 예를 들어, 비디오 안에서 어느 한 프레임에서 나타난 특정 객체의 외관 정보(appearance information)를 다른 프레임에서의 해당 객체의 외관 정보로 표현하는 것이 가능할 것이다. CrossVIS [13]는 이 점을 착안하여, 프레임 간 객체의 맥락 정보(contextual information)를 활용하여

[6]명시적인 언급은 없으나, 관측된 객체 i에 새로운 객체 레이블을 할당하는 것으로 판단된다.

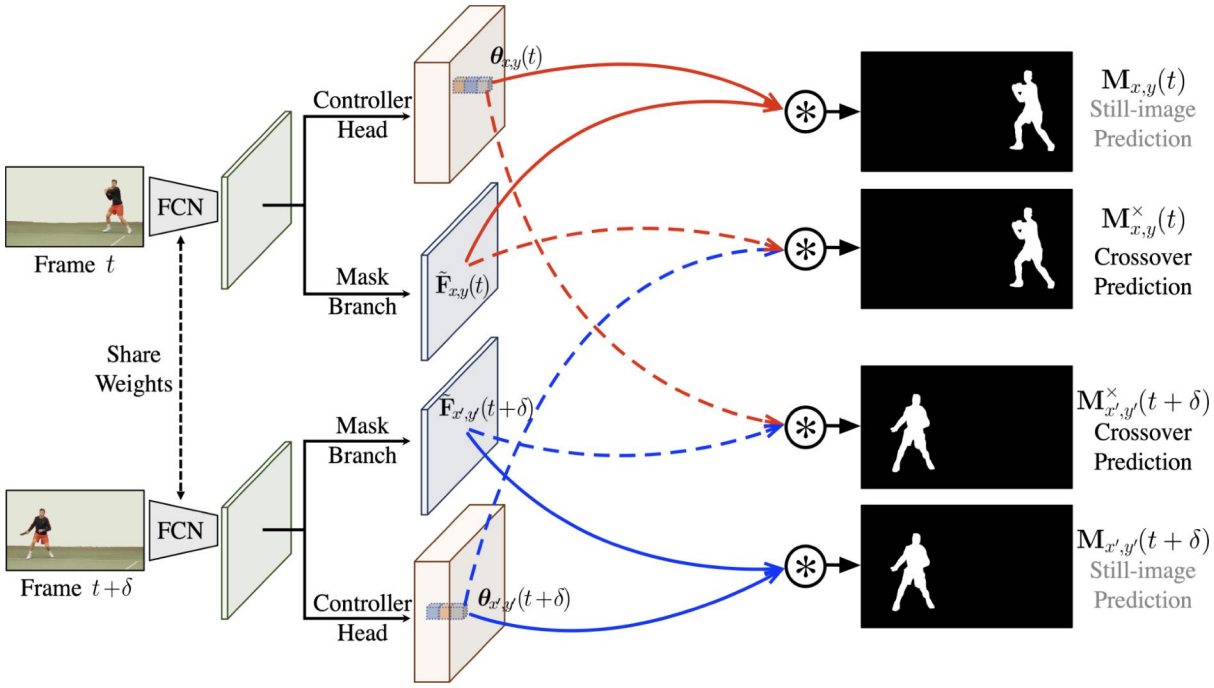

그림 4: CrossVIS 구조도

객체의 외관 정보는 강화하고 그 이외의 정보는 약화시키는 크로스오버 학습(crossover learning) 체계를 제안한다. 본 장에서는 CrossVIS의 전체적인 구조와 주요 요소, 선행 기술과 비교했을 때의 장점을 알아본다.

5.1 CrossVIS의 구조

그림 4는 CrossVIS의 구조도 중 마스크 생성과 관련된 부분을 나타낸 것이다. CrossVIS는 이미지 객체 분할 모델 중 1단계 절차 모델인 CondInst [8]를 기반으로 하여 만들어졌다. CondInst에는 FPN의 P3, P4, P5와 연결된 별도의 마스크 브랜치가 있어, 각 객체의 마스크를 추출하는 마스크 헤드의 입력에 해당하는 특징맵 $\mathbf{F}_{mask}$를 생성한다. 이때, $\mathbf{F}_{mask}$에, 현 위치 (x, y)를 중심으로 하는 상대 좌표(relative coordinates) $\mathbf{O}_{x,y}$를 붙여서 $\widetilde{\mathbf{F}}_{x,y}$를 생성한다. 그리고 $\widetilde{\mathbf{F}}_{x,y}$를 다음과 같이 마스크 헤드에 통과시켜 (x, y)에서의 객체 마스크 $\mathbf{M}_{x,y}$를 생성한다.

$$\mathbf{M}_{x,y} = \text{MaskHead}(\widetilde{\mathbf{F}}_{x,y}; \boldsymbol{\theta}_{x,y}). \tag{8}$$

이때, $\boldsymbol{\theta}_{x,y}$는 별도 컨트롤러(controller) 헤드에서 생성된 각 객체별(instance-specific) 동적 필터(dynamic filter)로, 객체의 외관 정보를 담고 있는 마스크 헤드의 모수(parameter)이다.

CrossVIS는 더 정확한 객체 정보를 얻어내기 위해 추가적으로 크로스오버 학습 체계를 갖추고 있다. 그림 4에서 빨간색, 파란색 선은 각각 시간 t와 $t+\delta$에서의 동적 필터 $\boldsymbol{\theta}_{x,y}(t), \boldsymbol{\theta}_{x',y'}(t+\delta)$와 특징맵 $\widetilde{\mathbf{F}}_{x,y}(t), \widetilde{\mathbf{F}}_{x',y'}(t+\delta)$의 적용 방향을 나타낸다. 실선은 각 프레임별 객체 분할 과정을, 점선은 크로스오버 학습 체계를 나타낸다.

5.1.1 크로스오버 학습 체계

크로스오버 학습 체계를 설명하기 위해, 관측된 객체 i가 시간 t와 $t+\delta$에서 모두 존재한다고 가정한다. 먼저 시간 t에 대해, 현 위치 (x, y)에서의 객체 i의 마스크는 다음과 같이 계산된다.

$$\mathbf{M}_{x,y}(t) = \text{MaskHead}(\widetilde{\mathbf{F}}_{x,y}(t); \boldsymbol{\theta}_{x,y}(t)). \tag{9}$$

시간 $t+\delta$에 대해, 객체 i는 (x, y)에서 (x', y')로 이동한다고 하면 객체 i의 마스크는 다음과 같이 계산된다.

$$\mathbf{M}_{x',y'}(t+\delta) = \text{MaskHead}(\widetilde{\mathbf{F}}_{x',y'}(t+\delta); \boldsymbol{\theta}_{x',y'}(t+\delta)). \tag{10}$$

크로스오버 학습 체계는 한 프레임에서의 동적 필터와 다른 프레임에서의 마스크 정보를 연결시킨다. 구체적으로, 객체 i에 대해 시간 t에서의 동적 필터 $\boldsymbol{\theta}_{x,y}(t)$를 통해 다음과 같이 시간 $t+\delta$에서의 마스크를 계산할 수 있어야 한다.

$$\mathbf{M}^{\times}_{x',y'}(t+\delta) = \text{MaskHead}(\widetilde{\mathbf{F}}_{x',y'}(t+\delta); \boldsymbol{\theta}_{x,y}(t)). \tag{11}$$

마찬가지로, 객체 i에 대해 시간 $t+\delta$에서의 동적 필터 $\boldsymbol{\theta}_{x',y'}(t+\delta)$를 통해 다음과 같이 시간 t에서의 마스크를 계산할 수 있어야 한다.

$$\mathbf{M}^{\times}_{x,y}(t) = \text{MaskHead}(\widetilde{\mathbf{F}}_{x,y}(t); \boldsymbol{\theta}_{x',y'}(t+\delta)). \tag{12}$$

학습 과정에서는 예측한 마스크 $\mathbf{M}_{x,y}(t), \mathbf{M}_{x',y'}(t+\delta), \mathbf{M}^{\times}_{x',y'}(t+\delta), \mathbf{M}^{\times}_{x,y}(t)$ 모두 dice loss [14]를 통해 최적화된다.

추론 단계에서는 크로스오버 체계 없이 3.2.3절과 같은 방법으로 마스크를 생성한다.

5.1.2 추적 방법

비디오 객체 분할 문제에서 또다른 중요한 하위 작업은 바로 추적이다. 이전의 추적 방법은 3.2.1절에서도 언급했던 수식 (1)을 사용하는 것이다. 하지만 이 접근법은 후보 객체 i의 특징 벡터 f_i는 현재 프레임에서 생성된 특징 공간에 위치한 반면 이미 관측된 N개의 특징 벡터 g_j는 이전 프레임에서 생성된 특징 공간에 위치하기 때문에, 후보 객체 i의 레이블 선정 과정이 이전 프레임의 특징 공간과 밀접하게 연관된다. 따라서 추적 과정 자체가 샘플링 방법에 따라 크게 흔들리게 되어 전체적인 학습이 불안정해지고 수렴 속도도 느려지게 된다.

이러한 문제를 해결하기 위해, 객체의 특징 벡터들을 전역(global) 공간에서 수렴하도록 하기 위한 M-클래스 분류 문제로 모델을 학습시킨다. 여기서 M은 전체 학습 데이터셋에서 등장하는 모든 객체의 수를 나타낸다. 각 객체의 특징 벡터를 $w_1, \cdots, w_M$이라 할 때[7], 후보 객체 i에 레이블 n을 할당하는 확률은 다음과 같이 정의한다.

$$p_i(n) = \frac{\exp(f_i^T w_j)}{\sum_{j=1}^{M} \exp(f_i^T w_j)}, \quad n = 1, \cdots, M. \tag{13}$$

[7]이 특징 벡터들도 학습 대상이다.

하지만 M-클래스 분류 문제는 굉장히 큰 데이터셋에서는 negative 클래스가 너무나 많아져서 데이터 불균형(imbalance) 문제가 발생한다. 이를 해결하기 위해 다음과 같이 $p_i(n)$를 정의하고 focal loss [15]를 사용하여 추적 손실 $\mathcal{L}_{track}$을 정의한다.

$$p_i(n) = \begin{cases} \sigma(f_i^T w_n), & \text{candidate } i \text{ belongs to the } n\text{th identity,} \\ 1 - \sigma(f_i^T w_n), & \text{otherwise.} \end{cases} \tag{14}$$

여기서 $\sigma(\cdot)$은 sigmoid 함수를 의미한다.

추론 단계에서는 3.2.3절과 같은 온라인 방법으로 마스크 추적을 수행한다.

5.2 선행 기술 대비 장점

CrossVIS는 시간적 일관성을 고려한 크로스오버 학습 체계를 도입하면서 추가적인 네트워크 블록(block)을 도입하지 않아, 추론 시에 추가적인 계산 비용 없이 성능 향상이 가능하다. 또, 크로스오버 학습 체계는 별도의 추가적인 손실을 도입하지 않고 기존의 객체 분할 손실과 통합되므로 시간 정보를 활용하는 다른 방법들을 어려움 없이 적용하는 것도 가능하다. 추적 과정도 전역 공간에서 모든 객체의 특징 정보를 수렴시키는 방향으로 학습함으로써 추적 성능의 안정성(stability)을 올릴 수 있었다.

종합하면, CrossVIS는 비디오 객체 분할 문제에서, 비디오가 주는 시간적 특징들을 잘 활용하여 별도의 오버헤드(overhead) 없이 전반적인 성능을 향상시킨 모델이라 할 수 있다. 기존의 비디오 객체 분할 연구들은 개별 프레임에서 한정된 정보만으로 객체 분할을 독립적으로 수행한 후 프레임 간 마스크를 연결시키는 형태로 진행하였지만, CrossVIS는 동일 객체의 시간적 일관성을 고려하여 모델에 큰 변화 없이 여러 프레임의 많은 정보를 보고 객체 분할을 수행하여 전반적인 결과를 향상시킬 수 있다는 점을 보여주었다. 따라서 향후 비디오 객체 분할 연구에서는 CrossVIS처럼 비디오에서 얻을 수 있는 시간적 특징을 어떻게 잘 활용할 수 있을 것인가를 중점적으로 공략할 것으로 전망해볼 수 있다.

참고문헌

[1] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Object instance segmentation and fine-grained localization using hypercolumns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 627–639, 2017.

[2] L. Yang, Y. Fan, and N. Xu, "Video instance segmentation," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 5187–5196.

[3] P. Voigtlaender, M. Krause, A. Ošep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "Mots: Multi-object tracking and segmentation," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7934–7943.

[4] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, "Blendmask: Top-down meets bottom-up for instance segmentation," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8570–8578.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, 2020.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[7] D. Liu, Y. Cui, W. Tan, and Y. Chen, "Sg-net: Spatial granularity network for one-stage video instance segmentation," in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9811–9820.

[8] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *Proceedings of the 2020 European Conference on Computer Vision (ECCV)*, 2020, pp. 282–298.

[9] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9626–9635.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[11] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

[12] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 658–666.

[13] S. Yang, Y. Fang, X. Wang, Y. Li, C. Fang, Y. Shan, B. Feng, and W. Liu, "Crossover learning for fast online video instance segmentation," in *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 8043–8052.

[14] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV)*, 2016, pp. 565–571.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.