(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0061495
(43) 공개일자 2025년05월08일

(51) 국제특허분류(Int. Cl.)
G06V 20/40 (2022.01) *G06V 10/34* (2022.01)
G06V 10/764 (2022.01)
(52) CPC특허분류
G06V 20/46 (2022.01)
G06V 10/34 (2023.08)
(21) 출원번호 10-2023-0145995
(22) 출원일자 2023년10월27일
심사청구일자 2023년10월27일

(71) 출원인
에스케이텔레콤 주식회사
서울특별시 중구 을지로 65 (을지로2가)
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
백정렬
서울특별시 중구 을지로 65 SKT-타워
이병원
서울특별시 중구 을지로 65 SKT-타워
(뒷면에 계속)
(74) 대리인
이철희

전체 청구항 수 : 총 12 항

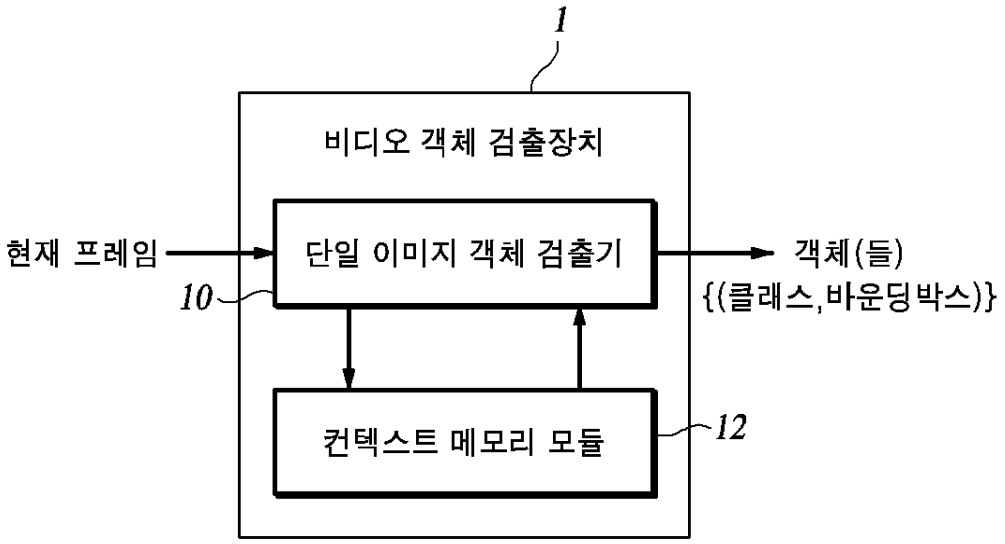
(54) 발명의 명칭 **단일 이미지를 이용하여 비디오에서 객체를 검출하기 위한 방법 및 장치**

(57) 요 약

단일 이미지를 이용하여 비디오에서 객체를 검출하기 위한 방법 및 장치를 개시한다.

본 개시의 일 측면에 의하면, 비디오에서 객체를 검출하기 위한 컴퓨터 구현 방법으로서, 현재 프레임에서 추출된 이미지 특성맵 및 미리 저장된 메모리 특성맵을 인코딩하는 과정 - 상기 메모리 특성맵은, 예측 가능한 하나 이상의 클래스들 각각에 대해 미리 정의된 개수의 메모리 특성들을 포함함 - ; 상기 인코딩된 메모리 특성맵으로부터 산출된 분류 스코어들 및 상기 메모리 특성맵을 기초로, 샘플링된 메모리 특성맵을 생성하는 과정; 상기 인코딩된 이미지 특성맵 및 상기 샘플링된 메모리 특성맵을 기초로, 상기 현재 프레임 내의 객체의 바운딩박스 및 클래스를 예측하는 과정; 및 상기 인코딩된 이미지 특성맵을 기초로, 상기 메모리 특성맵을 업데이트하는 과정을 포함하는, 방법을 제공한다.

대 표 도 - 도1



(52) CPC특허분류
G06V 10/764 (2023.08)
G06V 20/41 (2022.01)
G06T 2210/12 (2013.01)
(72) 발명자
김승룡
서울특별시 서대문구 증가로 150, 108-1501
김경년
서울특별시 성북구 고려대로 47, 207호
박성훈
서울특별시 노원구 상계로23길 57-19, 503호
안승준
서울 성북구 고려대로2길 53, 4층 1호
강충헌
서울특별시 중구 을지로 65 SKT-타워

명 세 서

청구범위

청구항 1

비디오에서 객체를 검출하기 위한 컴퓨터 구현 방법으로서,

현재 프레임으로부터 추출된 이미지 특성맵 및 미리 저장된 메모리 특성맵을 인코딩하는 과정 - 상기 메모리 특성맵은, 예측 가능한 하나 이상의 클래스들 각각에 대해 미리 정의된 개수의 메모리 특성들을 포함함 - ;

상기 인코딩된 메모리 특성맵으로부터 산출된 분류 스코어들 및 상기 메모리 특성맵을 기초로, 샘플링된 메모리 특성맵을 생성하는 과정;

상기 인코딩된 이미지 특성맵 및 상기 샘플링된 메모리 특성맵을 기초로, 상기 현재 프레임 내의 객체의 바운딩박스 및 클래스를 예측하는 과정; 및

상기 인코딩된 이미지 특성맵을 기초로, 상기 메모리 특성맵을 업데이트하는 과정을 포함하는, 방법

청구항 2

제1항에 있어서,

상기 업데이트하는 과정은,

상기 예측된 바운딩박스 또는 미리 지정된 GT(ground truth) 바운딩박스를 기초로, 상기 인코딩된 이미지 특성맵에서 하나 이상의 인스턴스 특성들을 추출하는 과정; 및

상기 메모리 특성맵에서, 상기 추출된 인스턴스 특성들에 대응하는 하나 이상의 타겟 메모리 특성들을 업데이트하는 과정 - 각 타겟 메모리 특성은, 각 인스턴스 특성과 동일한 클래스에 대응하는 상기 메모리 특성들 중에서 선별된 것 임 - 을 더 포함하는, 방법.

청구항 3

제2항에 있어서,

상기 타겟 메모리 특성들을 업데이트 하는 과정은,

각 인스턴스 특성 및 상기 각 인스턴스 특성에 대응하는 타겟 메모리 특성 간의 선형 보간에 기초하여, 상기 각 인스턴스 특성에 대응하는 타겟 메모리 특성을 업데이트하는, 방법.

청구항 4

제2항에 있어서,

상기 업데이트하는 과정은,

각 인스턴스 특성과 동일한 클래스에 대응하는 상기 메모리 특성들 중, 상기 각 인스턴스 특성과 가장 큰 상관관계(correlation)를 갖는 메모리 특성을 상기 타겟 메모리 특성으로 선별하는 과정을 더 포함하는, 방법.

청구항 5

제1항에 있어서,

상기 인코딩하는 과정은,

상기 이미지 특성맵 및 상기 메모리 특성맵을 연결(concatenate)하는 과정; 및

상기 연결된 특성맵에 대한 셀프-어텐션(self-attention)을 기초로, 상기 현재 프레임 이전에 관측된 전체 프레임들의 시공간 컨텍스트가 내재된 상기 인코딩된 이미지 특성맵 및 상기 현재 프레임 내 객체의 클래스에 관한 정보가 내재된 상기 인코딩된 메모리 특성맵을 획득하는 과정을 포함하는, 방법.

청구항 6

제1항에 있어서,

상기 인코딩된 메모리 특성맵 및 상기 샘플링된 메모리 특성맵은, 각 메모리 특성에 대응하는 인코딩된 메모리 특성 및 샘플링된 메모리 특성을 각각 포함하고,

상기 생성하는 과정은,

각 인코딩된 메모리 특성에 대해 분류 스코어를 산출하는 과정;

기정의된 하나 이상의 임계치들에 기초하여, 상기 각 인코딩된 메모리 특성에 대응하는 메모리 특성을 임계화하는 과정; 및

하나 이상의 임계화된 메모리 특성들을 임베딩하여, 각 메모리 특성에 대응하는 샘플링된 메모리 특성을 생성하는 과정을 포함하는, 방법.

청구항 7

제6항에 있어서,

상기 임계화하는 과정은,

기정의된 하나 이상의 임계치들을 기초로 상기 분류 스코어를 이진화하여, 하나 이상의 샘플링 인덱스들을 획득하는 과정; 및

각 샘플링 인덱스의 값을 기초로, 상기 대응하는 메모리 특성 및 클래스-없음 임베딩 중 하나를 임계화된 메모리 특성으로 선택하는 과정을 포함하는 방법.

청구항 8

제1항에 있어서,

상기 예측하는 과정은,

객체 쿼리를 메모리-유도 디코더에 적용하여, 상기 현재 프레임에 대한 시맨틱 정보 및 현재 프레임과 연관된 클래스에 관한 정보가 향상된 출력 객체 쿼리를 획득하는 과정; 및

상기 출력 객체 쿼리로부터 상기 객체의 바운딩박스 및 클래스를 예측하는 과정을 포함하는, 방법.

청구항 9

제8항에 있어서,

상기 메모리-유도 디코더는, 하나 이상의 디코더 블록을 포함하되,

각 디코더 블록은,

입력된 객체 쿼리에 대한 셀프-어텐션을 기초로, 제1 객체 쿼리를 생성하는 제1 어텐션 레이어;

상기 제1 객체 쿼리와 상기 인코딩된 이미지 특성맵 간의 크로스-어텐션(cross-attention)을 기초로, 제2 객체 쿼리를 생성하는 제2 어텐션 레이어; 및

상기 제2 객체 쿼리와 상기 샘플링된 메모리 특성맵 간의 크로스-어텐션을 기초로, 제3 객체 쿼리를 생성하는 제3 어텐션 레이어를 포함하는, 방법.

청구항 10

제9항에 있어서,

상기 샘플링된 메모리 특성맵은, 각 메모리 특성에 대응하는 샘플링된 메모리 특성을 각각 포함하고,

상기 출력 객체 쿼리를 획득하는 과정은,

각 샘플링된 메모리 특성에 대응하는 메모리 특성들의 위치를 나타내는 메모리 임베딩들을 상기 샘플링된 메모

리 특성맵에 결합하여 상기 제3 어텐션 레이어에 제공하는 과정

을 포함하는, 방법.

청구항 11

명령어들을 저장하는 메모리; 및 적어도 하나의 프로세서를 포함하되,

상기 적어도 하나의 프로세서는 상기 명령어들을 실행함으로써,

현재 프레임에서 추출된 이미지 특성맵 및 미리 저장된 메모리 특성맵을 인코딩하고 - 상기 메모리 특성맵은, 예측 가능한 클래스들 각각에 대해 미리 정의된 개수의 메모리 특성들을 포함함 - ,

상기 인코딩된 메모리 특성맵으로부터 산출된 분류 스코어들 및 상기 메모리 특성맵을 기초로, 샘플링된 메모리 특성맵을 생성하고,

상기 인코딩된 이미지 특성맵 및 상기 샘플링된 메모리 특성맵을 기초로, 상기 현재 프레임 내의 객체의 바운딩 박스 및 클래스를 예측하고,

상기 인코딩된 이미지 특성맵을 기초로, 상기 메모리 특성맵을 업데이트하는, 장치.

청구항 12

제1항 내지 제10항 중 어느 한 항에 따른 방법이 포함하는 각 과정을 실행시키기 위하여 컴퓨터로 읽을 수 있는 기록매체에 저장된 컴퓨터프로그램.

발명의 설명

기 술 분 야

[0001] 본 개시는 단일 이미지를 이용하여 비디오에서 객체를 검출하기 위한 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] 이하에 기술되는 내용은 단순히 본 실시예와 관련되는 배경 정보만을 제공할 뿐 종래기술을 구성하는 것이 아니다.

[0003] 객체 검출(Object Detection)은 컴퓨터 비전 분야에서 기본적이고 필수적인 작업 중 하나로, 다양한 응용 분야에서 광범위하게 사용되고 있다. 특히, CCTV, 자율 주행 및 로봇 내비게이션 등의 다양한 응용 분야에서는 비디오 데이터에 대한 객체 검출이 활용되고 있다. 단일 이미지에 대한 객체 검출 모델은 높은 성공률을 보이고 있지만, 이러한 모델을 비디오 데이터에 직접 적용할 경우 움직임과 가려짐(occlusion)에 의한 영상 내 물체의 변형으로 인해 성능 저하가 발생한다.

[0004] 이러한 문제를 해결하기 위해, 비디오 데이터의 처리에 특화된 비디오 객체 검출(Video Object Detection, VOD) 모델들이 제안되고 있다. 비디오 객체 검출 모델들은, 이미지의 시퀀스를 처리하기 위해, 광학 흐름(optical flow), LSTM(Long Short-Term Memory) 또는 어텐션 메커니즘(attention mechanism) 등을 활용하고 있다. 광학 흐름 또는 LSTM을 이용하는 방식은 주로 현재 프레임과 가까운 단기간의 프레임들에 중점을 두기 때문에 더 넓은 특성 표현을 캡처하는 데 제한이 있다. 한편, 어텐션 메커니즘에 기반한 방식은 무작위로 샘플링된 이미지로부터 전반적인 컨텍스트 정보를 얻기 때문에, 비디오 데이터의 전반적인 정보를 통합하는데 어려움이 있다. 또한, 비디오 객체 검출 모델들에서는 인접한 프레임의 특성을 활용하기 위해, 참조 프레임을 추가로 입력받거나 선행하는 전체 프레임의 정보를 누적하여 저장하고 있는데, 이로 인해 불필요한 정보까지 참조하거나, 높은 계산 비용 및 불필요한 메모리 사용이 발생한다는 단점이 있다.

발명의 내용

해결하려는 과제

[0005] 본 개시는, 주어진 데이터 전체에 걸쳐 컨텍스트 정보를 효과적으로 통합할 수 있는 단일 이미지 기반 비디오 객체 검출 방법 및 장치를 제공하는 데 일 목적이 있다.

[0006] 본 발명이 해결하고자 하는 과제들은 이상에서 언급한 과제들로 제한되지 않으며, 언급되지 않은 또 다른 과제

들은 아래의 기재로부터 통상의 기술자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0007] 본 개시의 일 측면에 의하면, 비디오에서 객체를 검출하기 위한 컴퓨터 구현 방법으로서, 미리 저장된 메모리 특성맵 - 상기 메모리 특성맵은, 예측 가능한 클래스들 각각에 대해 미리 정의된 개수의 메모리 특성들을 포함함 - 및 현재 프레임에서 추출된 이미지 특성맵을 인코딩하는 과정; 상기 인코딩된 메모리 특성맵으로부터 산출된 분류 스코어들 및 상기 메모리 특성맵을 기초로, 샘플링된 메모리 특성맵을 생성하는 과정; 상기 인코딩된 이미지 특성맵 및 상기 샘플링된 메모리 특성맵을 기초로, 상기 현재 프레임 내의 객체의 바운딩박스 및 클래스를 예측하는 과정; 및 상기 인코딩된 이미지 특성맵을 기초로, 상기 메모리 특성맵을 업데이트하는 과정을 포함하는, 방법을 제공한다.

[0008] 본 개시의 다른 측면에 의하면, 명령어들을 저장하는 메모리; 및 적어도 하나의 프로세서를 포함하되, 상기 적어도 하나의 프로세서는 상기 명령어들을 실행함으로써, 미리 저장된 메모리 특성맵 - 상기 메모리 특성맵은, 예측 가능한 클래스들 각각에 대해 미리 정의된 개수의 메모리 특성들을 포함함 - 및 현재 프레임에서 추출된 이미지 특성맵을 인코딩하고, 상기 인코딩된 메모리 특성맵으로부터 산출된 분류 스코어들 및 상기 메모리 특성맵을 기초로, 샘플링된 메모리 특성맵을 생성하고, 상기 인코딩된 이미지 특성맵 및 상기 샘플링된 메모리 특성맵을 기초로, 상기 현재 프레임 내의 객체의 바운딩박스 및 클래스를 예측하고, 상기 인코딩된 이미지 특성맵을 기초로, 상기 메모리 특성맵을 업데이트하는, 장치를 제공한다.

[0009] 본 개시의 또 다른 측면에 의하면, 전술한 방법이 포함하는 각 과정을 실행시키기 위하여 컴퓨터로 읽을 수 있는 기록매체에 저장된 컴퓨터프로그램을 제공한다.

발명의 효과

[0010] 본 개시의 실시예에 의하면, 컨텍스트 메모리 모듈(context memory module)을 활용함으로써, 관측된 데이터 전체에 걸쳐 컨텍스트 정보를 효과적으로 통합할 수 있다. 이를 통해 카메라가 설치된 환경에 대한 이해를 바탕으로, 강인한 객체 검출이 가능하다는 효과가 있다.

[0011] 본 개시의 실시예에 의하면, 추가적인 참조 프레임을 입력하거나 보조 네트워크를 사용하지 않고도, 이전 프레임들에서의 시간적 컨텍스트를 활용할 수 있다. 이에 따라, 참조 프레임 또는 보조 네트워크로 인한 처리 속도의 감소를 방지할 수 있어, 실제 응용 프로그램에 필수적인 실시간 요구 사항을 충족시킬 수 있다는 효과가 있다.

[0012] 본 개시의 실시예에 의하면, 현재 프레임에서 추출 및 가공된 정보 중 필요한 정보만을 선택적으로 저장하여, 컨텍스트 메모리 모듈을 고정된 크기로 유지할 수 있다. 또한, 스코어 기반 샘플링을 통해, 컨텍스트 메모리 모듈에서 현재 프레임과 관련된 정보를 효과적으로 캡처할 수 있다.

[0013] 본 개시의 효과들은 이상에서 언급한 효과들로 제한되지 않으며, 언급되지 않은 또 다른 효과들은 아래의 기재로부터 통상의 기술자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0014] 도 1은 본 개시의 일 실시예에 따른 비디오 객체 검출장치를 개략적으로 나타낸 블록구성도이다.

도 2는 본 개시의 일 실시예에 따른 단일 이미지 객체 검출기의 구조를 설명하기 위해 참조되는 도면이다.

도 3은 본 개시의 일 실시예에 따른 업데이트 모듈의 동작을 설명하기 위해 참조되는 도면이다.

도 4는 본 개시의 일 실시예에 따른 샘플링 모듈의 동작을 설명하기 위해 참조되는 도면이다.

도 5a 및 도 5b는 본 개시의 다양한 실시예에 따른 디코더 블록의 구조를 설명하기 위해 참조되는 도면이다,

도 6은 본 개시의 일 실시예에 따른 객체 검출방법을 나타내는 흐름도이다.

도 7은 본 개시에서 설명된 장치 및 방법들을 구현하기 위해 사용될 수 있는 예시적인 컴퓨팅 장치를 개략적으로 나타낸 블록구성도이다.

발명을 실시하기 위한 구체적인 내용

[0015] 이하, 본 개시의 일부 실시예들을 예시적인 도면을 이용해 상세하게 설명한다. 각 도면의 구성 요소들에 참조 부호를 부가함에 있어서, 동일한 구성 요소들에 대해서는 비록 다른 도면 상에 표시되더라도 가능한 한 동일한 부호를 가지도록 하고 있음에 유의해야 한다. 또한, 본 개시를 설명함에 있어, 관련된 공지 구성 또는 기능에 대한 구체적인 설명이 본 개시의 요지를 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명은 생략한다.

[0016] 본 개시에 따른 실시예의 구성요소를 설명하는 데 있어서, 제1, 제2, i), ii), a), b) 등의 부호를 사용할 수 있다. 이러한 부호는 그 구성요소를 다른 구성 요소와 구별하기 위한 것일 뿐, 그 부호에 의해 해당 구성요소의 본질 또는 차례나 순서 등이 한정되지 않는다. 명세서에서 어떤 부분이 어떤 구성요소를 '포함' 또는 '구비'한다고 할 때, 이는 명시적으로 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0017] 첨부된 도면과 함께 이하에 개시될 상세한 설명은 본 개시의 예시적인 실시형태를 설명하고자 하는 것이며, 본 개시가 실시될 수 있는 유일한 실시형태를 나타내고자 하는 것이 아니다.

[0018] 도 1은 본 개시의 일 실시예에 따른 비디오 객체 검출장치를 개략적으로 나타낸 블록구성도이다. 도 2는 본 개시의 일 실시예에 따른 컨텍스트 메모리 모듈에 저장되는 메모리 특성을 예시한 도면이다.

[0019] 도 1에 도시되듯이, 본 개시의 일 실시예에 따른 비디오 객체 검출장치(1)는 단일 이미지 객체 검출기(10) 및 컨텍스트 메모리 모듈(12)을 전부 또는 일부 포함할 수 있다. 도 1에 도시된 모든 블록이 필수 구성요소인 것은 아니며, 다른 실시예에서 일부 블록이 추가, 변경 또는 삭제될 수 있다. 한편, 도 1에 도시된 구성요소들은 기능적으로 구분되는 요소들을 나타낸 것으로서, 적어도 하나의 구성요소가 실제 물리적 환경에서는 서로 통합되는 형태로 구현될 수도 있다.

[0020] 단일 이미지 객체 검출기(10)는 현재 프레임으로부터 하나 이상의 객체들을 검출한다. 객체를 검출하는 것은, 객체의 클래스(class)에 대한 정보 및 해당 객체를 감싸는 바운딩박스(bounding box)에 대한 정보를 예측하는 것을 포함할 수 있다. 클래스에 대한 정보는, 객체가 미리 정의된 복수개의 클래스들 각각에 속할 확률 및/또는 가장 높은 확률을 갖는 클래스의 식별자를 포함할 수 있다. 바운딩박스에 대한 정보는, 예컨대, 바운딩박스의 좌상단 좌표, 바운딩박스의 우하단 좌표, 바운딩박스의 중심 좌표, 및/또는 바운딩박스의 크기(너비 및 높이) 중 2 이상의 조합을 포함할 수 있다.

[0021] 단일 이미지 객체 검출기(10)는 신경망(neural network) 기반의 모델로 구현될 수 있다. 본 개시를 설명함에 있어 단일 이미지 객체 검출기(10)는, 적어도 일부 구성요소들이 트랜스포머(Transformer)의 검출기인 DETR(DEtection TRansformer)의 아키텍처를 따르는 것으로 가정한다. 그러나 본 개시의 기술적 사상은, 컨텍스트 메모리 모듈(12)과 연동하여 동작할 수 있는 한, 다른 유형의 아키텍처를 갖는 단일 이미지 객체 검출기에 대해서도 적용될 수 있음에 유의하여야 한다.

[0022] 컨텍스트 메모리 모듈(12)은 선행하는 프레임들의 시간적 및/또는 공간적 컨텍스트를 고정된 크기로 저장한다. 컨텍스트 메모리 모듈(12)은 예측 가능한 클래스들 각각에 대해, 고정된 개수의 클래스별 특성 표현들(class-wise feature representations)을 저장할 수 있다. 특성 표현은 메모리 특성 또는 프로토타입(prototype)으로 지칭될 수 있으며, 이들의 집합은 클래스별 메모리 특성맵(class-wise memory feature map) 또는 메모리 특성맵으로 지칭될 수 있다. 예컨대, 컨텍스트 메모리 모듈(12)은 각 클래스에 대해 복수개의 프로토타입을 가진, 멀티 프로토타입(multi-prototype class-wise memory) 클래스별 메모리 특성맵을 저장할 수 있다. 예측 가능한 클래스들의 개수를 C라 하고, 프로토타입의 개수를 K라 할 때, 메모리 특성맵은 수학식 1과 같이 표현될 수 있다.

수학식 1

[0023]
$$M = \{m_{c,k}\}_{c,k=1}^{C,K}$$

[0024] 여기서, c는 각 메모리 특성에 대응하는 클래스를 나타내는 식별자로, 1과 C 사이의 값을 가질 수 있다. k는 동일한 클래스에 대응하는 메모리 특성들을 구별하기 위한 식별자로, 1과 K 사이의 값을 가질 수 있다. 컨텍스트 메모리 모듈(12)은 각 클래스를 프로토타입의 집합으로 표현함으로써, 클래스 내(intra-classes)에 다양한 속성들이 포함할 수 있도록 한다.

[0025] 단일 이미지 객체 검출기(10)는 각 프레임의 처리 과정에서 필요한 정보만 선택적으로 컨텍스트 메모리 모듈(12)에 저장하고, 현재 프레임에 유용한 정보만을 활용할 수 있다. 이에 따라, 단일 이미지 객체 검출기(10)는 각 프레임마다, 현재까지 관측된 전체 데이터셋에 대한 컨텍스트 정보를 효율적으로 활용할 수 있다.

[0026] 이하, 도 2 내지 도 5b를 참조하여, 본 개시의 일 실시예에 따른 단일 이미지 객체 검출기(10)의 동작을 설명하도록 한다.

[0027] 도 2는 본 개시의 일 실시예에 따른 단일 이미지 객체 검출기의 구조를 설명하기 위해 참조되는 도면이다.

[0028] 단일 이미지 객체 검출기(10)는 백본 네트워크(20), 인코더(21), 업데이트 모듈(22), 샘플링 모듈(23), 메모리-유도 디코더(Memory-Guided Decoder, 25), 분류 헤드(26) 및 회귀 헤드(27)를 전부 또는 일부 포함할 수 있다.

[0029] 백본 네트워크(20)는 현재 프레임으로부터 이미지 특성맵(200)을 추출한다. 백본 네트워크(20)로는, 사전-훈련된 합성곱 신경망(CNN, Convolutional Neural Network)이 이용될 수 있다. 각 특성들의 상대적인 공간 위치 정보(spatial positional information)를 인코더(21)에 제공하기 위해, 이미지 특성맵(200)에 포지셔널 인코딩(positional encoding)이 더해질 수 있다. 포지셔널 인코딩은, 사인 함수 및/또는 코사인 함수 등과 같이 고정된 함수를 통해 생성되거나, 학습 가능한 추가적인 임베딩 레이어(embedding layer)에 의해 생성될 수도 있다.

[0030] 인코더(21)는 백본 네트워크(20)에 의해 추출된 이미지 특성맵(200) 및 컨텍스트 메모리 모듈(12)에 저장된 메모리 특성맵(220)을 각각 전달받아, 인코딩된 이미지 특성맵(202) 및 인코딩된 메모리 특성맵(222)을 생성한다. 인코더(21)는 트랜스포머 인코더로 구현될 수 있다. 인코더(21)는 복수개의 인코더 블록을 포함할 수 있고, 각 인코더 블록은 멀티-헤드 셀프-어텐션(multi-head self-attention) 레이어를 포함할 수 있다. 인코더(21)는 이미지 특성맵(200) 및 메모리 특성맵(220)을 연결(concatenate)하고, 연결된 특성맵에 대해 복수개의 인코더 블록을 순차적으로 적용함으로써 인코딩된 이미지 특성맵(202) 및 인코딩된 메모리 특성맵(222)을 생성할 수 있다.

[0031] 대부분의 단일 이미지 기반 객체 검출기는, 현재 프레임에서 추출된 이미지 특성에서 공간 정보(spatial information)를 모으기 위해 트랜스포머 인코더를 사용한다. 트랜스포머 인코더의 셀프-어텐션 구조에 의해, 각 이미지 특성은 현재 프레임의 공간 컨텍스트 정보를 포함할 수 있게 된다. 그러나 백본 네트워크(20)가 현재 프레임에 대해 모호한 특성을 추출하는 경우(예컨대, 이미지 품질이 낮은 경우 또는 일부 영역이 가려진 경우)에는, 이미지 특성을 효과적으로 개선하는 데 어려움을 겪을 수 있다. 이러한 제한을 완화하기 위해, 본 개시에 따른 인코더(21)는 선행하는 전체 프레임들에서 얻은 고정 크기의 메모리 특성을 사용하여 각 단일 프레임의 이미지 특성을 향상시킨다. 이를 통해 다른 프레임의 데이터를 직접 사용할 필요 없이 현재 프레임의 이미지 특성을 개선할 수 있다.

[0032] 이미지 특성맵(200), 메모리 특성맵(220), 인코딩된 이미지 특성맵(202), 및 인코딩된 메모리 특성맵(222)을 각각 F, M, $\mathcal{F}$, 및 $\mathcal{M}$ 라 할때, 인코더(21)는 수학식 2와 같이 표현될 수 있다.

수학식 2

[0033] $$[\mathcal{F}, \mathcal{M}] = \mathrm{Enc}([F, M]), \qquad \text{where } F \in \mathbb{R}^{H \cdot W \times d} \text{ and } M \in \mathbb{R}^{C \cdot K \times d}$$

[0034] 여기서, [·,·]는 두 특성 맵 간의 연결을 나타내고, H 및 W는 이미지 특성맵(200)의 높이 및 너비를 나타내고, C는 단일 이미지 객체 검출기(10)에서 예측하고자 하는 클래스의 개수를 나타내며, K는 클래스별로 컨텍스트 메모리 모듈(12)에 저장되는 특성들(즉, 프로토타입)의 개수를 나타내며, d는 특성의 차원(dimension)을 나타낸다. 수학식 1에 나타나듯이, 시퀀스를 입력으로 받는 인코더(21)에서 이미지 특성맵(200)을 처리할 수 있도록, 이미지 특성맵(200)의 공간 차원들이 하나의 차원으로 플래튼(flatten) 될 수 있다. 마찬가지로, 메모리 특성맵(220)에서, 클래스 차원 및 프로토타입 차원이 하나의 차원으로 플래튼될 수 있다.

[0035] 인코더(21)에서는 이미지 특성맵(200)에 내재된 현재 프레임에 대한 정보와 메모리 특성맵(220)에 내재된 이전 프레임들에서의 시공간적 컨텍스트 정보(spatio-temporal contextual information)가 집계(aggregate)된다. 따라서 인코딩된 이미지 특성맵(202)은 메모리 특성맵(220)으로부터 풍부한 컨텍스트 정보를 얻게 되며, 동시에 인코딩된 메모리 특성맵(222)은 현재 프레임에 맞는 클래스 정보를 획득하게 된다.

[0036] 인코딩된 메모리 특성맵(222)은 현재 프레임에 대한 분류 스코어를 얻기 위해 샘플링 모듈(23)로 전달된다. 인코딩된 이미지 특성맵(202)은 객체 검출을 위해 메모리-유도 디코더(25)로 전달되며, 메모리 업데이트를 위해 업데이트 모듈(22)로도 전달된다.

[0037] 업데이트 모듈(22)은 인코딩된 이미지 특성맵(202)을 기초로, 컨텍스트 메모리 모듈(12)에 저장된 메모리 특성들 중 적어도 일부를 업데이트한다. 메모리 특성의 업데이트로 방법으로는, 비-모수적 방식(non-parametric manner)인 모멘텀 업데이트(momentum update)가 적용될 수 있다. 업데이트 모듈(22)의 예시적인 실시형태는 도 3을 참조하여 후술하도록 한다.

[0038] 샘플링 모듈(23)은 컨텍스트 메모리 모듈(12)에서 현재 프레임의 처리에 필요한 정보를 샘플링한다. 샘플링 모듈(23)은 컨텍스트 메모리 모듈(12)에 저장된 메모리 특성들 중 현재 이미지와 관련된 정보를 추출하여, 샘플링된 메모리 특성맵(224)을 구성할 수 있다. 메모리에서 관련 정보를 신중하게 선택하는 것은 고품질 메모리를 만드는 것만큼 중요하다. 메모리에서 정보를 랜덤하게 샘플링하거나 현재 프레임과 인접한 프레임의 정보만을 이용하는 것은, 최적의 메모리 샘플링을 보장할 수 없다. 이러한 문제를 해결하기 위해, 샘플링 모듈(23)은 인코딩된 메모리 특성맵(222)으로부터 산출된 스코어에 기반하여 컨텍스트 메모리 모듈(12)에서 현재 프레임과 관련된 정보를 추출한다. 샘플링 모듈(23)의 예시적인 실시형태는 도 4를 참조하여 후술하도록 한다.

[0039] 메모리-유도 디코더(25)는 하나 이상의 디코더 블록(250-1 내지 250-L)을 포함한다. 각 디코더 블록(250-1 내지 250-L)에는, 고정된 개수의 학습 가능한(또는 학습된) 포지셔널 임베딩들인 객체 쿼리들(object queries, 240), 인코딩된 이미지 특성맵(202) 및 샘플링된 메모리 특성맵(224)이 전달된다. 디코더 블록(250-1 내지 250-L)은 샘플링된 시공간 메모리 정보를 활용하여, 입력된 객체 쿼리들의 시맨틱 정보(semantic information)를 강화할 수 있다. 디코더 블록(250-1 내지 250-L)의 예시적인 실시형태는 도 5a 및 도 5b를 참조하여 후술하도록 한다.

[0040] 분류 헤드(26) 및 회귀 헤드(27)는 마지막 디코더 블록(240-L)이 출력하는 최종 객체 쿼리들(242)을 이용하여, 미리 정의된 개수의 클래스 및 바운딩박스의 세트를 예측할 수 있다. 분류 헤드(26) 및 회귀 헤드(27)는 예컨대, 피드-포워드 네트워크(feed-forward network)로 구성될 수 있다.

[0041] 추론 과정의 경우, 예측된 클래스 및 박스의 세트들이 현재 프레임에 대한 객체 검출 결과로서 출력될 수 있다. 훈련 과정의 경우, 소정의 매칭 알고리즘에 따라 헤드들(26 및 27)에 의해 예측된 바운딩박스 및 클래스의 세트와 GT(Ground truth) 세트가 매칭될 수 있고, 매칭된 페어를 기반으로 산출된 손실을 기초로 단일 이미지 객체 검출기(10)의 학습 가능한 파라미터들이 업데이트될 수 있다. 매칭 알고리즘 및 손실로는, 예컨대, DETR에서 사용되는 헝가리 매칭(Hungarian matching) 및 헝가리 손실(Hungarian loss)이 이용될 수 있으나 이러한 예시에 한정되는 것은 아니다.

[0042] 도 3은 본 개시의 일 실시예에 따른 업데이트 모듈의 동작을 설명하기 위해 참조되는 도면이다.

[0043] 업데이트 모듈(22)은 인코딩된 이미지 특성맵(202)에서 하나 이상의 인스턴스 특성들(300)을 추출한다. 훈련 과정에서, 인스턴스 특성들(300)은 GT 바운딩박스에서 추출되고, 추론 과정에서는 예측된 바운딩박스에서 추출될 수 있다. 업데이트 모듈(22)은 RoI 정렬(Region of Interest Align)을 통해 GT 바운딩박스 또는 예측된 바운딩박스를 축소된 특성맵의 크기에 맞게 조정하고, 이로부터 인스턴스 특성들(300)을 추출할 수 있다. 업데이트 모듈(22)은 GT 클래스 또는 단일 이미지 객체 검출기(10)에 의해 예측된 클래스를 기초로, 각 인스턴스 특성에 대응하는 클래스를 식별할 수 있다. 인코딩된 이미지 특성맵(202)에서 N 개의 인스턴스 특성들(300)을 추출한다고 가정할 때, 인스턴스 특성들(300)은 수학식 3과 같이 표현될 수 있다.

수학식 3

[0044] $$\tilde{\mathcal{F}} = \{f_n^c\}_{n=1}^N$$

[0045] 여기서, c는 각 인스턴스 특성에 대응하는 클래스의 식별자를 나타내며, 예컨대, 1과 C 사이의 값을 가질 수 있다.

[0046] 업데이트 모듈(22)은 각 인스턴스 특성과 동일한 클래스에 대응하는 K 개의 메모리 특성들 중에서, 해당 인스턴스 특성과의 상관관계(correlation)가 가장 큰 메모리 특성을 업데이트할 수 있다. 예컨대, 컨텍스트 메모리 모듈(12)의 업데이트 과정은 수학식 4와 같이 표현될 수 있다.

수학식 4

[0047]

$$m_{c,k_n} = \alpha \cdot m_{c,k_n} + (1-\alpha) \cdot f_n^c,$$

$$\text{where } k_n = \text{argmax}_{\text{k}}\{\langle f_n^c, m_{c,k}\rangle\}_{k=1}^{K}$$

[0048] 여기서, <·,·>는 두 특성 간의 상관관계로 정의되며, α는 모멘텀 계수(momentum coefficient)로 0 이상 1 이하의 값으로 설정될 수 있다.

[0049] 인코딩된 이미지 특성맵(202)에는 메모리 특성맵(220)으로부터 집계된 풍부한 컨텍스트 정보가 포함된다. 따라서, 비디오의 복수개의 프레임들의 순차적인 처리과정에서 인코딩된 이미지 특성맵(202)에 의해 반복적으로 업데이트되는 메모리 특성맵(220)은, 이전에 관찰된 전체 데이터셋에 대한 컨텍스트 정보를 포함하게 된다. 또한, 메모리 특성맵(220)은 각 클래스에 대해 여러 프로토타입(multi prototype)을 가짐에 따라, 전체 데이터셋에 나타나는 인스턴스 특성들의 다양한 분포를 수용할 수 있다.

[0050] 한편, 컨텍스트 메모리 모듈(12)은 인코더(21)의 출력 측의 임베딩 공간(embedding space) 내의 특징들을 저장한다. 이를 이미지 특성맵(200)과 동일한 임베딩 공간으로 전사시키기 위해, 메모리 특성맵(220)이 인코더(21)로 입력되기에 앞서 메모리 임베딩(memory embedding)이 적용될 수 있다. 메모리 임베딩은, 예컨대, 얕은 다층 퍼셉트론(MLP, multi-layer perceptron)에 의해 수행될 수 있다.

[0051] 도 4는 본 개시의 일 실시예에 따른 샘플링 모듈의 동작을 설명하기 위해 참조되는 도면이다.

[0052] 샘플링 모듈(23)의 동작은, 분류 과정 및 다중 임계치 샘플링 과정을 포함할 수 있다.

[0053] 분류 과정에서, 샘플링 모듈(23)은 현재 프레임의 클래스 정보를 포함하는 인코딩된 메모리 특성맵(222)으로부터, 각각의 인코딩된 메모리 특성에 대응하는 분류 스코어(classification score)를 획득한다. 본 개시에서, 분류 스코어는 확신도(confidence)로 지칭될 수도 있다.

[0054] 샘플링 모듈(23)은 각 인코딩된 메모리 특성을 해당 인코딩된 메모리 특성에 대해 독립적으로 구성된 분류 헤드(classification head, 400)에 적용하여, 분류 스코어를 획득할 수 있다. 분류 헤드(400)는 예컨대, 피드-포워드 네트워크(feed-forward network)로 구성될 수 있다. 분류 스코어를 0과 1 사이의 값으로 표현하기 위해, 시그모이드 함수(sigmoid function, 410)가 분류 헤드(400)에 후속될 수 있다. 이때, 인코딩된 메모리 특성맵(222)은 수학식 5와 같이 인코딩된 메모리 특성들의 집합으로 표현될 수 있으며, 이때 c번째 클래스의 k번째 인코딩된 메모리 특성에 대한 분류 스코어는 수학식 6과 같이 계산될 수 있다.

수학식 5

[0055]

$$\mathcal{M} = \{\mathrm{m}_{c,k}\}_{c,k=1}^{C,K}$$

수학식 6

[0056]

$$p_{c,k} = \text{Sigmoid}\left(\text{FFN}_{c,k}(\mathrm{m}_{c,k})\right)$$

[0057] 여기서, $\text{FFN}_{c,k}$는 c번째 클래스의 k번째 인코딩된 메모리 특성에 대응하는 분류 헤드(400)를 나타내고, Sigmoid(·)는 시그모이드 함수를 나타낸다.

[0058] 다중 임계치 샘플링 과정에서, 샘플링 모듈(23)은 메모리 특성맵(220)을 구성하는 메모리 특성들 각각에 대해, 동일한 클래스-프로토타입 위치에 대응하는 분류 스코어 및 서로 다른 값을 갖는 다중 임계치(440)에 의해 임계화된 메모리 특성(thresholded memory feature)의 조합을 획득한다. 예컨대, c번째 클래스의 k번째 메모리 특성

$m_{c,k}$에 대응하는, t번째 임계치 τ_t에 의해 임계화된 메모리 특성은 수학식 7과 같이 계산될 수 있다.

수학식 7

[0059]
$$\tilde{m}_{c,k}^{t} = s_{c,k}^{t} m_{c,k} + (1 - s_{c,k}^{t})\emptyset,$$
$$\text{where } s_{c,k}^{t} = \delta(p_{c,k} > \tau_t)$$

[0060] 여기서, Φ는 학습 가능한(learnable) 임베딩(추론 과정의 경우, 미리 학습된 임베딩)인, 클래스-없음(no-class) 임베딩을 나타낸다. δ(·)는 조건이 참일 때 1을 출력하고, 그렇지 않으면 0을 출력한다. 즉, 샘플링 인덱스 $s_{c,k}^{t}$ 는, c번째 클래스의 k번째 인코딩된 메모리 특성에 대한 분류 스코어 $p_{c,k}$를 t번째 임계치 τ_t로 이진화한 값이다.

[0061] 샘플링 모듈(23)은 프로젝션 레이어(460)를 이용하여 임계화된 메모리 특성들의 조합을 임베딩함으로써, 각 메모리 특성에 대응하는 샘플링된 메모리 특성을 획득한다. 예컨대, 샘플링된 메모리 특성맵(224)은 수학식 8과 같이 각 메모리 특성에 대응하는 샘플링된 메모리 특성의 집합으로 표현될 수 있으며, c번째 클래스의 k번째 메모리 특성에 대응하는 샘플링된 메모리 특성은 수학식 9와 같이 계산될 수 있다.

수학식 8

[0062]
$$\tilde{M} = \{\tilde{m}_{c,k}\}_{c,k=1}^{C,K}$$

수학식 9

[0063]
$$\tilde{m}_{c,k} = \mathrm{Proj}(\tilde{m}_{c,k}^{1}, \tilde{m}_{c,k}^{2}, \cdots \tilde{m}_{c,k}^{T})$$

[0064] 여기서, T는 임계치 및/또는 샘플링 인덱스의 개수를 나타내고, Proj(·)는 프로젝션 레이어(460)를 나타낸다. 프로젝션 레이어(460)는 다양한 확신도를 가진 다중 임계화된 메모리 특성들(multi-thresholded memory features)을 결합하여 샘플링된 메모리 특성을 생성한다.

[0065] 훈련 과정에는 분류 헤드(400)를 훈련시키고, 인코더(21)의 클래스 구별 능력을 향상시키기 위해, 비대칭 손실 함수(Asymmetric Loss, ASL)가 추가적으로 사용될 수 있다.

[0066] 도 5a는 본 개시의 일 실시예에 따른 디코더 블록의 구조를 설명하기 위해 참조되는 도면이다.

[0067] 도 5a에 도시되듯이 디코더 블록(250)은 셀프-어텐션 레이어(self-attention layer, 500), 크로스-어텐션 레이어(cross-attention layer, 520) 및 메모리 크로스-어텐션 레이어(memory cross-attention layer, 540)를 포함할 수 있다. 각 어텐션 레이어는 복수개의 어텐션 헤드(attention head)들을 포함하는 멀티-헤드 어텐션 레이어로 구성될 수 있다. 한편, 도 5a에는 도시되지 않았으나, 어텐션 레이어들(500, 520 및 540) 중 적어도 일부에 잔차 연결 및 정규화 레이어가 후속될 수 있다.

[0068] 디코더 블록(250)은 선행하는 디코더 블록(미도시)이 출력하는 객체 쿼리들(502)을 셀프-어텐션 레이어(500) 및 크로스-어텐션 레이어(520)에 순차적으로 적용하여, 현재 프레임에 대한 시맨틱 정보를 포함하는 객체 쿼리를 획득할 수 있다. 디코더 블록(250)은 크로스-어텐션 레이어(520)가 출력하는 객체 쿼리를 메모리 크로스-어텐션 레이어(540)에 적용하여, 현재 프레임과 연관된 클래스 정보(class information)를 강화할 수 있다.

[0069] 셀프-어텐션 레이어(500)는 객체 쿼리들(502)에 대한 셀프 어텐션을 계산한다. 1번째 디코더 블록의 셀프-어텐션 레이어(500)가 출력하는 객체 쿼리는 수학식 10과 같이 표현될 수 있다.

수학식 10

[0070] $$\mathcal{O}_{sa}^{l} = \text{Self}-\text{Attn}\left(Q = \mathcal{O}^{l-1}, K = \mathcal{O}^{l-1}, V = \mathcal{O}^{l-1}\right)$$

[0071] 여기서, $\mathcal{O}^{l-1}$ 는 l-1번째 디코더 블록이 출력하는 객체 쿼리(또는 초기 객체 쿼리)를 나타낸다.

[0072] 크로스-어텐션 레이어(520)는 셀프-어텐션 레이어(500)가 출력하는 객체 쿼리들과 인코딩된 이미지 특성맵(202) 사이의 크로스-어텐션을 계산한다. l번째 디코더 블록의 크로스-어텐션 레이어(520)가 출력하는 객체 쿼리는 수학식 11과 같이 표현될 수 있다.

수학식 11

[0073] $$\mathcal{O}_{ca}^{l} = \text{Cross}-\text{Attn}\left(Q = \mathcal{O}_{sa}^{l}, K = \mathcal{F}, V = \mathcal{F}\right)$$

[0074] 메모리 크로스-어텐션 레이어(540)는, 크로스-어텐션 레이어(520)가 출력하는 객체 쿼리들과 샘플링된 메모리 특성맵(224) 사이의 크로스-어텐션을 계산한다. l번째 디코더 블록의 메모리 크로스-어텐션 레이어(540)가 출력하는 객체 쿼리(504)는 수학식 12와 같이 표현될 수 있다.

수학식 12

[0075] $$\mathcal{O}_{mca}^{l} = \text{Mem.Cross}-\text{Attn}\left(Q = \mathcal{O}_{ca}^{l}, K = \widetilde{M}, V = \widetilde{M}\right)$$

[0076] 어텐션 레이어들(500, 520 및 540)에서 쿼리 임베딩들(query embeddings) 및 키 임베딩들(key embeddings)을 계산하기 위한 입력 토큰들에는 포지셔널 정보(positional information)가 더해질 수 있다. 각 어텐션 레이어(500, 520 및 540)에 제공되는 포지셔널 정보는 예컨대, 전체 디코더 블록들(250-1 내지 250-L) 사이에서 공유되는 학습 가능한(또는 학습된) 포지셔널 임베딩들인 객체 쿼리들(506), 이미지 특성맵(200)의 포지셔널 인코딩 및/또는 메모리 인코딩을 포함할 수 있다. 객체 쿼리들(506)과의 구별을 위해, 선행하는 디코더 블록 또는 선행하는 어텐션 레이어에서 출력하는 객체 쿼리들(502)은 디코더 임베딩들(decoder embeddings)로 지칭될 수 있다.

[0077] 일 예로, 셀프 어텐션 레이어(500)에서, 쿼리 임베딩들 및 키 임베딩들은 선행하는 디코더 블록이 출력하는 디코더 임베딩들(502)에 객체 쿼리들(506)의 합에 근거하여 산출될 수 있다. 다른 예로, 크로스 어텐션 레이어(520)에서, 쿼리 임베딩들은 셀프 어텐션 레이어(500)가 출력하는 디코더 임베딩들과 객체 쿼리들(506)의 합에 근거하여 산출되고, 키 임베딩들은 인코딩된 이미지 특성맵(202)과 포지셔널 인코딩의 합에 근거하여 산출될 수 있다. 또 다른 예로, 메모리 어텐션 레이어(540)에서, 쿼리 임베딩들은 크로스 어텐션 레이어(520)가 출력하는 디코더 임베딩들과 객체 쿼리들(506)의 합에 근거하여 산출되고, 키 임베딩들은 샘플링된 메모리 특성맵(224)과 메모리 인코딩의 합에 근거하여 산출될 수 있다. 여기서, 메모리 인코딩은 이미지 특성맵(200)의 포지셔널 인코딩과 동일 또는 유사한 방식을 통해 생성될 수 있으며, 입력 토큰이 컨텍스트 메모리 모듈(12)에서 추출된 특성들임을 알려주는 기능을 할 수 있다.

[0078] 도 5b는 본 개시의 다른 실시예에 따른 디코더 블록의 구조를 설명하기 위해 참조되는 도면이다.

[0079] 도 5b를 참조하면, 본 개시의 다른 실시예에 따른 디코더 블록(250)은 학습 가능한(또는 학습된) 앵커 박스들(anchor boxes, 512)로부터 추출된 포지셔널 정보를 각 어텐션 레이어들(500, 530 및 540)에 제공할 수 있다. 본 실시예에서는, 앵커 박스들(512)의 크기에 맞게 크로스-어텐션 맵의 크기를 조정하기 위해, 크로스 어텐션 레이어(520)가 너비-및-높이-변조된 크로스-어텐션 레이어(width&height-modulated cross-attention layer, 530)로 대체될 수 있다. 너비-및-높이-변조된 크로스-어텐션 레이어(530) 및 메모리 크로스-어텐션 레이어(540)에서, 쿼리 임베딩들 및 키 임베딩들을 계산하기 위한 입력 토큰들에는 포지셔널 정보가 연결(concatenate)될 수 있다.

[0080] 본 실시예에서, 앵커 박스들(512)은 디코더 블록(250) 단위로 갱신될 수 있다. 예컨대, 각 디코더 블록(250)에서, 너비-및-높이-변조된 크로스-어텐션 레이어(530)의 출력을 기초로 앵커 박스들의 위치 및 크기 변화량(514)이 예측될 수 있다. 예측된 변화량(514)을 기초로 갱신된 신규 앵커 박스들(516)은 후속하는 디코더 블록(미도시)으로 전달될 수 있다.

[0081] 도 6은 본 개시의 일 실시예에 따른 객체 검출방법을 나타내는 흐름도이다.

[0082] 도 6에 도시된 방법은, 전술한 비디오 객체 검출장치(1)의 하나 이상의 구성요소의 기능이 적어도 하나의 컴퓨팅 장치에 의해 실행됨으로써 구현될 수 있다. 따라서 이하의 설명은 컴퓨팅 장치가 수행하는 동작 측면에서 서술한다.

[0083] 컴퓨팅 장치는 현재 프레임으로부터 이미지 특성맵을 추출한다(S600). 컴퓨팅 장치는, 현재 프레임을 미리 훈련된 백본 네트워크에 적용하여, 이미지 특성맵을 획득할 수 있다. 컴퓨팅 장치는 이미지 특성맵에 포지셔널 인코딩을 적용하고, 차원 수를 조정할 수 있다.

[0084] 컴퓨팅 장치는 이미지 특성맵 및 미리 저장된 메모리 특성맵을 인코딩한다(S620). 컴퓨팅 장치는 컨텍스트 메모리 모듈에서 메모리 특성맵을 읽어들일 수 있다. 메모리 특성맵은, 예측 가능한 하나 이상의 클래스들 각각에 대해 미리 정의된 개수의 메모리 특성들을 포함할 수 있다. 컴퓨팅 장치는 이미지 특성맵 및 메모리 특성맵을 연결(concatenate)하고, 연결된 특성맵에 대한 셀프-어텐션을 기초로 인코딩된 이미지 특성맵 및 인코딩된 메모리 특성맵을 획득할 수 있다. 인코딩된 이미지 특성맵에는 현재 프레임 이전에 관측된 전체 프레임들의 시공간 컨텍스트가 내재될 수 있다. 인코딩된 메모리 특성맵에는 현재 프레임 내 객체의 클래스에 관한 정보가 내재될 수 있다. 인코딩된 메모리 특성맵은 각 메모리 특성에 대응하는 인코딩된 메모리 특성을 각각 포함할 수 있다. 특정 메모리 특성에 대응하는 인코딩된 메모리 특성은, 해당 메모리 특성과 동일한 클래스-프로토타입 위치를 가질 수 있다.

[0085] 컴퓨팅 장치는 인코딩된 메모리 특성맵으로부터 산출된 분류 스코어들 및 메모리 특성맵을 기초로, 샘플링된 메모리 특성맵을 생성한다(S640). 샘플링된 메모리 특성맵은 각 메모리 특성에 대응하는 샘플링된 메모리 특성을 각각 포함할 수 있다. 특정 메모리 특성에 대응하는 샘플링된 메모리 특성은, 해당 메모리 특성과 동일한 클래스-프로토타입 위치를 가질 수 있다. 컴퓨팅 장치는 각 인코딩된 메모리 특성에 대해 분류 스코어를 산출할 수 있다. 컴퓨팅 장치는 기정의된 하나 이상의 임계치들에 기초하여 각 인코딩된 메모리 특성에 대응하는 메모리 특성을 임계화할 수 있다. 예컨대, 컴퓨팅 장치는 기정의된 하나 이상의 임계치들을 기초로 특정 클래스-프로토타입 위치에 대응하는 분류 스코어를 이진화하여 하나 이상의 샘플링 인덱스들을 획득하고, 각 샘플링 인덱스의 값을 기초로 해당 클래스-프로토타입 위치의 메모리 특성 및 클래스-없음 임베딩 중 하나를 임계화된 메모리 특성으로 선택할 수 있다. 컴퓨팅 장치는 샘플링 인덱스별로 선택된 하나 이상의 임계화된 메모리 특성들을 임베딩하여, 각 메모리 특성에 대응하는 샘플링된 메모리 특성을 생성할 수 있다.

[0086] 컴퓨팅 장치는 인코딩된 이미지 특성맵 및 샘플링된 메모리 특성맵을 기초로, 현재 프레임 내의 객체의 바운딩박스 및 클래스를 예측한다(S660). 컴퓨팅 장치는 객체 쿼리를 메모리-유도 디코더에 적용하여, 현재 프레임에 대한 시맨틱 정보 및 현재 프레임과 연관된 클래스에 관한 정보가 향상된 출력 객체 쿼리를 획득하고, 출력 객체 쿼리로부터 객체의 바운딩박스 및 클래스를 예측할 수 있다.

[0087] 메모리-유도 디코더는, 하나 이상의 디코더 블록을 포함할 수 있다. 각 디코더 블록은, 입력된 객체 쿼리에 대한 셀프-어텐션을 기초로 제1 객체 쿼리를 생성하는 제1 어텐션 레이어, 제1 객체 쿼리와 인코딩된 이미지 특성맵 간의 크로스-어텐션을 기초로, 제2 객체 쿼리를 생성하는 제2 어텐션 레이어, 및 제2 객체 쿼리와 샘플링된 메모리 특성맵 간의 크로스-어텐션을 기초로, 제3 객체 쿼리를 생성하는 제3 어텐션 레이어를 포함할 수 있다. 컴퓨팅 장치는 각 샘플링된 메모리 특성에 대응하는 메모리 특성들의 위치(예컨대, 클래스-프로토타입 위치)를 나타내는 메모리 임베딩들을 샘플링된 메모리 특성맵에 결합하여 제3 어텐션 레이어에 제공할 수 있다.

[0088] 컴퓨팅 장치는 인코딩된 이미지 특성맵을 기초로, 메모리 특성맵을 업데이트한다(S680). 컴퓨팅 장치는 예측된 바운딩박스 또는 미리 지정된 GT(ground truth) 바운딩박스를 기초로, 인코딩된 이미지 특성맵에서 하나 이상의 인스턴스 특성들을 추출하고, 메모리 특성맵에서 추출된 인스턴스 특성들에 대응하는 하나 이상의 타겟 메모리 특성들을 업데이트할 수 있다. 여기서, 각 타겟 메모리 특성은, 각 인스턴스 특성과 동일한 클래스에 대응하는 메모리 특성들 중에서 선별된 것일 수 있다. 예컨대, 특정 인스턴스 특성과 동일한 클래스에 대응하는 메모리 특성들 중, 해당 인스턴스 특성과 가장 큰 상관관계(correlation)를 갖는 메모리 특성이 해당 인스턴스 특성에 대응하는 타겟 메모리 특성으로 선별될 수 있다. 특정 인스턴스 특성에 대응하는 타겟 메모리 특성은, 해당 인

스턴스 특성 및 해당 타겟 메모리 특성 간의 선형 보간에 기초하여 업데이트될 수 있다.

[0089] 도 7은 본 개시에서 설명된 장치 및 방법들을 구현하기 위해 사용될 수 있는 예시적인 컴퓨팅 장치를 개략적으로 나타낸 블록구성도이다.

[0090] 컴퓨팅 장치(70)는 메모리(700), 프로세서(720), 스토리지(740), 입출력 인터페이스(760) 및 통신 인터페이스(780) 중 일부 또는 전부를 포함할 수 있다. 컴퓨팅 장치(70)는 비디오 객체 검출장치(1)의 적어도 일부를 구조적 및/또는 기능적으로 포함할 수 있다. 컴퓨팅 장치(70)는 데스크탑 컴퓨터, 서버 등과 같은 고정형(stationary) 컴퓨팅 장치뿐만 아니라, 랩탑 컴퓨터, 스마트 폰 등과 같은 휴대용(mobile) 컴퓨팅 장치일 수도 있다. 컴퓨팅 장치(70)는 인공지능 모델에 대한 연산들을 효율적인 방식으로 처리하는 것이 가능한 임의의 특수화된 하드웨어 가속기(accelerator)로 구현될 수도 있다. 예컨대, 컴퓨팅 장치(70)는 그래픽 처리 장치(graphic processing unit, GPU), 텐서 처리 장치(Tensor Processing Unit, TPU) 또는 신경망 처리 장치(neural processing unit, NPU) 등으로 구현될 수 있다.

[0091] 메모리(700)는 프로세서(720)로 하여금 본 개시의 다양한 실시예에 따른 방법 또는 동작을 수행하도록 하는 프로그램을 저장할 수 있다. 예를 들면, 프로그램은 프로세서(720)에 의해서 실행 가능한(executable) 복수의 명령어들을 포함할 수 있고, 복수의 명령어들이 프로세서(720)에 의해서 실행됨으로써 도 6에 도시된 방법이 수행될 수 있다. 메모리(700)는 단일 메모리 또는 복수의 메모리들일 수 있다. 이 경우, 본 개시의 다양한 실시예에 따른 방법 또는 동작을 수행하기 위해 필요한 정보는 단일 메모리에 저장되거나 복수의 메모리들에 나뉘어 저장될 수 있다. 메모리(700)가 복수의 메모리들로 구성된 경우, 복수의 메모리들은 물리적으로 분리될 수 있다. 메모리(700)는 휘발성 메모리 및 비휘발성 메모리 중 적어도 하나를 포함할 수 있다. 휘발성 메모리는 SRAM(Static Random Access Memory) 또는 DRAM(Dynamic Random Access Memory) 등을 포함하고, 비휘발성 메모리는 플래시 메모리(flash memory) 등을 포함한다.

[0092] 프로세서(720)는 적어도 하나의 명령어들을 실행할 수 있는 적어도 하나의 코어를 포함할 수 있다. 프로세서(720)는 메모리(700)에 저장된 명령어들을 실행할 수 있다. 프로세서(720)는 단일 프로세서 또는 복수의 프로세서들일 수 있다.

[0093] 스토리지(740)는 컴퓨팅 장치(70)에 공급되는 전력이 차단되더라도 저장된 데이터를 유지한다. 예를 들면, 스토리지(740)는 비휘발성 메모리를 포함할 수도 있고, 자기 테이프, 광학 디스크, 자기 디스크와 같은 저장 매체를 포함할 수도 있다.

[0094] 스토리지(740)에 저장된 프로그램은 프로세서(720)에 의해서 실행되기 이전에 메모리(700)로 로딩될 수 있다. 스토리지(740)는 프로그램 언어로 작성된 파일을 저장할 수 있고, 파일로부터 컴파일러 등에 의해서 생성된 프로그램은 메모리(700)로 로딩될 수 있다. 스토리지(740)는 프로세서(720)에 의해서 처리될 데이터 및/또는 프로세서(720)에 의해서 처리된 데이터를 저장할 수 있다.

[0095] 입출력 인터페이스(760)는 터치인터페이스, 키보드, 마우스 등과 같은 입력 장치를 포함할 수 있고, 디스플레이 장치 또는 스피커 등과 같은 출력 장치를 포함할 수 있다. 사용자는 입출력 인터페이스(760)를 통해 프로세서(720)에 의한 프로그램의 실행을 트리거하고/거나 프로세서(720)의 처리 결과를 확인할 수 있다.

[0096] 통신 인터페이스(780)는 외부 네트워크에 대한 액세스를 제공할 수 있다. 예를 들면, 컴퓨팅 장치(70)는 통신 인터페이스(780)를 통해 다른 장치들(예컨대, 카메라)과 통신할 수 있다.

[0097] 본 발명에 따른 장치 또는 방법의 각 구성요소는 하드웨어 또는 소프트웨어로 구현되거나, 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다. 또한, 각 구성요소의 기능이 소프트웨어로 구현되고 마이크로프로세서가 각 구성요소에 대응하는 소프트웨어의 기능을 실행하도록 구현될 수도 있다.

[0098] 본 명세서에 설명되는 시스템들 및 기법들의 다양한 구현예들은, 디지털 전자 회로, 집적회로, FPGA(field programmable gate array), ASIC(application specific integrated circuit), 컴퓨터 하드웨어, 펌웨어, 소프트웨어, 및/또는 이들의 조합으로 실현될 수 있다. 이러한 다양한 구현예들은 프로그래밍가능 시스템 상에서 실행 가능한 하나 이상의 컴퓨터 프로그램들로 구현되는 것을 포함할 수 있다. 프로그래밍가능 시스템은, 저장 시스템, 적어도 하나의 입력 디바이스, 그리고 적어도 하나의 출력 디바이스로부터 데이터 및 명령들을 수신하고 이들에게 데이터 및 명령들을 전송하도록 결합되는 적어도 하나의 프로그래밍가능 프로세서(이것은 특수 목적 프로세서일 수 있거나 혹은 범용 프로세서일 수 있음)를 포함한다. 컴퓨터 프로그램들(이것은 또한 프로그램들, 소프트웨어, 소프트웨어 애플리케이션들 혹은 코드로서 알려져 있음)은 프로그래밍가능 프로세서에 대한 명령어

들을 포함하며 "컴퓨터가 읽을 수 있는 기록매체"에 저장된다.

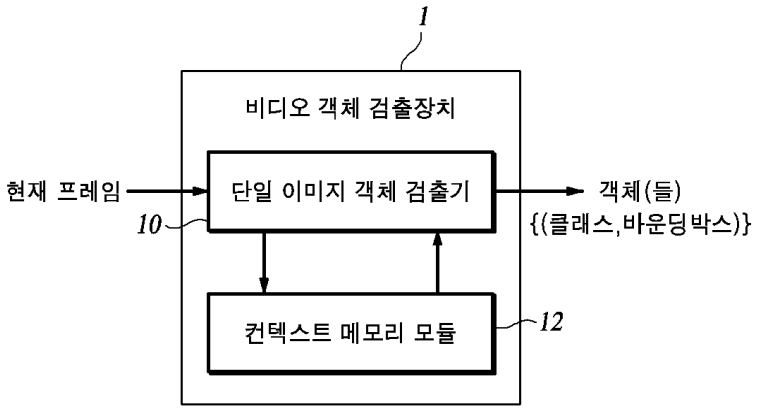
[0099] 컴퓨터가 읽을 수 있는 기록매체는, 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 이러한 컴퓨터가 읽을 수 있는 기록매체는 ROM, CD-ROM, 자기 테이프, 플로피디스크, 메모리 카드, 하드 디스크, 광자기 디스크, 스토리지 디바이스 등의 비휘발성(non-volatile) 또는 비일시적인(non-transitory) 매체일 수 있으며, 또한 데이터 전송 매체(data transmission medium)와 같은 일시적인(transitory) 매체를 더 포함할 수도 있다. 또한, 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수도 있다.

[0100] 본 명세서의 흐름도/타이밍도에서는 각 과정들을 순차적으로 실행하는 것으로 기재하고 있으나, 이는 본 개시의 일 실시예의 기술 사상을 예시적으로 설명한 것에 불과한 것이다. 다시 말해, 본 개시의 일 실시예가 속하는 기술 분야에서 통상의 지식을 가진 자라면 본 개시의 일 실시예의 본질적인 특성에서 벗어나지 않는 범위에서 흐름도/타이밍도에 기재된 순서를 변경하여 실행하거나 각 과정들 중 하나 이상의 과정을 병렬적으로 실행하는 것으로 다양하게 수정 및 변형하여 적용 가능할 것이므로, 흐름도/타이밍도는 시계열적인 순서로 한정되는 것은 아니다.

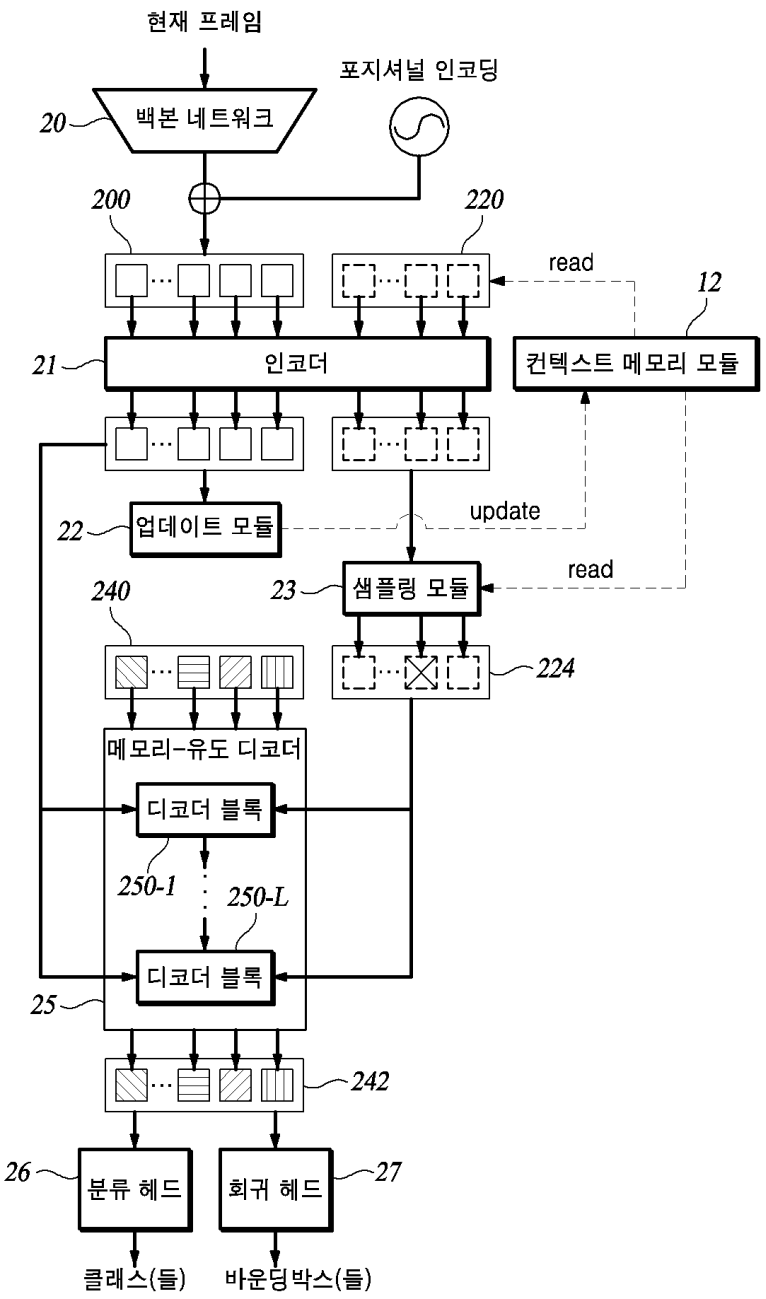
[0101] 이상의 설명은 본 실시예의 기술 사상을 예시적으로 설명한 것에 불과한 것으로서, 본 실시예가 속하는 기술 분야에서 통상의 지식을 가진 자라면 본 실시예의 본질적인 특성에서 벗어나지 않는 범위에서 다양한 수정 및 변형이 가능할 것이다. 따라서, 본 실시예들은 본 실시예의 기술 사상을 한정하기 위한 것이 아니라 설명하기 위한 것이고, 이러한 실시예에 의하여 본 실시예의 기술 사상의 범위가 한정되는 것은 아니다. 본 실시예의 보호범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 실시예의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

도면

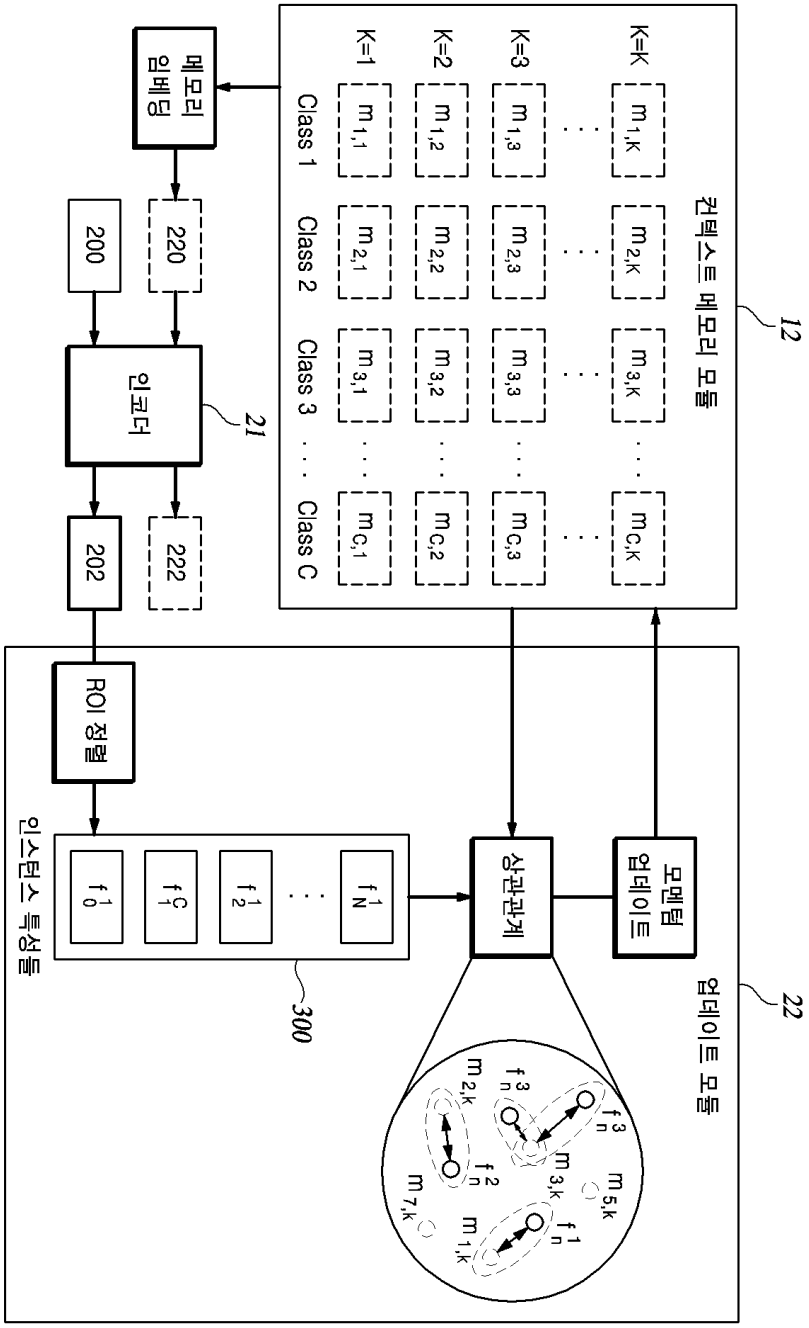
도면1



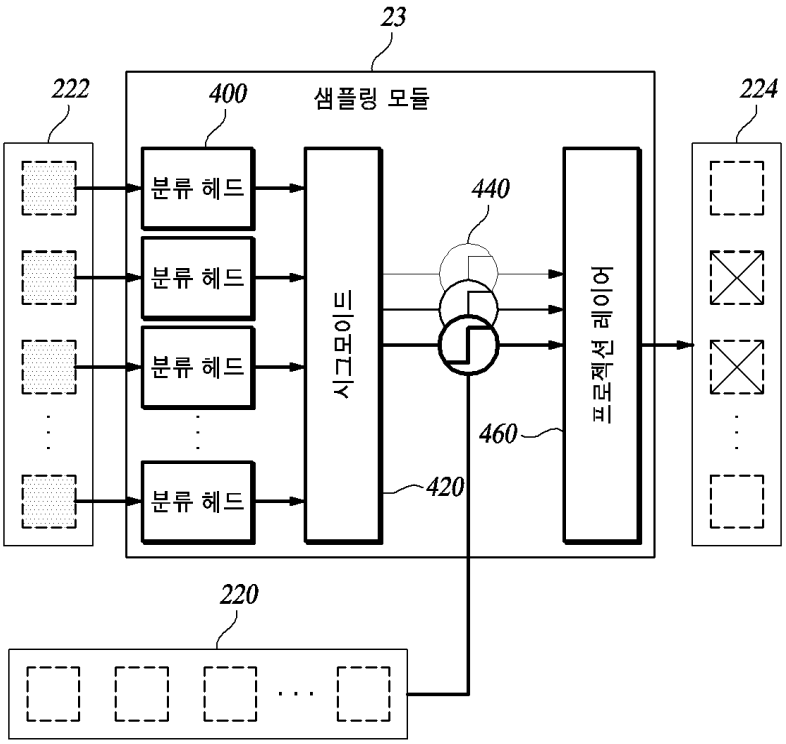
도면2

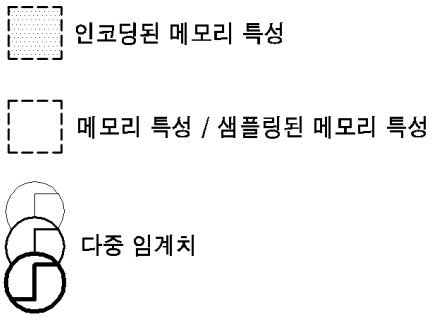

현재 프레임
포지셔널 인코딩
20
백본 네트워크
200
220
read
12
21
인코더
컨텍스트 메모리 모듈
22
업데이트 모듈
update
read
240
23
샘플링 모듈
224
메모리-유도 디코더
디코더 블록
250-1
250-L
디코더 블록
25
242
26
분류 헤드
회귀 헤드
27
클래스(들)
바운딩박스(들)

도면3

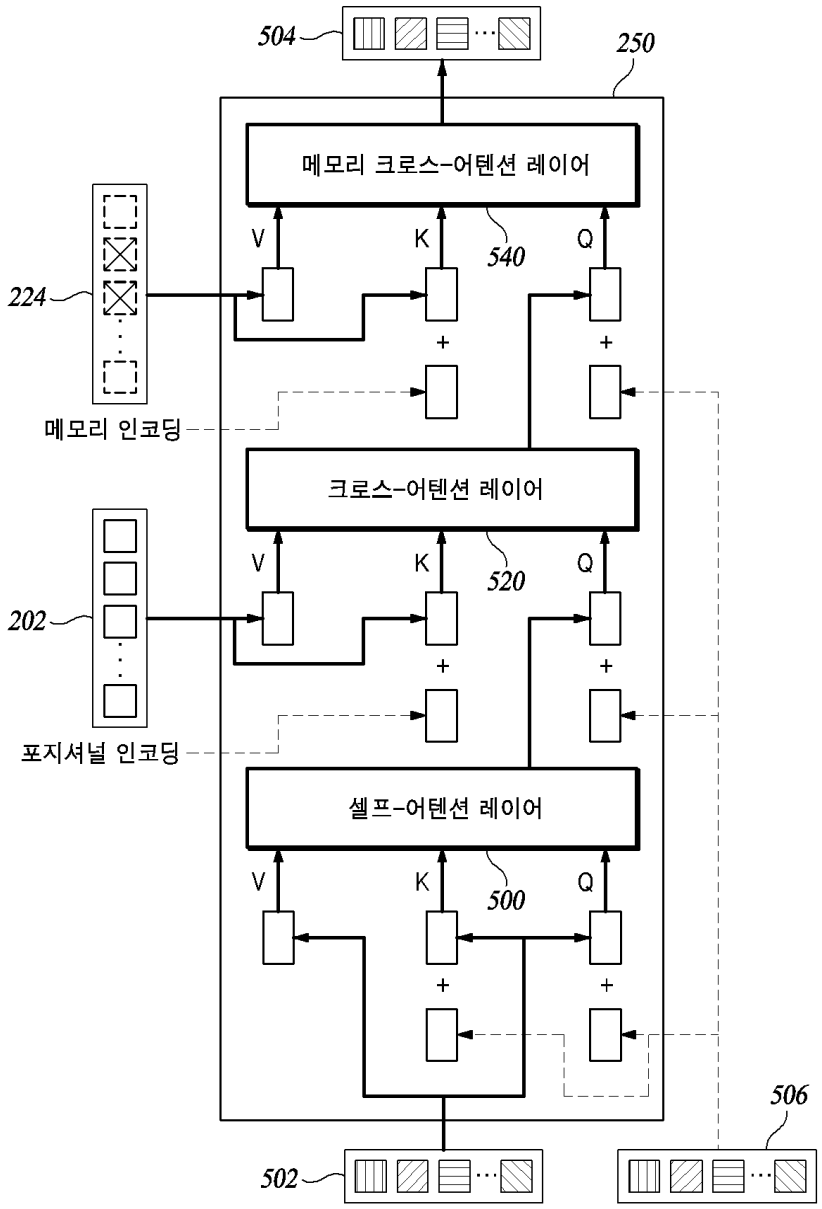

컨텍스트 메모리 모듈
12
K=K
K=3
K=2
K=1
Class 1
Class 2
Class 3
Class C
메모리 임베딩
220
222
200
인코더
21
202
ROI 정렬
인스턴스 특성들
300
상관관계
모멘텀 업데이트
업데이트 모듈
22

도면4

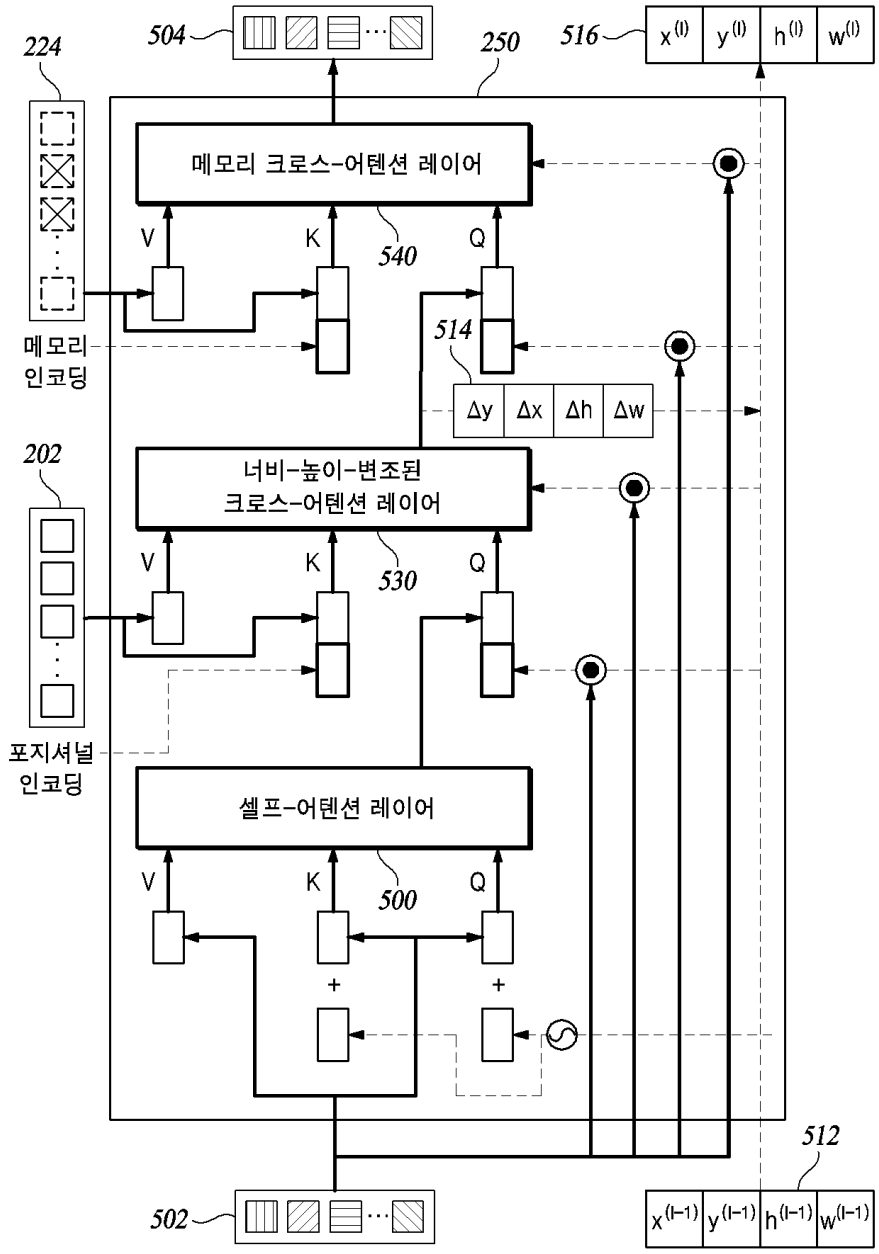

23
222
400
샘플링 모듈
224
분류 헤드
분류 헤드
분류 헤드
분류 헤드
시그모이드
440
프로젝션 레이어
460
420
220


인코딩된 메모리 특성
메모리 특성 / 샘플링된 메모리 특성
다중 임계치

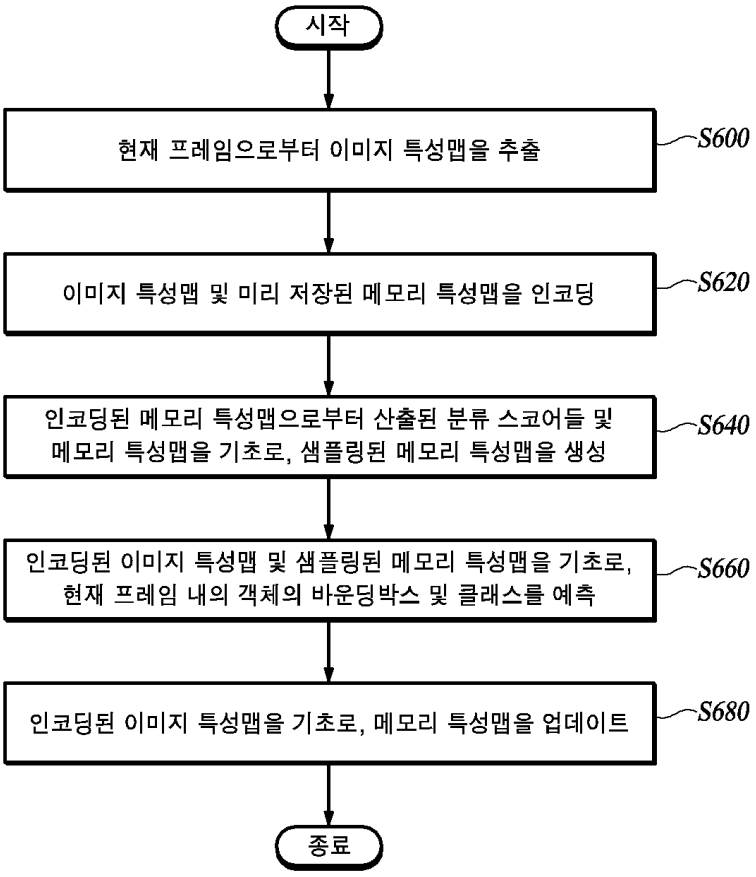
도면5a


504
250
메모리 크로스-어텐션 레이어
V
K
Q
540
224
+
+
메모리 인코딩
크로스-어텐션 레이어
V
K
Q
520
202
+
+
포지셔널 인코딩
셀프-어텐션 레이어
V
K
Q
500
+
+
506
502

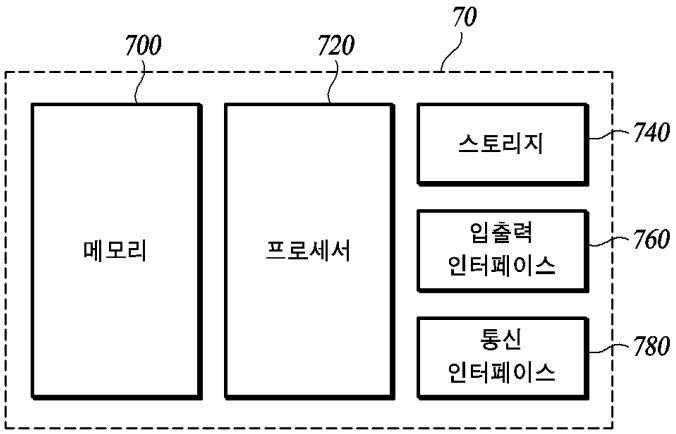
도면5b


224
504
250
516
x(l)
y(l)
h(l)
w(l)
메모리 크로스-어텐션 레이어
V
K
Q
540
514
메모리
인코딩
Δy
Δx
Δh
Δw
202
너비-높이-변조된
크로스-어텐션 레이어
530
포지셔널
인코딩
셀프-어텐션 레이어
500
+
512
502
x(l-1)
y(l-1)
h(l-1)
w(l-1)

도면6


시작
현재 프레임으로부터 이미지 특성맵을 추출
S600
이미지 특성맵 및 미리 저장된 메모리 특성맵을 인코딩
S620
인코딩된 메모리 특성맵으로부터 산출된 분류 스코어들 및 메모리 특성맵을 기초로, 샘플링된 메모리 특성맵을 생성
S640
인코딩된 이미지 특성맵 및 샘플링된 메모리 특성맵을 기초로, 현재 프레임 내의 객체의 바운딩박스 및 클래스를 예측
S660
인코딩된 이미지 특성맵을 기초로, 메모리 특성맵을 업데이트
S680
종료

도면7


70
700
720
메모리
프로세서
스토리지
740
입출력 인터페이스
760
통신 인터페이스
780