추상적인.우리는 객체 감지를 직접 세트 예측 문제로 보는 새로운 방법을 제시합니다.우리의 접근 방식은 탐지 파이프 라인을 간소화하여 비 이용성 억제 절차 또는 작업에 대한 사전 지식을 명시 적으로 인코딩하는 앵커 생성과 같은 많은 손으로 디자인 된 구성 요소의 필요성을 효과적으로 제거합니다.Detection Transformer 또는 DETR이라고하는 새로운 프레임 워크의 주요 성분은 이분자 매칭을 통해 고유 한 예측을 강요하는 세트 기반의 글로벌 손실과 변압기 인코더 디코더 아키텍처입니다.고정 된 작은 객체 쿼리 세트가 주어지면, 최종 예측 세트를 병렬로 직접 출력하기 위해 객체의 관계 및 글로벌 이미지 컨텍스트에 대한 이유를 detr.새로운 모델은 개념적으로 단순하며 다른 많은 현대 탐지기와 달리 특수 라이브러리가 필요하지 않습니다.DETR은 도전적인 Coco Object Detection DataSet에서 잘 확립되고 최적화 된 더 빠른 RCNN 기준선과 동등한 정확도 및 런타임 성능을 보여줍니다.더욱이, DETR은 통일 된 방식으로 판 토틱 세분화를 생성하기 위해 쉽게 일반화 될 수있다.우리는 그것이 경쟁 기준을 훨씬 능가한다는 것을 보여줍니다.교육 코드 및 사전 처리 된 모델은 https://github.com/facebookresearch/detr에서 제공됩니다.

객체 감지의 목표는 관심있는 각 객체에 대한 경계 박스 세트 및 범주 레이블을 예측하는 것입니다.현대의 탐지기는 대규모 제안 세트 [37,5], 앵커 [23] 또는 윈도우 센터 [53,46]에서 대리 회귀 및 분류 문제를 정의 함으로써이 세트 예측 작업을 간접적 인 방식으로 해결합니다.그들의 성능은 앵커 세트의 설계 및 대상 상자를 앵커에 할당하는 휴리스틱에 의해 거의 약한 예측을 막기위한 후 처리 단계에 크게 영향을받습니다 [52].이러한 파이프 라인을 단순화하기 위해 대리 작업을 우회하기위한 직접 설정 예측 접근법을 제안합니다.이 엔드 투 엔드 철학은 기계 번역 또는 음성 인식과 같은 복잡한 구조화 된 예측 작업에서 상당한 발전을 가져 왔지만 아직 객체 감지에서는 그렇지 않은 경우 : 이전 시도 [43,16,4,39]는 다른 형태의 사전 지식을 추가합니다.도전적인 벤치 마크에서 강력한 기준선과 경쟁력이있는 것으로 입증되지 않았습니다.이 백서는이 격차를 해소하는 것을 목표로합니다.

그림 1 : DETR은 공통 CNN과 변압기 아키텍처를 결합하여 최종 탐지 세트를 직접 예측합니다 (병렬).훈련하는 동안, 이중 파파트는 어울리는 지상 진실 상자와 함께 예측을 할당합니다.일치하지 않는 예측은 "객체 없음"(∅) 클래스 예측을 산출해야합니다.

직접 세트 예측 문제로 객체 감지를 볼 때 교육 파이프 라인을 간소화합니다.우리는 서열 예측을위한 인기있는 아키텍처 인 Transformers [47]를 기반으로 한 인코더 디코더 아키텍처를 채택합니다.순서대로 요소들 사이의 모든 쌍별 상호 작용을 명시 적으로 모델링하는 변압기의 자체 변환 메커니즘은 이들 아키텍처가 중복 예측 제거와 같은 세트 예측의 특정 제약에 특히 적합하게 만듭니다.

우리의 검출 변압기 (DETR, 그림 1 참조)는 한 번에 모든 물체를 예측하고 예측 된 물체와 지상 진실 대상 사이의 이중 파파트 매칭을 수행하는 세트 손실 함수를 갖춘 엔드 투 엔드 교육을받습니다.DETR은 공간 앵커 또는 비 최대 억제와 같은 사전 지식을 인코딩하는 여러 손으로 디자인 된 구성 요소를 삭제하여 감지 파이프 라인을 단순화합니다.대부분의 기존 탐지 방법과 달리 DETR은 사용자 정의 레이어가 필요하지 않으므로 표준 CNN 및 변압기 클래스를 포함하는 모든 프레임 워크에서 쉽게 재현 할 수 있습니다.

직접 세트 예측에 대한 대부분의 이전 연구와 비교하여 DETR의 주요 특징은 이중 파파 타이트 매칭 손실과 (비 유적지) 병렬 디코딩과의 변압기의 결합이다 [29,12,10,8].대조적으로, 이전 연구는 RNN과의 자기 회귀 디코딩에 중점을 두었다 [43,41,30,36,42].우리의 일치하는 손실 함수는 지상 진실 대상에 예측을 고유하게 할당하며 예측 된 객체의 순열에 변하지 않으므로 병렬로 방출 할 수 있습니다.우리는 매우 인기있는 객체 감지 데이터 세트 중 하나 인 CoCO [24]에서 매우 경쟁력있는 R-CNN 기준선 [37]에 대해 DETR을 평가합니다.더 빠른 RCNN은 많은 설계 반복을 겪었으며 원래 출판물 이후 성능이 크게 향상되었습니다.우리의 실험은 새로운 모델이 비슷한 성능을 달성한다는 것을 보여줍니다.보다 정확하게, DETR은 큰 물체에서 훨씬 더 나은 성능을 보여 주며, 결과는 변압기의 비 국소 계산에 의해 가능할 수 있습니다.그러나 작은 물체에서 성능이 낮아집니다.우리는 미래의 작업이 FPN의 개발이 더 빠른 R-CNN을 위해 수행 한 것과 같은 방식으로 이러한 측면을 향상시킬 것으로 기대합니다.

DETR에 대한 교육 설정은 여러 가지 방법으로 표준 객체 탐지기와 다릅니다.새로운 모델은 긴 교육 일정이 필요하며 변압기의 보조 디코딩 손실로 인한 혜택을받습니다.우리는 시연 된 성능에 어떤 구성 요소가 중요한지 철저히 탐구합니다.

DETR의 설계 정신은보다 복잡한 작업으로 쉽게 확장됩니다.우리의 실험에서, 우리는 최근에 인기를 얻은 도전적인 픽셀 수준 인식 작업 인 Panoptic Segmentation에서 경쟁이 치열한 경쟁 기준에 대해 사전에 걸린 Detr Outfoms 경쟁 기준 위에 훈련 된 간단한 세분화 헤드가 보여줍니다.

우리의 작업은 여러 도메인의 이전 작업을 기반으로합니다. 세트 예측을위한 이분자 매칭 손실, 변압기, 병렬 디코딩 및 객체 감지 방법을 기반으로 한 엔코더-디코더 아키텍처.

세트를 직접 예측하는 표준 딥 러닝 모델은 없습니다.기본 세트 예측 작업은 기준선 접근 방식 인 1-VS 실용이요소 사이의 기본 구조 (즉, 거의 동일한 상자).이 작업의 첫 번째 어려움은 거의 중독성을 피하는 것입니다.대부분의 전류 감지기는이 문제를 해결하기 위해 비 최대 억제와 같은 후 프로세싱을 사용하지만 직접 설정 예측은 후 처리가 필요하지 않습니다.중복성을 피하기 위해 예측 된 모든 요소 간의 상호 작용을 모델링하는 글로벌 추론 체계가 필요합니다.일정한 크기의 세트 예측의 경우, 밀집된 완전히 연결된 네트워크 [9]는 충분하지만 비용이 많이 듭니다.일반적인 접근법은 재발 신경 네트워크와 같은 자동 반복 시퀀스 모델을 사용하는 것입니다 [48].모든 경우에, 손실 함수는 예측의 순열에 의해 변하지 않아야한다.일반적인 솔루션은 헝가리 알고리즘 [20]을 기반으로 손실을 설계하여 지상 진실과 예측 사이의 분파 적 일치를 찾는 것입니다.이는 순열 불변성을 시행하고 각 대상 요소에 고유 한 일치를 보장합니다.우리는 이중 파파트 매칭 손실 접근법을 따릅니다.그러나 대부분의 이전 작업과 달리, 우리는자가 회귀 모델에서 벗어나 병렬 디코딩과 함께 변압기를 사용합니다.

변압기는 Vaswani et al.[47] 기계 번역을위한 새로운주의 기반 빌딩 블록으로.주의 메커니즘 [2]은 전체 입력 순서에서 정보를 집계하는 신경망 계층입니다.변압기는 비 국소 신경 네트워크 [49]와 유사하게 시퀀스의 각 요소를 스캔하고 전체 시퀀스에서 정보를 집계하여 업데이트하는 자체 변환 계층을 도입했습니다.주의 기반 모델의 주요 장점 중 하나는 글로벌 계산과 완벽한 메모리로 긴 시퀀스에서 RNN보다 더 적합합니다.트랜스포머는 현재 자연어 처리, 언어 처리 및 컴퓨터 비전에서 많은 문제에서 RNN을 대체하고 있습니다 [8,27,45,34,31].

트랜스포머는 초기 시퀀스 시퀀스 모델 [44]에 따라 자동 반복 모델에서 처음으로 사용되었으며, 출력 토큰을 하나씩 생성했습니다.그러나 오디오 도메인 [29], 기계 번역 [12,10], 워드 표현 학습 [8], 오디오 도메인 [29]에서 병렬 서열 생성이 발생하는 금지 추론 비용 (출력 길이에 비례 및 배치가 어려워집니다.그리고보다 최근에는 음성 인식 [6].또한 계산 비용과 설정 예측에 필요한 글로벌 계산 능력 사이의 적절한 트레이드 오프를 위해 변압기와 병렬 디코딩을 결합합니다.

대부분의 현대 객체 감지 방법은 초기 추측에 비해 예측을합니다.2 단계 탐지기 [37,5] 예측 상자 W.R.T.단일 단계 방법은 예측을 W.R.T.앵커 [23] 또는 가능한 물체 중심의 그리드 [53,46].최근의 연구 [52]는 이러한 시스템의 최종 성능이 이러한 초기 추측이 설정된 정확한 방식에 크게 의존한다는 것을 보여줍니다.우리의 모델에서 우리는이 손으로 만들어진 프로세스를 제거하고 절대 상자 예측 W.R.T.로 일련의 탐지 세트를 직접 예측하여 감지 프로세스를 간소화 할 수 있습니다.앵커가 아닌 입력 이미지.

설정 기반 손실.몇몇 객체 감지기 [9,25,35]는 2 분파 일치 손실을 사용했습니다.그러나, 이러한 초기 딥 러닝 모델에서, 상이한 예측 사이의 관계는 컨볼 루션 또는 완전히 연결된 층으로 만 모델링되었으며 손으로 디자인 된 NMS 후 처리는 성능을 향상시킬 수있다.보다 최근의 탐지기 [37,23,53]는 NMS와 함께 지상 진실과 예측 사이에 비 유니 퀴크 할당 규칙을 사용합니다.

학습 가능한 NMS 방법 [16,4] 및 관계 네트워크 [17]는주의를 기울인 다른 예측 사이의 관계를 명시 적으로 모델링합니다.직접 세트 손실을 사용하면 사후 처리 단계가 필요하지 않습니다.그러나 이러한 방법은 제안 상자와 같은 추가 손으로 제작 된 컨텍스트 기능을 사용하여 탐지 간의 모델 관계를 효율적으로 모델링하는 반면, 모델에서 인코딩 된 사전 지식을 줄이는 솔루션을 찾습니다.

반복 검출기.우리의 접근 방식에 가장 가까운 것은 물체 감지 [43] 및 인스턴스 세분화 [41,30,36,42]에 대한 엔드 투 엔드 세트 예측입니다.우리와 마찬가지로, 그들은 CNN 활성화를 기반으로 인코더-디코더 아키텍처를 사용하여 양당-매칭 손실을 사용하여 경계 박스 세트를 직접 생성합니다.그러나 이러한 접근법은 현대 기준선에 대한 것이 아니라 작은 데이터 세트에서만 평가되었습니다.특히, 이들은 자기 회귀 모델 (보다 정확하게 RNN)을 기반으로하므로 최근 변압기를 병렬 디코딩으로 활용하지 않습니다.

두 가지 성분은 탐지에서 직접 세트 예측에 필수적입니다. (1) 예측 된 진실 상자와 지상 진실 상자 사이에 고유 한 일치를 강요하는 정해진 예측 손실;(2) 일련의 객체를 (단일 패스로) 예측하는 아키텍처와 그들의 관계를 모델링합니다.우리는 그림 2에서 아키텍처를 자세히 설명합니다.

DETR은 디코더를 통한 단일 패스에서 고정 크기 N 예측 세트를 유추하며, 여기서 N은 이미지의 일반적인 객체 수보다 상당히 큰 것으로 설정됩니다.훈련의 주요 어려움 중 하나는 지상 진실과 관련하여 예측 된 대상 (클래스, 위치, 크기)을 점수하는 것입니다.우리의 손실은 예측 된 진실 객체와 지상 진실 객체 사이에 최적의 이중 파파트 일치를 생성 한 다음 객체 별 (경계 박스) 손실을 최적화합니다.

우리가 y 객체의 지상 진실 세트와 n 예측 세트를 표시합시다.n이 이미지의 물체 수보다 크다고 가정하면 y는 ∅ (객체 없음)로 패딩 된 크기 n 세트로 간주합니다.이 두 세트 사이의 양자당 일치를 찾으려면 가장 낮은 비용으로 N 요소 σ ∈ SN의 순열을 검색합니다.

여기서 lmatch (yi, y σ (i))는 지상 진실 yi와 인덱스 σ (i)와의 예측 사이의 쌍별 일치 비용입니다.이 최적의 과제는 이전 작업에 따라 헝가리 알고리즘으로 효율적으로 계산됩니다 (예 : [43]).

일치하는 비용은 클래스 예측과 예측 된 진실 상자의 유사성을 모두 고려합니다.지상 진실 세트의 각 요소 I는 yi = (ci, bi)로 볼 수 있습니다.중앙 좌표 및 이미지 크기에 대한 높이 및 너비.인덱스 σ (i)를 사용한 예측을 위해 클래스 CI의 확률을 ˆpσ (i) (ci)로, 예측 된 상자를 ˆbσ (i)로 정의합니다.이 표기법으로 우리는 lmatch (yi, y σ (i))를 다음과 같이 정의합니다

일치하는이 절차는 현대 탐지기의 제안 [37] 또는 앵커 [22]와 일치하는 데 사용되는 휴리스틱 할당 규칙과 동일한 역할을합니다.주요 차이점은 복제없이 직접 세트 예측을 위해 일대일 매칭을 찾아야한다는 것입니다.

두 번째 단계는 손실 함수를 계산하는 것입니다. 이전 단계에서 일치하는 모든 쌍의 헝가리 손실.우리는 공통 객체 감지기의 손실, 즉 클래스 예측에 대한 음의 로그-원성과 나중에 정의 된 박스 손실의 선형 조합과 유사하게 손실을 정의합니다.

여기서 ˆσ는 첫 번째 단계 (1)에서 계산 된 최적의 할당입니다.실제로, 우리는 클래스 불균형을 설명하기 위해 ci = ∅에 ci = ∅ 할 때 로그 확률 항을 중량으로 중량화합니다.이것은 R-CNN 훈련 절차가 서브 샘플링을 통해 긍정적/부정적인 제안의 균형을 얼마나 빠른지와 유사하다 [37].물체와 ∅ 사이의 일치하는 비용은 예측에 의존하지 않으므로이 경우 비용이 일정하다는 것을 의미합니다.일치하는 비용에서 우리는 확률 ˆpσˆ (i) (ci)를 로그-자극성 대신 사용합니다.이것은 클래스 예측 용어를 Lbox (·, ·) (아래 설명)로 만들 수있게 해주 며, 더 나은 경험적 성능을 관찰했습니다.

경계 박스 손실.일치하는 비용과 헝가리 손실의 두 번째 부분은 경계 상자를 채점하는 Lbox (·)입니다.상자 예측을 ∆ W.R.T.로 수행하는 많은 탐지기와 달리초기 추측은 상자 예측을 직접 만듭니다.이러한 접근 방식은 구현을 단순화하지만 상대적인 손실 스케일링에 문제가 있습니다.가장 일반적으로 사용되는`1 손실은 상대 오류가 비슷하더라도 작고 큰 상자의 스케일이 다릅니다.이 문제를 완화하기 위해 우리는`1 손실과 일반화 된 IOU 손실 [38] liou (·, ·)의 선형 조합을 사용합니다.전반적으로, 우리의 박스 손실은 λiouliou (bi, ˆbσ (i)) + λl1 || bi - ˆbσ (i) || 1로 정의 된 lbox (bi, ˆbσ (i))입니다.이 두 손실은 배치 내부의 물체 수에 의해 정규화됩니다.

전체 DETR 아키텍처는 놀랍게도 단순하고 그림 2에 묘사되어 있습니다. 여기에는 다음과 같은 세 가지 주요 구성 요소가 포함되어 있습니다. 여기에는 다음에 설명하는 CNN 백본, 컴팩트 한 기능 표현, 인코더 디코더 변압기 및 간단한 피드 포워드 네트워크 (FFN)가 포함됩니다.최종 탐지 예측을 만듭니다.

많은 현대 탐지기와 달리 DETR은 공통 CNN 백본과 수백 줄의 변압기 아키텍처 구현을 제공하는 딥 러닝 프레임 워크에서 구현할 수 있습니다.DETR에 대한 추론 코드는 Pytorch [32]에서 50 라인 미만으로 구현 될 수 있습니다.우리는 우리의 방법의 단순성이 새로운 연구자들을 탐지 커뮤니티에 끌어들일 수 있기를 바랍니다.

등뼈.초기 이미지에서 시작하여 XIMG ∈ R3 × H0 × W0 (3 개의 컬러 채널 포함)에서 기존의 CNN 백본은 하부 해상도 활성화 맵 F ∈ RC × H × W를 생성합니다.우리가 사용하는 일반적인 값은 C = 2048입니다.

변압기 인코더.먼저, 1x1 컨볼 루션은 고급 활성화 맵 F의 채널 차원을 C에서 더 작은 치수 d로 감소시킵니다. d.새 기능 맵 생성 z0 ∈ Rd × H × W.인코더는 시퀀스를 입력으로 예상하므로 Z0의 공간 치수를 1 차원으로 붕괴시켜 D × HW 기능 맵을 초래합니다.각 인코더 층은 표준 아키텍처를 가지며 멀티 헤드 자체 변환 모듈과 ​​FFN (Feed Forder Network)으로 구성됩니다.트랜스포머 아키텍처는 순열 불균형이므로 각주의 계층의 입력에 추가되는 고정 위치 인코딩 [31,3]으로 보충합니다.우리는 보충 자료에 건축의 상세한 정의를 연기하며, 이는 [47]에 설명 된 것을 따릅니다.

그림 2 : DETR은 기존 CNN 백본을 사용하여 입력 이미지의 2D 표현을 학습합니다.모델은 그것을 평평하게하고 변압기 인코더로 전달하기 전에 위치 인코딩으로 보충합니다.그런 다음 변압기 디코더는 소량의 고정 된 수의 학습 위치 임베드를 입력하여 객체 쿼리를 호출하고 인코더 출력에 추가로 참석합니다.디코더의 각 출력 임베딩을 감지 (클래스 및 경계 박스) 또는 "객체 없음"클래스를 예측하는 공유 피드 포워드 네트워크 (FFN)로 전달합니다.

변압기 디코더.디코더는 변압기의 표준 아키텍처를 따라 다중 머리 자체 및 인코더 디코더주의 메커니즘을 사용하여 크기 D의 N 임베딩을 변환합니다.원래 변압기와의 차이점은 모델이 각 디코더 층에서 N 물체를 병렬로 디코딩하는 반면 Vaswani et al.[47] 출력 시퀀스를 한 번에 한 요소 씩 예측하는 자동 회귀 모델을 사용하십시오.우리는 개념에 익숙하지 않은 독자를 보충 자료에 대해 언급합니다.디코더는 또한 순열 불균형이므로 N 입력 임베딩은 다른 결과를 생성하기 위해 달라야합니다.이러한 입력 임베딩은 객체 쿼리라고하는 위치 인코딩을 배우고 인코더와 유사하게 각주의 레이어의 입력에 추가합니다.N 객체 쿼리는 디코더에 의해 출력 임베딩으로 변환됩니다.그런 다음 피드 포워드 네트워크 (다음 하위 섹션에 설명)에 의해 상자 좌표 및 클래스 레이블로 독립적으로 디코딩되어 최종 예측이 발생합니다.이러한 임베딩에 대한 자체 및 인코더 디코더주의를 사용하여 전체 이미지를 컨텍스트로 사용할 수있는 동시에 모든 객체에 대한 전 세계적으로 모든 객체에 대한 이유가 있습니다.

예측 피드 포워드 네트워크 (FFN).최종 예측은 Relu 활성화 기능 및 숨겨진 치수 D 및 선형 투영 층을 갖는 3 층 퍼셉트론에 의해 계산된다.FFN은 정규화 된 중심 좌표, 상자의 높이 및 너비를 예측합니다. W.R.T.입력 이미지 및 선형 레이어는 SoftMax 함수를 사용하여 클래스 레이블을 예측합니다.n 고정 크기의 n 경계 상자 세트를 예측하기 때문에 N은 일반적으로 이미지에서 실제 관심있는 객체 수보다 훨씬 크기 때문에 추가 특수 클래스 레이블 ∅는 슬롯 내에서 객체가 감지되지 않음을 나타내는 데 사용됩니다.이 클래스는 표준 객체 감지 방법에서 "배경"클래스와 비슷한 역할을합니다.

보조 디코딩 손실.우리는 훈련 중 디코더에서 보조 손실 [1]을 사용하는 데 도움이되었으며, 특히 모델이 각 클래스의 올바른 수의 객체를 출력하는 데 도움이되었습니다.각 디코더 층 후 예측 FFN 및 헝가리 손실을 추가합니다.모든 예측 FFN은 매개 변수를 공유합니다.우리는 추가 공유 레이어-노름을 사용하여 다른 디코더 층의 예측 FFN에 대한 입력을 정규화합니다.

우리는 DETR이 COCO에 대한 정량적 평가에서 더 빠른 R-CNN에 비해 경쟁 결과를 달성 함을 보여줍니다.그런 다음 통찰력과 질적 결과와 함께 건축 및 손실에 대한 자세한 절제 연구를 제공합니다.마지막으로, DETR이 다재다능하고 확장 가능한 모델임을 보여주기 위해, 우리는 Panoptic 세분화에 대한 결과를 제시하여 고정 DETR 모델에서 작은 확장 만 훈련합니다.우리는 https://github.com/facebookresearch/detr에서 실험을 재현하기위한 코드 및 사전 치료 모델을 제공합니다.

데이터 세트.우리는 118K 훈련 이미지와 5K 검증 이미지를 포함하는 Coco 2017 탐지 및 Panoptic 세분화 데이터 세트 [24,18]에 대한 실험을 수행합니다.각 이미지에는 경계 상자 및 판 토틱 세분화로 주석이 달라집니다.동일한 이미지에서는 작은 것부터 큰 것부터 큰 훈련 세트의 단일 이미지에서 평균적으로 이미지 당 7 개의 인스턴스가 있습니다.지정되지 않으면 AP를 여러 임계 값에 대한 적분 지표 인 Bbox AP로보고합니다.더 빠른 R-CNN과 비교하기 위해 마지막 훈련 에포크에서 검증 AP를보고합니다. 절제에 대해서는 마지막 10 개의 에포크의 유효성 검사 결과 중앙값을보고합니다.

기술적 인 세부 사항.우리는 ADAMW [26]로 DETR을 훈련시켜 초기 변압기의 학습 속도를 10-4, 백본의 10-5, 무게 부패로 10-4로 설정합니다.모든 변압기 가중치는 Xavier Init [11]로 초기화되며, 백본은 냉동 배치 름기 층을 갖는 Torchvision의 ImageNet-restrained RESNET 모델 [15]으로 초기화됩니다.우리는 RESNET50과 RESNET-101의 두 가지 다른 백본으로 결과를보고합니다.해당 모델을 각각 DETR 및 DETR-R101이라고합니다.[21]에 이어, 우리는 또한 백본의 마지막 단계에 팽창을 추가 하고이 단계의 첫 번째 컨볼 루션에서 보폭을 제거함으로써 기능 해상도를 증가시킵니다.상응하는 모델은 각각 DETR-DC5 및 DETR-DC5-R101 (확장 된 C5 단계)이라고합니다.이 수정은 해상도를 2 배로 증가시켜 엔코더의 자체 항목에서 16 배 높은 비용으로 작은 물체의 성능을 향상시켜 계산 비용의 전체 2 배 증가를 초래합니다.이들 모델의 플롭과 더 빠른 R-CNN의 전체 비교는 표 1에 나와있다.

우리는 가장 짧은면이 최소 480이고 최대 800 픽셀 인 반면 최대 1333 [50]을 조정하여 입력 이미지를 크기를 조정합니다.인코더의 자체 변환을 통해 글로벌 관계를 배우는 데 도움이되기 위해 훈련 중 임의의 작물 증강을 적용하여 약 1 AP만큼 성능을 ​​향상시킵니다.구체적으로, 트레인 이미지는 랜덤 직사각형 패치에 0.5로 확률로 잘린 다음 800-1333으로 다시 크기를 조정합니다.변압기는 기본 드롭 아웃 0.1로 훈련됩니다.추론

표 1 : Coco 검증 세트에서 RESNET-50 및 RESNET-101 백본과 더 빠른 R-CNN과 비교.상단 섹션은 Detectron2 [50]에서 더 빠른 R-CNN 모델에 대한 결과를 보여 주며, 중간 섹션은 GIOU [38], Random Crops Train Time 확대 및 긴 9X 훈련 일정을 가진 더 빠른 R-CNN 모델에 대한 결과를 보여줍니다.DETR 모델은 APS가 낮지 만 APL이 크게 개선 된 더 빠른 R-CNN 기준선과 비슷한 결과를 얻습니다.우리는 Torchscript 더 빠른 R-CNN 및 DETR 모델을 사용하여 플롭 및 FPS를 측정합니다.이름으로 R101이없는 결과는 RESNET-50에 해당합니다.

시간, 일부 슬롯은 빈 클래스를 예측합니다.AP를 최적화하기 위해 해당 신뢰를 사용하여 두 번째로 높은 스코어링 클래스로 이러한 슬롯의 예측을 무시합니다.이렇게하면 빈 슬롯을 필터링하는 것과 비교하여 AP를 2 포인트로 향상시킵니다.다른 훈련 초분수는 섹션 A.4에서 찾을 수 있습니다.우리의 절제 실험의 경우, 우리는 200 개의 에포크 후 10의 계수로 학습 속도가 감소한 300 개의 에포크의 훈련 일정을 사용합니다.16 V100 GPU에서 300 개의 에포크에 대한 기준선 모델을 훈련시키는 데 3 일이 걸리며 GPU 당 4 개의 이미지 (따라서 총 배치 크기 64).더 빠른 R-CNN과 비교하는 데 사용되는 더 긴 일정의 경우 400 개의 에포크 후 학습 속도 감소로 500 개의 에포크를 훈련시킵니다.이 일정은 짧은 일정에 비해 1.5 AP를 추가합니다.

트랜스포머는 일반적으로 교육 일정과 중퇴가 매우 긴 Adam 또는 Adagrad Optimizers로 훈련되어 있으며 DETR도 마찬가지입니다.그러나 더 빠른 R-CNN은 최소한의 데이터 확대로 SGD로 교육을받으며 Adam 또는 Dropout의 성공적인 응용 프로그램을 알지 못합니다.이러한 차이에도 불구하고 우리는 더 빠른 R-CNN 기준선을 강하게 만들려고 노력합니다.DETR과 정렬하기 위해, 우리는 상자 손실에 일반화 된 IOU [38]를 추가하여 결과를 개선하는 것으로 알려진 동일한 임의의 작물 확대 및 긴 훈련을 상자 손실에 추가합니다 [13].결과는 표 1에 나와 있습니다. 상단 섹션에서는 3 배 스케줄로 훈련 된 모델에 대한 Detectron2 Model Zoo [50]의 더 빠른 R-CNN 결과를 보여줍니다.중간 섹션에서 우리는 동일한 모델에 대한 결과 ( "+")를 보여 주지만 훈련 된 결과를 보여줍니다.

표 2 : 인코더 크기의 영향.각 행은 다양한 수의 인코더 층과 고정 된 수의 디코더 층을 갖는 모델에 해당합니다.더 많은 인코더 레이어로 성능이 점차 향상됩니다.

9X 일정 (109 에포크)과 설명 된 개선 사항을 사용하여 총 1-2 AP를 추가합니다.표 1의 마지막 섹션에서는 여러 DETR 모델에 대한 결과를 보여줍니다.매개 변수의 수에서 비교할 수 있으려면 6 개의 변압기와 6 개의주의 헤드가있는 6 개의 변압기와 6 개의 디코더 층이있는 모델을 선택합니다.FPN이있는 더 빠른 R-CNN과 마찬가지로이 모델에는 41.3m 매개 변수가 있으며 그 중 23.5m는 RESNET-50이고 17.8m는 변압기에 있습니다.더 빠른 R-CNN과 DETR이 여전히 더 긴 훈련으로 더욱 향상 될 가능성이 있지만 DETR이 동일한 수의 매개 변수로 더 빠른 R-CNN과 경쟁력을 발휘하여 Coco Val 서브 세트에서 42 AP를 달성 할 수 있다고 결론 지을 수 있습니다.Detr이이를 달성하는 방식은 APL (+7.8)을 개선하는 것입니다. 그러나 모델은 여전히 ​​APS에서 뒤쳐져 있습니다 (-5.5).동일한 수의 매개 변수 및 유사한 플롭 카운트를 갖는 DETR-DC5는 AP가 높지만 APS에서도 여전히 상당히 뒤떨어져 있습니다.RESNET-101 백본으로 더 빠른 R-CNN 및 DETR은 비슷한 결과를 보여줍니다.

변압기 디코더의주의 메커니즘은 다른 탐지의 기능 표현간에 관계를 모델링하는 주요 구성 요소입니다.절제 분석에서 우리는 아키텍처 및 손실의 다른 구성 요소가 최종 성능에 어떤 영향을 미치는지 탐구합니다.이 연구의 경우 6 개의 인코더, 6 개의 디코더 층 및 너비 256을 갖는 RESNET-50 기반 DETR 모델을 선택합니다.이 모델은 41.3m 매개 변수를 가지며 각각 단기 및 긴 일정에서 40.6 및 42.0 AP를 달성하며 28 FPS에서 실행됩니다.동일한 백본으로 더 빠른 R-CNN-FPN.

인코더 층의 수.우리는 인코더 층의 수를 변경하여 글로벌 Imagelevel 자체 변환의 중요성을 평가합니다 (표 2).인코더 층이 없으면 전체 AP는 3.9 포인트로 떨어지고 큰 물체에서 6.0 AP의 더 큰 드롭이 줄어 듭니다.우리는 글로벌 장면 추론을 사용하여 인코더가 객체를 분리하는 데 중요하다는 가설을 세웁니다.그림 3에서는 이미지의 몇 가지 점에 중점을 둔 숙련 된 모델의 마지막 인코더 층의주의지도를 시각화합니다.인코더는 이미 인스턴스를 분리하는 것으로 보이며, 이는 디코더의 객체 추출 및 국소화를 단순화 할 수 있습니다.

디코더 층의 수.각 디코딩 층 후에 보조 손실을 적용합니다 (섹션 3.2 참조). 따라서 예측 FFN은 설계에 의해 사전에 교육됩니다.

그림 3 : 일련의 기준점에 대한 인코더 자체 변환.인코더는 개별 인스턴스를 분리 할 수 ​​있습니다.검증 세트 이미지에서 기준선 DETR 모델로 예측이 이루어집니다.

모든 디코더 층의 출력에서 ​​객체를 dict.우리는 디코딩의 각 단계에서 예측 될 물체를 평가하여 각 디코더 층의 중요성을 분석합니다 (그림 4).AP와 AP50은 모든 계층 후에 개선되어 첫 번째 층과 마지막 층 사이의 매우 유의 한 +8.2/9.5 AP 개선으로 전체적으로 개선됩니다.설정 기반 손실로 DETR은 설계별로 NMS가 필요하지 않습니다.이를 확인하기 위해 각 디코더 후 출력에 대한 기본 매개 변수 [50]가있는 표준 NMS 절차를 실행합니다.NMS는 첫 번째 디코더의 예측 성능을 향상시킵니다.이것은 변압기의 단일 디코딩 층이 출력 요소들 사이의 상호 상관 관계를 계산할 수 없다는 사실에 의해 설명 될 수 있으므로 동일한 물체에 대해 여러 예측을하기 쉽다.두 번째 및 후속 층에서, 활성화에 대한 자체 변환 메커니즘은 모델이 중복 예측을 억제 할 수있게한다.우리는 깊이가 증가함에 따라 NMS에 의한 개선이 감소 함을 관찰합니다.마지막 층에서는 NMS가 실제 긍정적 예측을 잘못 제거함에 따라 AP의 작은 손실을 관찰합니다.

인코더주의 시각화와 마찬가지로, 우리는 그림 6에서 디코더 관심을 시각화하여 각각의 예측 된 물체의 다른 색상으로주의 맵을 채색합니다.우리는 디코더의 관심이 상당히 국소 적이라는 것을 관찰합니다. 즉, 주로 머리 나 다리와 같은 물체 사지에 참석합니다.우리는 인코더가 글로벌주의를 통해 인스턴스를 분리 한 후에 디코더가 클래스와 객체 경계를 추출하기 위해 사지에만 참석하면됩니다.

FFN의 중요성.Tranformers 내부의 FFN은 1 × 1 컨볼 루션 층으로 볼 수 있으며, 인코더는주의 증강 컨볼 루션 네트워크와 유사하게 만듭니다 [3].우리는 변압기 층에서만주의를 기울이지 않으려 고 노력합니다.네트워크 매개 변수의 수를 41.3m에서 28.7m로 줄이면 변압기에 10.8m 만 남겨두고 성능이 2.3 AP로 떨어지면 FFN이 좋은 결과를 달성하는 데 중요하다고 결론을 내립니다.

위치 인코딩의 중요성.우리 모델에는 두 가지 종류의 위치 인코딩이 있습니다 : 공간 위치 인코딩 및 출력 위치 인코딩-

ings (객체 쿼리).우리는 고정 및 학습 인코딩의 다양한 조합을 실험하며, 결과는 표 3에서 찾을 수 있습니다. 출력 위치 인코딩이 필요하고 제거 할 수 없으므로 디코더 입력시 한 번 전달하거나 모든 디코더주의 레이어에서 쿼리에 추가하는 것을 실험합니다.첫 번째 실험에서 우리는 입력시 공간 위치 인코딩 및 패스 출력 위치 인코딩을 완전히 제거하고 흥미롭게도 모델은 여전히 ​​32 AP 이상을 달성하여 기준선에 7.8 AP를 잃습니다.그런 다음 원래 변압기 [47]에서와 같이 고정 된 사인 공간 위치 인코딩 및 출력 인코딩을 입력 한 번에 전달하고, 이는 위치 인코딩을 직접주의를 기울이는 것과 비교하여 1.4 AP 드롭으로 이어진다는 것을 발견합니다.배운 공간 인코딩이 관심을 끌면서 비슷한 결과를 얻습니다.놀랍게도, 우리는 인코더에서 공간 인코딩을 전달하지 않으면 1.3 AP의 작은 AP 드롭만으로 이어집니다.인코딩을 관심에 전달하면 모든 계층에서 공유되며 출력 인코딩 (객체 쿼리)은 항상 학습됩니다.

이러한 절제를 감안할 때, 우리는 변압기 구성 요소 : 인코더, FFN, 다중 디코더 층 및 위치 인코딩의 전역 자체 변환이 모두 최종 객체 감지 성능에 크게 기여한다고 결론 지었다.

손실 절제.일치하는 비용과 손실의 다른 구성 요소의 중요성을 평가하기 위해 여러 모델을 켜고 끄는 여러 모델을 훈련시킵니다.손실에는 분류 손실,`1 경계 박스 거리 손실, Giou [38] 손실에 세 가지 구성 요소가 있습니다.분류 손실은 훈련에 필수적이며 꺼질 수 없으므로 우리는 경계 박스 거리 손실없이 모델을 훈련시키고 GIOU 손실이없는 모델을 훈련시키고 세 가지 손실과 함께 훈련 된 기준선과 비교합니다.결과는 표 4에 나와 있습니다. 자체 계정의 GIOU 손실

그림 6 : 모든 예측 된 물체에 대한 디코더주의 시각화 (Coco Val 세트의 이미지).예측은 DETR-DC5 모델로 만들어집니다.주의 점수는 다른 객체에 대해 다른 색상으로 코딩됩니다.디코더는 일반적으로 다리와 머리와 같은 물체 사지에 참석합니다.색상으로 가장 잘 보입니다.

표 3 : 사인 PO가 고정 된 기준선 (마지막 행)과 비교하여 다른 위치 인코딩에 대한 결과.인코딩은 인코더와 디코더의 모든주의 층에서 전달되었습니다.배운 임베딩은 모든 계층간에 공유됩니다.공간 위치 인코딩을 사용하지 않으면 AP가 크게 감소합니다.흥미롭게도, 디코더에 전달하면 사소한 AP 드롭만으로 이어집니다.이 모든 모델은 학습 된 출력 위치 인코딩을 사용합니다.공간 POS.enc.출력 pos.enc.

없음 입력 32.8 -7.8 55.2 -6.5 입력에 입력 사인에 사인을 배웠다.ATTN에서 배웠습니다.ATTN에서 배웠습니다.39.6 -1.0 60.7 -0.9 ATTN에서 사인.ATTN에서 배웠습니다.39.3 -1.3 60.3 -1.4 Attn에서 사인.Attn에서 사인.ATTN에서 배웠습니다.40.6-61.6-

표 4 : AP에 대한 손실 성분의 영향.우리는`1 손실과 Giou 손실을 끄고`1이 자체적으로 불량한 결과를 제공한다는 것을 관찰하지만 Giou와 결합하면 APM 및 APL을 향상 시킨다는 것을 관찰합니다.우리의 기준선 (마지막 행)은 두 손실을 결합합니다.

그림 7 : Coco 2017 VAL의 모든 이미지에 대한 모든 상자 예측 시각화 Detr Decoder의 총 n = 100 예측 슬롯 중 20 개에 대한 세트.각 상자 예측은 각 이미지 크기에 의해 정규화 된 1 x-1 제곱의 중심 좌표가있는 점으로 표시됩니다.점은 색상으로 구분되므로 녹색은 작은 상자, 빨간색에서 큰 수평 상자, 파란색에서 큰 수직 상자에 해당합니다.우리는 각 슬롯이 여러 운영 모드로 특정 영역과 박스 크기를 전문화하는 법을 배웁니다.거의 모든 슬롯에는 Coco 데이터 세트에서 흔한 큰 이미지 전체 상자를 예측하는 모드가 있습니다.

디코더 출력 슬롯 분석 그림 7 우리는 Coco 2017 VAL 세트의 모든 이미지에 대해 다른 슬롯으로 예측 된 상자를 시각화합니다.DETR은 각 쿼리 슬롯마다 다른 전문화를 배웁니다.우리는 각 슬롯이 다른 영역과 박스 크기에 중점을 둔 여러 작동 모드를 가지고 있음을 관찰합니다.특히, 모든 슬롯에는 이미지 전체 상자를 예측하기위한 모드가 있습니다 (플롯의 중간에 빨간 도트가 정렬 된 빨간 도트가 표시됨).우리는 이것이 Coco의 물체 분포와 관련이 있다는 것을 가설합니다.

보이지 않는 수의 인스턴스에 대한 일반화.Coco의 일부 클래스는 동일한 이미지에서 동일한 클래스의 많은 인스턴스로 잘 표현되지 않습니다.예를 들어, 훈련 세트에 13 개 이상의 기린이있는 이미지가 없습니다.우리는 DETR의 일반화 능력을 확인하기 위해 합성 이미지를 만듭니다 (그림 5 참조).우리의 모델은 이미지에서 24 개의 기린을 모두 찾을 수 있으며, 이는 분포가 분명하지 않습니다.이 실험은 각 객체 쿼리에 강력한 클래스 전문화가 없음을 확인합니다.

Panoptic 세분화 [19]는 최근 컴퓨터 비전 커뮤니티로부터 많은 관심을 끌었다.더 빠른 R-CNN [37]의 R-CNN [14]의 확장과 유사하게, 디코더 출력 위에 마스크 헤드를 추가하여 DETR을 자연스럽게 확장 할 수있다.이 섹션에서 우리는 그러한 머리가 물건과 물건 수업을 치료함으로써 panoptic segmentation [19]을 생성하는 데 사용될 수 있음을 보여줍니다.

그림 8 : Panoptic 헤드의 그림.이진 마스크는 각각의 감지 된 물체에 대해 병렬로 생성 된 다음 마스크는 픽셀 별 Argmax를 사용하여 병합된다.

그림 9 : DETR-R101에 의해 생성 된 Panoptic 세분화에 대한 질적 결과.DETR은 사물과 물건에 대해 통일 된 방식으로 정렬 된 마스크 예측을 생성합니다.

통일 된 방식으로.우리는 80 가지 카테고리 외에 53 개의 물건 범주가있는 Coco 데이터 세트의 Panoptic 주석에 대한 실험을 수행합니다.

우리는 동일한 레시피를 사용하여 Coco의 클래스와 클래스 주위의 상자를 예측하기 위해 Detr를 훈련시킵니다.헝가리 인 매칭은 상자 사이의 거리를 사용하여 계산되기 때문에 훈련이 가능하기 위해서는 상자 예측이 필요합니다.또한 예측 된 각 상자에 대한 이진 마스크를 예측하는 마스크 헤드를 추가합니다. 그림 8을 참조하십시오. 각 객체에 대한 변압기 디코더의 출력을 입력 하고이 임베딩의 멀티 헤드 (M 헤드)주의 점수를 계산합니다.엔코더의 출력으로 작은 해상도로 물체 당 M주의 열 맵을 생성합니다.최종 예측을 만들고 해상도를 높이려면 FPN 유사 아키텍처가 사용됩니다.보충제에 대한 자세한 내용으로 아키텍처를 설명합니다.마스크의 최종 해상도는 보폭 4 개이며 각 마스크는 주사위/F-1 손실 [28] 및 초점 손실 [23]을 사용하여 독립적으로 감독됩니다.

마스크 헤드는 공동으로 또는 두 단계 프로세스로 훈련을받을 수 있습니다. 여기서 우리는 박스를 위해 DETR을 훈련시킨 다음 모든 무게를 얼리고 25 개의 에포크를 위해 마스크 헤드 만 훈련합니다.실험적으로,이 두 가지 접근법은 비슷한 결과를 제공합니다. 우리는 후자의 방법을 사용하여 결과를보고하여 총 벽 클록 시간 훈련이 짧기 때문입니다.

표 5 : Coco Val 데이터 세트에서 최첨단 방법 UPSNET [51] 및 Panoptic FPN [18]과의 비교는 공정 비교를 위해 18 배 일정에서 DETR과 동일한 데이터를 사용하여 PanOpticFPN을 재교육했습니다.UPSNET은 1X 일정을 사용하고 UPSNET-M은 멀티 스케일 테스트 시간 증강 버전입니다.

최종 Panoptic 세분화를 예측하기 위해 간단히 각 픽셀의 마스크 점수를 통해 Argmax를 사용하고 해당 카테고리를 결과 마스크에 할당합니다.이 절차는 최종 마스크가 겹치지 않도록 보장하므로 DETR은 종종 다른 마스크를 정렬하는 데 사용되는 휴리스틱을 요구하지 않습니다 [19].

교육 세부 사항.우리는 Coco 데이터 세트의 물건 주위와 물건 주위의 상자를 예측하기 위해 Bounding Box Detection의 레시피에 따라 DETR, DETR-DC5 및 DETR-R101 모델을 훈련시킵니다.새로운 마스크 헤드는 25 개의 에포크를 위해 훈련되었습니다 (자세한 내용은 보충 참조).추론 중에 먼저 85%미만의 신뢰로 탐지를 필터링 한 다음 픽셀 당 Argmax를 계산하여 각 픽셀이 속한 마스크를 결정합니다.그런 다음 동일한 재료 범주의 다른 마스크 예측을 하나로 파악하고 빈 픽셀 미만을 필터링합니다.

주요 결과.질적 결과는 그림 9에 나와 있습니다. 표 5에서 우리는 통합 된 panoptic segmenation 접근법을 사물과 물건을 다르게 처리하는 몇 가지 확립 된 방법과 비교합니다.우리는 panoptic 품질 (PQ)과 사물 (PQTH) 및 물건 (PQST)을보고합니다.우리는 또한 치료 후 (우리의 경우, 픽셀 현지 Argmax를 복용하기 전에) 마스크 AP (사물 클래스에 계산)를보고합니다.우리는 DETR이 Coco-Val 2017에서 게시 된 결과와 강력한 PanOpticFPPN 기준선 (공정한 비교를 위해 DETR과 동일한 데이터 침체로 훈련)을 능가한다는 것을 보여줍니다.결과 분류는 DETR이 물건 클래스에서 특히 지배적이라는 것을 보여 주며, 인코더주의가 허용하는 글로벌 추론 이이 결과의 핵심 요소라는 가설을 세웁니다.사물 클래스의 경우, 마스크 AP 계산의 기준에 비해 최대 8 개의 맵의 심각한 결함에도 불구하고 DETR은 경쟁력있는 PQTH를 얻습니다.또한 Coco 데이터 세트의 테스트 세트에 대한 방법을 평가하고 46 PQ를 얻었습니다.우리는 우리의 접근 방식이 향후 작업에서 Panoptic 세분화를위한 완전히 통합 된 모델의 탐색에 영감을주기를 바랍니다.

우리는 변압기와 직접 세트 예측을위한 이분자 매칭 손실을 기반으로 객체 감지 시스템을위한 새로운 설계 인 DETR을 제시했습니다.이 접근법은 도전적인 Coco 데이터 세트에서 최적화 된 더 빠른 R-CNN 기준선과 비슷한 결과를 달성합니다.DETR은 구현하기가 간단하며 유연한 아키텍처를 가지고 있으며 경쟁력있는 결과와 함께 Panoptic 세분화에 쉽게 확장 할 수 있습니다.또한, 자체 정보에 의해 수행 된 글로벌 정보의 처리 덕분에 더 빠른 R-CNN보다 큰 객체에서 훨씬 더 나은 성능을 달성합니다.

이 새로운 탐지기 디자인에는 특히 작은 물체에 대한 교육, 최적화 및 성능에 관한 새로운 도전이 있습니다.현재 탐지기는 유사한 문제에 대처하기 위해 몇 년의 개선이 필요했으며, 향후 작업에서 Detr을 위해이를 성공적으로 해결할 것으로 기대합니다.

우리는 Sainbayar Sukhbaatar, Piotr Bojanowski, Natalia Neverova, David Lopez-Paz, Guillaume Lample, Danielle Rothermel, Ross Girshick, Xinlei Chen 및이 작품이 가능하지 않은 토론 및 조언에 대해 Facebook AI Research Paris 팀 전체에 감사드립니다.

Al-Rfou, R., Choe, D., Constant, N., Guo, M., Jones, L. : 더 깊은 자기 정보를 가진 문자 수준 언어 모델링.에서 : 인공 지능에 관한 AAAI 회의 (2019)

Chan, W., Saharia, C., Hinton, G., Norouzi, M., Jaitly, N. : 임피 : 대치 및 동적 프로그래밍을 통한 시퀀스 모델링.ARXIV : 2002.08926 (2020)

Devlin, J., Chang, M.W., Lee, K., Toutanova, K. : Bert : 언어 이해를위한 깊은 양방향 변압기의 사전 훈련.에서 : NAACL-HLT (2019)

He, K., Girshick, R., Dolla´r, P. : imagenet 사전 훈련을 다시 생각합니다.에서 : ICCV (2019) 14. He, K., Gkioxari, G., Dolla´r, P., Girshick, R.B. : Mask R-CNN.에서 : ICCV (2017) 15. He, K., Zhang, X., Ren, S., Sun, J. : 이미지 인식을위한 심층 잔차 학습.에서 : CVPR (2016)

Lin, T.Y., Maire, M., Progysie, S., Hays, J., Perona, P., Ramanan, D., Dolla´r, P., Zitnick, C.L. : Microsoft Coco : 맥락의 공통 개체.에서 : ECCV (2014) 25. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C.Y., Berg, A.C. : SSD : 단일 샷 Multibox Detector.에서 : ECCV (2016)

Lu¨scher, C., Beck, E., Irie, K., Kitza, M., Michel, W., Zeyer, A., Schlu¨ter, R., Ney, H. : Librispeech 용 Rwth ASR 시스템 :하이브리드 대주의 - 데이터 증강 w/o.ARXIV : 1905.03072 (2019)

Oord, A.V.D., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., Driessche, G.V.D., Lockhart, E., Cobo, L.C., Stimberg, F., et al.: 병렬와 베넷 : 빠른 고전성 음성 합성.ARXIV : 1711.10433 (2017)

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L.,Desmaison, A., Kopf, A., Yang, E., Devito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J.,Chintala, S. : Pytorch : 명령적인 스타일, 고성능 딥 러닝 라이브러리.에서 : Neurips (2019)

Pineda, L., Salvador, A., Drozdzal, M., Romero, A. : 이미지-설정 예측을 설명 : 모델, 손실 및 데이터 세트 분석.ARXIV : 1904.05709 (2019)

Rezatofighi, S.H., Kaskman, R., Motlagh, F.T., Shi, Q., Cremers, D., Leal-Taix´e, L., Reid, I. : Deep Perm-set Net : 알 수없는 순열을 가진 세트를 예측하는 법을 배우십시오.심층 신경망을 사용하는 기타.ARXIV : 1805.00613 (2018)

Rezatofighi, S.H., Milan, A., Abbasnejad, E., Dick, A., Reid, I., Kaskman, R., Cremers, D., Leal-Taix, L. : Deepsetnet : 심층 신경망을 가진 세트 예측.에서 : ICCV (2017)

Salvador, A., Bellver, M., Baradad, M., Marqu´es, F., Torres, J., Giro´, x. : 시맨틱 인스턴스 세분화를위한 재발 신경 네트워크.ARXIV : 1712.00617 (2017)

Synnaeve, G., Xu, Q., Kahn, J., Grave, E., Likhomanenko, T., Pratap, V., Sriram, A., Liptchinsky, V., Collobert, R. : End-to-EndASR : 현대 건축과의 감독에서 반 감독 학습에 이르기까지.ARXIV : 1911.08460 (2019)

Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z. : 적응 훈련 샘플 선택을 통한 앵커 기반과 앵커 프리 검출 사이의 간격을 연결합니다.ARXIV : 1912.02424 (2019)

우리 모델은 변압기 아키텍처를 기반으로하기 때문에 여기서 우리는 정복에 사용하는 일반적인주의 메커니즘을 상기시킵니다.주의 메커니즘은 [47]를 따른다 [47].

다중 헤드 M 차원 헤드와 함께 다중 헤드주의의 일반적인 형태의 일반적인 형태 D는 다음 시그니처를 갖는 기능입니다 (Underbrace에서 매트릭스/텐서 크기 사용)

여기서 XQ는 길이 NQ의 쿼리 시퀀스 인 경우 XKV는 길이 NKV의 키 값 시퀀스입니다 (박람회의 단순성을위한 동일한 수의 채널 D는 소위 쿼리, 키 및 값을 계산하는 중량 텐서입니다.임베딩 및 L은 투영 행렬입니다.출력은 쿼리 시퀀스와 동일합니다.세부 사항을 제공하기 전에 어휘를 고치려면 다중 헤드 자체 변환 (MH-S-Attn)은 특별한 경우 XQ = XKV입니다.

다중 헤드주의는 단순히 M 단일주의 헤드의 연결과 L과의 투영입니다. 일반적인 관행 [47]은 잔류 연결, 드롭 아웃 및 층 정규화를 사용하는 것입니다.다시 말해, x ~ q = mh-attn (xq, xkv, t, l) 및 x¯¯ (Q)를 나타내는주의 헤드의 연결을 나타냅니다.

ATTN (XQ, XKV, T0)으로 표시되는 중량 텐서 T0 ∈ R3 × D0 × D를 갖는 단일 헤드는 PQ ∈ RD × NQ 및 PKV ∈ RD × NKV에 의존합니다.쿼리 및 키 위치 인코딩을 추가 한 후 소위 쿼리, 키 및 가치 임베딩을 계산하는 것으로 시작합니다 [7] :

여기서 T0은 T10, T20, T30의 연결이다.그런 다음주의 웨이트 α는 쿼리와 키 사이의 DOT 제품의 SoftMax를 기반으로 계산되므로 쿼리 시퀀스의 각 요소가 키 값 시퀀스의 모든 요소에 참석할 수 있습니다 (I Query Index 및 J Key-Value Index).:

우리의 경우, 위치 인코딩은 학습 또는 고정 될 수 있지만, 주어진 쿼리/키 값 시퀀스에 대해 모든주의 레이어에서 공유되므로주의의 매개 변수로 명시 적으로 쓰지 않습니다.인코더와 디코더를 설명 할 때 정확한 가치에 대한 자세한 내용을 제공합니다.최종 출력은주의 가중에 의해 가중 된 값의 집계입니다. I-th Row는 ATTN에 의해 ​​제공됩니다.

FFN (Feed-Forward Network) 레이어 원래 변압기는 다중 헤드주의와 소위 FFN 층을 번갈아 가며 [47],이 경우 MD 입력 및 출력 채널을 효과적으로 다층 1x1 컨볼 루션이다.우리가 고려하는 FFN은 2 층의 1x1 컨볼 루션으로 구성되어 있습니다.방정식 6과 유사하게 두 층 후 잔류 연결/드롭 아웃/레이어 노르르가 있습니다.

완전성을 위해, 우리는 접근 방식에 사용 된 손실을 자세히 제시합니다.모든 손실은 배치 내부의 물체 수에 의해 정규화됩니다.분산 교육을 위해 추가주의를 기울여야합니다. 각 GPU는 서브 배치를 받기 때문에 일반적으로 서브 배치가 GPU에 걸쳐 균형을 이루지 않기 때문에 로컬 배치의 물체 수에 의해 정규화되는 것만으로는 충분하지 않습니다.대신, 모든 하위 배치의 총 객체 수에 의해 정규화하는 것이 중요합니다.

|. |"면적"을 의미하며 박스 좌표의 노조와 ​​교차점은 상자 자체의 속기로 사용됩니다.노조 또는 교차로 영역은 Bσ (i) 및 ˆbi의 선형 함수의 최소/최대에 의해 계산되며, 이는 확률 론적 구배에 대해 손실을 충분히 잘 행동하게 만듭니다.B (bσ (i), ˆbi)는 bσ (i), ˆbi를 포함하는 가장 큰 상자를 의미합니다 (b를 포함하는 영역은 또한 상자 좌표의 선형 함수의 최소/최대에 기초하여 계산됩니다).

주사위/F-1 손실 [28] 주사위 계수는 연합의 교차점과 밀접한 관련이 있습니다.Raw Mask Logits의 모델 예측과 이진 대상 마스크를 M으로 표시하면 손실은 다음과 같이 정의됩니다.

모든주의 층에서 위치 인코딩이 전달 된 DETR에 사용 된 변압기에 대한 자세한 설명은도 10에 나와있다.모든 멀티 헤드 자체 변환 레이어의 키.그런 다음 디코더는 쿼리 (처음에는 0으로 설정됨), 출력 위치 인코딩 (객체 쿼리) 및 인코더 메모리를 수신하고 여러 멀티 헤드 자체 변환 및 디코더 에코더주의를 통해 예측 된 클래스 레이블 및 경계 상자의 최종 세트를 생성합니다.첫 번째 디코더 층의 첫 번째 자체 변환 층을 건너 뛸 수 있습니다.

계산 복잡성 엔코더의 모든 자체 변환은 복잡성 O (D2HW+D (HW) 2) 를가집니다. : O (D0D)는 단일 쿼리/키/값 임베딩 (및 MD0 = D)을 계산하는 비용입니다.(HW) 2)는 한 헤드의주의 웨이트를 계산하는 데 드는 비용입니다.다른 계산은 무시할 수 있습니다.디코더에서, 각각의 자체 소지는 O (d2n +dn2)에 있고, 인코더와 디코더 사이의 교차 항소는 O (d2 (n +hw) +dnhw)에 있으며, 이는 실제로 이후 인코더보다 훨씬 낮습니다.

플롭 계산 더 빠른 R-CNN에 대한 플롭이 이미지의 제안 수에 따라 달라지면 CoCO 2017 유효성 검사 세트에서 처음 100 개의 이미지에 대한 평균 플롭 수를보고합니다.Detectron2 [50]의 도구 플롭 카운트 연산자로 플롭을 계산합니다.Detectron2 모델의 수정없이 사용하고 Detr 모델을 고려하여 BMM (Batch Matrix Multiply)을 가져 오도록 확장합니다.

우리는 ADAMW [26]를 사용하여 DETR을 개선하여 무게 붕괴 처리를 10-4로 설정합니다.또한 최대 구배 규범이 0.1 인 구배 클리핑을 적용합니다.백본과 변압기는 약간 다르게 처리되며 이제 두 가지 세부 사항에 대해 논의합니다.

백본 ImageNet 사전 배송 된 백본 RESNET-50은 TorchVision에서 가져와 마지막 분류 계층을 폐기합니다.백본 배치 정규화 가중치 및 통계는 객체 감지에서 널리 채택 된 실습에 따라 훈련 중에 얼어 붙습니다.학습 속도 10-5를 사용하여 백본을 미세 조정합니다.우리는 백본 학습 속도가 나머지 네트워크보다 대략 적은 순서를 갖는 것이 훈련을 안정화시키는 데 특히 중요하다는 것을 관찰합니다.

변압기 우리는 10-4의 학습 속도로 변압기를 훈련시킵니다.층 정규화 전에 모든 다중 헤드주의 및 FFN 후에 0.1의 첨가제 드롭 아웃이 적용된다.가중치는 Xavier 초기화로 무작위로 초기화됩니다.

손실 우리는 각각 λL1 = 5 및 λiou = 2 무게와 경계 박스 회귀를 위해`1과 giou 손실의 선형 조합을 사용합니다.모든 모델은 n = 100 디코더 쿼리 슬롯으로 훈련되었습니다.

기준선 우리의 향상된 RCNN+ 기준선은 경계 박스 회귀의 표준`1 손실과 함께 GIOU [38] 손실을 사용합니다.우리는 손실에 대한 최상의 가중치를 찾기 위해 그리드 검색을 수행했으며 최종 모델은 각각 상자 및 제안 회귀 작업에 대해 Weights 20과 1의 GIOU 손실 만 사용합니다.기준선의 경우 DETR에 사용 된 것과 동일한 데이터 증강을 채택하여 9 × 일정 (약 109 개의 에포크)으로 훈련합니다.다른 모든 설정은 Detectron2 모델 동물원의 동일한 모델과 동일합니다 [50].

공간 위치 인코딩 인코딩 활성화는 이미지 기능의 상응하는 공간 위치와 관련이 있습니다.우리의 모델에서 우리는 고정 된 절대 인코딩을 사용하여 이러한 공간 위치를 나타냅니다.우리는 원래 변압기 [47] 인코딩의 일반화를 2D 사례 [31]에 채택합니다.구체적으로, 각각의 임베딩의 공간 좌표 모두에 대해 우리는 다른 주파수를 가진 사인 및 코사인 함수를 독립적으로 사용한다.그런 다음 최종 D 채널 위치 인코딩을 얻기 위해 동의합니다.

DETR은 설계별로 인스턴스 수를 늘리면 실험에서 쿼리 슬롯이있는 것보다 더 많은 객체를 예측할 수 없습니다.이 섹션에서는이 한계에 접근 할 때 DETR의 동작을 분석합니다.주어진 클래스의 표준 정사각형 이미지를 선택하고 10 × 10 그리드에서 반복하고 모델에서 누락 된 인스턴스의 백분율을 계산합니다.100 인스턴스 미만의 모델을 테스트하기 위해 일부 셀을 무작위로 마스킹합니다.이렇게하면 객체의 절대 크기가 얼마나 많이 보이든 상관없이 동일합니다.마스킹의 무작위성을 설명하기 위해, 우리는 실험을 다른 마스크로 100 번 반복합니다.결과는 그림 12에 나와 있습니다.동작은 클래스 전체에서 비슷하며 모델은 최대 50 개가 보이면 모든 인스턴스를 감지하지만 포화가 시작되고 점점 더 많은 인스턴스가 누락됩니다.특히, 이미지에 100 인스턴스가 모두 포함되어있을 때, 모델은 평균적으로 30 만 감지하는데, 이는 이미지에 모두 감지 된 50 개의 인스턴스 만 포함 된 경우보다 적습니다.이미지와 탐지가 훈련 분포와는 거리가 멀기 때문에 모델의 반 직관적 인 동작은 가능성이 높습니다.

이 테스트는 단일 클래스의 인스턴스가 많은 예제 이미지가 거의 없기 때문에 설계별로의 일반화 외 분포 테스트입니다.실험에서 두 가지 유형의 도메인 외 일반화 : 이미지 자체와 클래스 당 객체 수를 풀기가 어렵습니다.그러나 Coco 이미지가 거의 없거나없는 것과 동일한 클래스의 많은 객체 만 포함하기 때문에,이 유형의 실험은 쿼리 객체가 데이터 세트의 레이블과 위치 분포에 과적되는지 여부를 이해하기위한 최선의 노력을 나타냅니다.전반적으로, 실험은 모델이 최대 50 개의 객체의 거의 완벽한 탐지를 생성하기 때문에 이러한 분포에 과잉이되지 않음을 시사합니다.

그림 12 : 이미지에 존재하는 수에 따라 DETR이 누락 된 다양한 클래스의 인스턴스 수 분석.우리는 평균과 표준 편차를보고합니다.인스턴스 수가 100에 가까워지면 DETR은 포화를 시작하고 점점 더 많은 객체를 놓치고 있습니다.

접근 방식의 단순성을 입증하기 위해 Listing 1에는 Pytorch 및 Torchvision 라이브러리와의 추론 코드를 포함합니다. 코드는 Python 3.6+, Pytorch 1.4 및 Torchvision 0.5로 실행됩니다.배치를 지원하지 않으므로 GPU 당 하나의 이미지를 갖춘 분산 DataparAllel을 사용한 추론 또는 훈련에만 적합합니다.또한 명확성을 위해이 코드는 고정 대신 인코더에서 학습 된 위치 인코딩을 사용하고 위치 인코딩은 각 변압기 층 대신 입력에 추가됩니다.이러한 변화를 만들려면 변압기의 Pytorch 구현을 넘어서서 가독성을 방해해야합니다.실험을 재현하기위한 전체 코드는 회의 전에 제공됩니다.

목록 1 : Detr Pytorch 추론 코드.명확성을 위해 그것은 고정 대신 인코더에서 학습 된 위치 인코딩을 사용하고, 위치 인코딩은 각 변압기 층 대신 입력에 추가됩니다.이러한 변화를 만들려면 변압기의 Pytorch 구현을 넘어서서 가독성을 방해해야합니다.실험을 재현하기위한 전체 코드는 회의 전에 제공됩니다.