DETR

Detection Transformer

2025.02.20

Seung min chung

Contents

01 Abstract

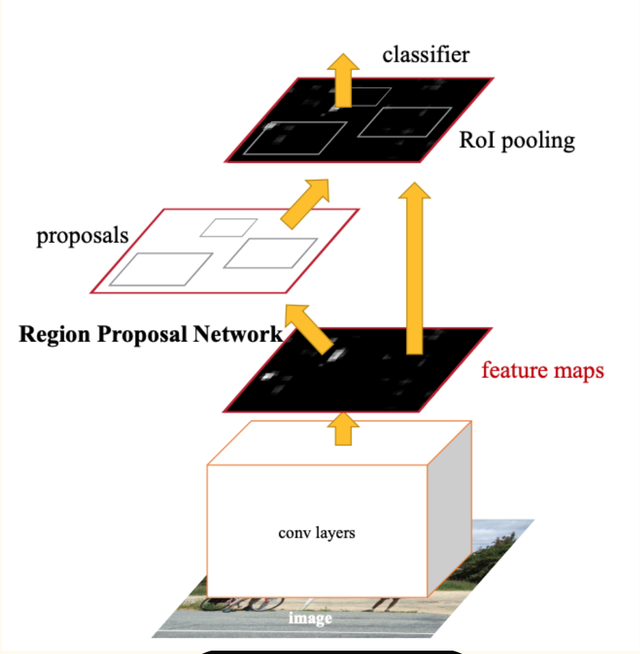
핵심 아이디어

객체 탐지 문제를 여러 단계의 복잡한 파이프라인으로 구성하지 않고 set prediction 이라는 관점을 도입하여 하나의 End-to-End 파이프라인으로 높은 성능의 객체 탐지를 가능하도록 하였습니다.

핵심

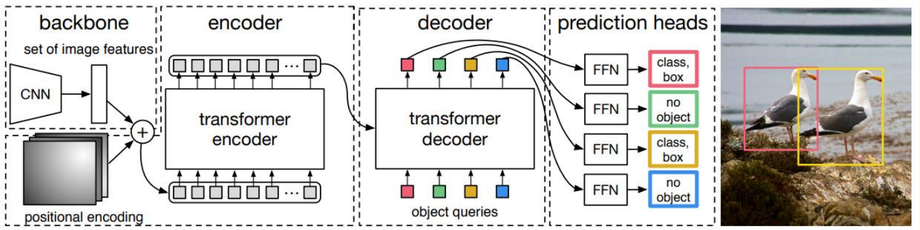
- Set prediction
- Transformer Encoder Decoder architecture

02 Introduction



기존

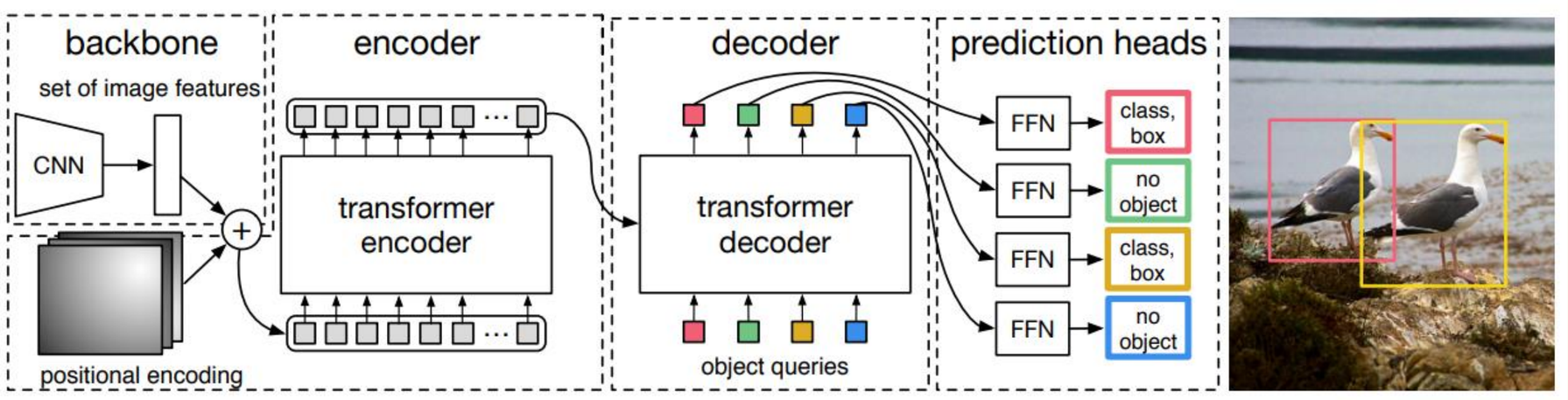
기존의 객체 탐지 파이프라인은 RPN층이 존재하여
객체가 있을 법한 위치를 추출하고, NMS를 통해서
중복을 제거한 후 각각의 박스와 레이블을 학습한다.



DETR

CNN 백본과 transformer 아키텍처를 활용하여 이미지를
넣으면 자동으로 박스와 레이블을 추출하도록 유도한다.

02 Introduction



Bipartite matching

이분 매칭을 활용합니다. 이분 매칭을 활용하는 이유는 서로 Unique한 쌍을 만들어서 중복되는 값을 갖지 않도록 유도하기 위함입니다. 이를 통해서 NMS와 같은 postprocessing이 필요없게 됩니다.

Transformer

Transformer의 아키텍처를 활용하여 픽셀 간의 상호작용을 통해서 전체적인 문맥파악에 도움을 주게 됩니다. 이를 통해서 높은 성능을 제공합니다.

03 set prediction loss

Set prediction loss – Matching loss

DETR에서는 총 N개의 객체를 예측하도록 합니다. 이때 N은 실제 객체보다 훨씬 큰 수로 설정됩니다.

$$\sigma = \arg\min_{\sigma \in S_N} \sum_{i=1}^{N} L_{\text{match}}(y_i, \hat{y}_{\sigma(i)})$$

다음과 같은 함수를 이용하여 총 N개에 대해서 예측값과 실제값의 손실을 최소화하는 쌍을 찾게 됩니다. 매칭의 경우 “헝가리안 알고리즘”을 활용하고, 모든 조합을 고려했을 때 가장 손실이 적도록 쌍을 매칭시킨다 (이분 매칭)

$$y_i = (c_i, b_i)$$

$$(b_i \in [0,1]^4)$$

각각의 y는 다음과 같이 클래스에 대한 확률, 그리고 박스에 대한 좌표 정보가 담겨있습니다. 각 박스의 경우 스케일링이 적용되어 모든 값이 0 ~ 1 사이의 값을 갖도록 구성됩니다.

Matching Loss에 대해서는 다음과 같이 객체인 경우에만 정의를 진행하게 됩니다.
만일 객체가 아닌 경우 손실을 특정 상수로 특정하게 됩니다. 객체가 아닌 경우 단순히 상수 값을 주어 이분 매칭시 일관된 매칭이 성사되도록 유도한다고 합니다.

$$L_{\text{match}}(y_i, \hat{y}\sigma(i)) = -1\{c_i \neq \varnothing\}\hat{p}\sigma(i)(c_i) + 1\{c_i \neq \varnothing\}L_{\text{box}}(b_i, b_{\sigma(i)})$$

기존의 box loss에서는 박스의 크기에 따라서 절대적인 손실값의 차이가 존재하기에 GIOU라는 박스의 픽셀도 고려하는 값을 추가하여 박스 크기에 상관없이 균형있는 손실을 제공할 수 있는 아래와 같은 식을 사용한다고 합니다 .

$$L_{\text{box}}(b_i, b_{\sigma(i)}) = \lambda_{\text{iou}} L_{\text{iou}}(b_i, b_{\sigma(i)}) + \lambda_{L1} \|b_i - b_{\sigma(i)}\|_1$$

03 set prediction loss

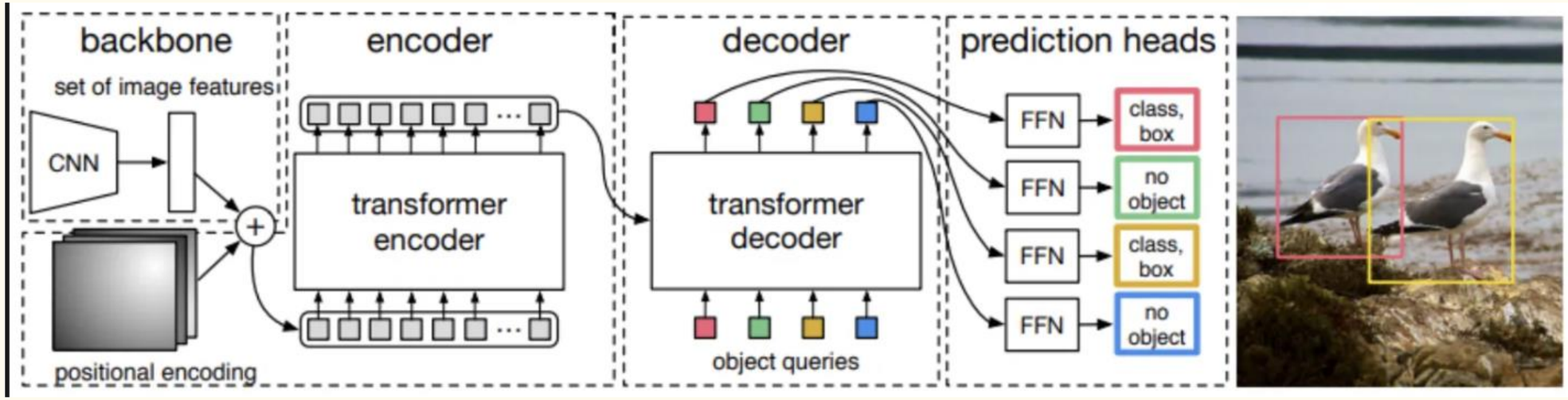
Set prediction loss – Matching loss

$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + 1\{c_i \neq \varnothing\} L_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}(i)}) \right]$$

위의 matching loss를 통해서 전체적인 손실을 최소화 하는 매칭을
찾았다면 이제 해당 클래스와 박스의 좌표가 GT와 비슷하도록
유도해야합니다. 이때 객체가 아닌 박스에 대해서는 Box loss를 주지 않아
실제 객체가 있는 경우에만 box loss에 집중할수 있도록 강제한다고 합니다.

실제 이전의 방법론에서는 박스의 좌표 자체를 에측하는 것이 아니라
offset을 에측 하도록 하였지만 DETR에서는 위의 LOSS들을 사용함으로 써
박스의 좌표를 직접 예측하도록 유도하였다고 합니다.

04 DETR architecture



CNN based Backbone

CNN 기반의 모델을 backbone으로 활용하여 Feature map을 추출하여 Transformer의 입력으로 사용합니다.

Transformer

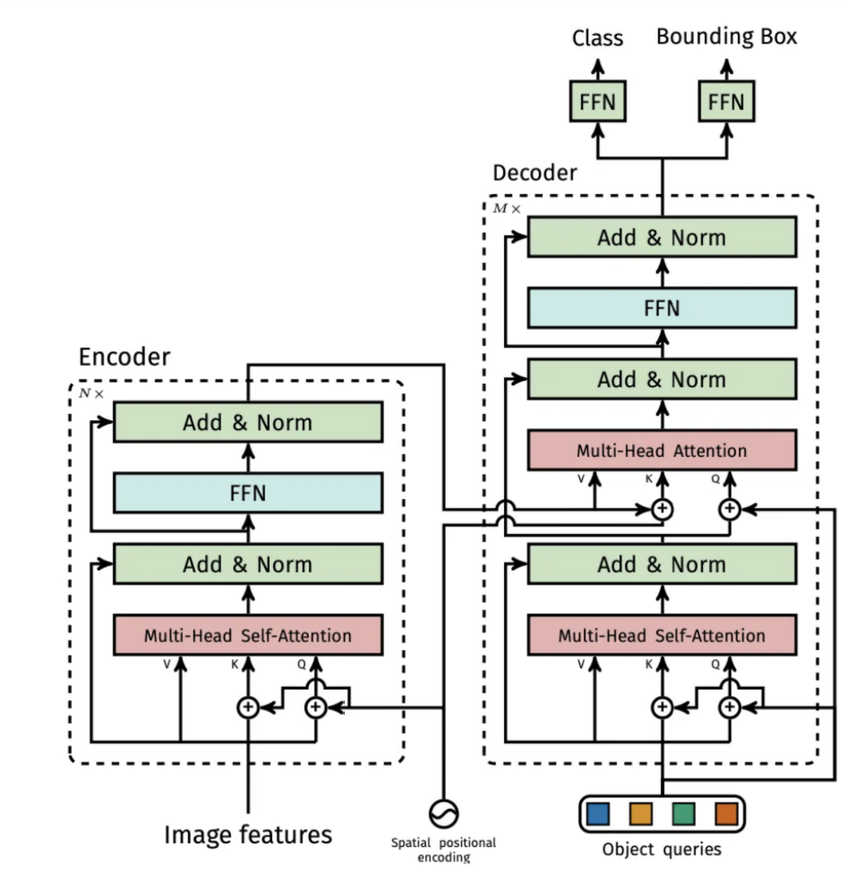
Transformer의 Encoder를 통해서 이미지의 전체적인 문맥을 파악하고 Decoder에서는 N개의 object queries를 활용하여 N개의 instance를 에측합니다.

FFN

공유되는 FFNs을 사용하여 decoder의 결과로 부터 박스의 좌표와 클래스를 예측하게 됩니다.

04 DETR architecture

Transformer



- 객체 탐지의 경우 autoregressive 방식의 접근이 필요 없기 때문에 Decoder의 첫 attention module이 masked attention이 아니라 그냥 multi-head attention을 사용하고 있습니다.

- Object queries의 경우 (d x N) 크기의 행렬이며 이들은 서로 다른 값을 갖는 랜덤으로 초기화된 값들입니다. Autoregressive하지 않고 병렬적으로 처리되다 보니 만일 동일한 값이 존재하는 경우 결과도 동일하게 나오게 됩니다. 그리고 각각의 query는 하나의 instance를 담당하게 됩니다.

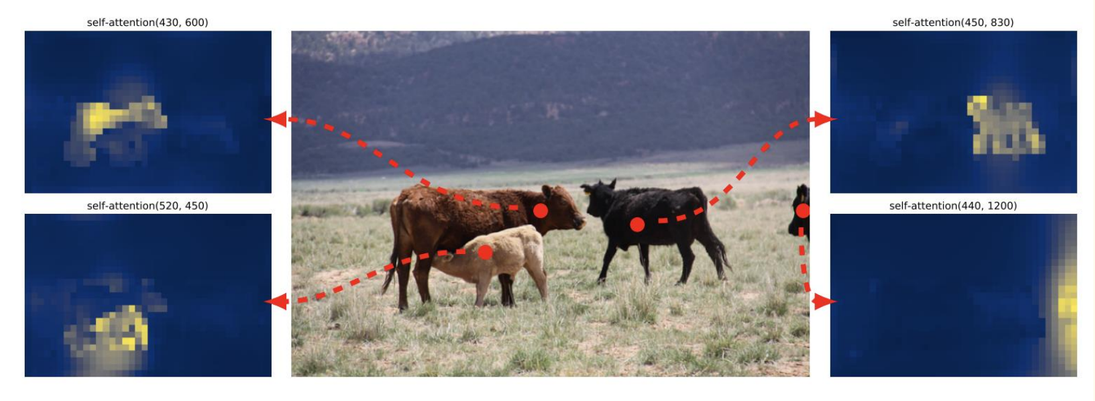
- Auxiliary decoding loss를 사용하여 해당 모델의 성능을 향상 시켰다고 합니다. Auxiliary의 경우 각 decode moudle 하나가 끝날 때 마다 prediction FFN과 Loss를 사용하여 decoder 모듈 하나도 하나의 에측을 진행하여 보다 빠르게 수렴을 할 수 있도록 도와준다고 합니다.

06 Experiments

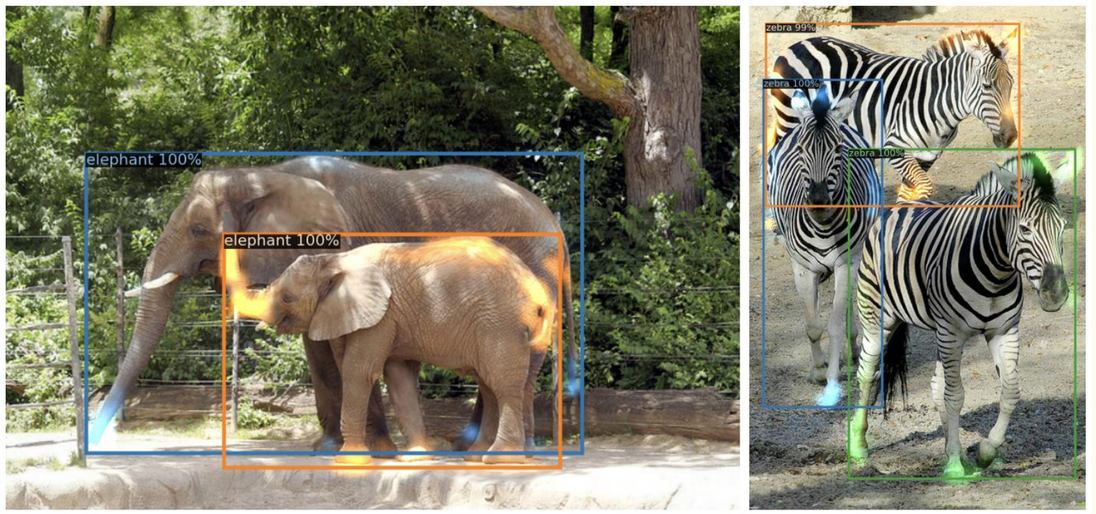
기존의 Object detection model들과의 성능 비교

Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-DC5+	320/16	166M	41.1	61.4	44.3	22.9	45.9	55.0
Faster RCNN-FPN+	180/26	42M	42.0	62.1	45.5	26.6	45.4	53.4
Faster RCNN-R101-FPN+	246/20	60M	44.0	63.9	**47.8**	**27.2**	48.1	56.0
DETR	86/28	41M	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	187/12	41M	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101	152/20	60M	43.5	63.8	46.4	21.9	48.0	61.8
DETR-DC5-R101	253/10	60M	**44.9**	**64.7**	47.7	23.7	**49.5**	**62.3**

Encoder activate map : 각각의 instance를 잘 포착한다



Decoder 시각화 : 특징과 박스를 찾기 위해 테두리쪽에 강하게 작용한다.



Encoder 깊이에 따른 성능 비교

#layers	GFLOPS/FPS	#params	AP	AP_{50}	AP_S	AP_M	AP_L
0	76/28	33.4M	36.7	57.4	16.8	39.6	54.2
3	81/25	37.4M	40.1	60.6	18.5	43.8	58.6
6	86/23	41.3M	40.6	61.6	19.9	44.3	60.2
12	95/20	49.2M	41.6	62.1	19.8	44.9	61.9

07 Summary

이분 매칭

객체 탐지 문제를 이분 매칭을 통해 집합 예측 문제로 재정의함

Transformer

Transformer를 이용하여 매우 간단한 파이프라인을 갖는다.

확장 가능성

또 다른 Transformer기반의 모델들에 대해서 확장 가능하며, 다른 task에 대해서도 잘 작동한다.

DETR의 경우 객체 탐지 분야에 Transformer를 적용함으로 써 복잡한 파이프라인을 간단한 구조로 변경하여 높은 정확도를 보여준 모델입니다. DETR의 가장 핵심 관점은 바로 객체 탐지를 이분 매칭 예측 문제로 치환한 것이라고 생각합니다. 그리고 이를 학습할 수 있는 Matching Loss, Hungarian Loss를 도입하였다는 것입니다. 이를 통해서 매우 간단한 구조를 통해서 객체 탐지를 End-to-End로 구현할 수 있게 되었습니다.

Thanks