# End-to-End Object Detection with Transformers

https://arxiv.org/abs/2005.12872

## 0. Introduction

- 객체 검출은 이미지 안의 물체 위치와 종류를 찾아내는 핵심 비전 과제임.
- 기존 방법(Faster R-CNN 등)은 앵커(anchor) 설계, NMS(비최대 억제) 같은 복잡한 후처리가 필수였음.
- 이 논문(DETR)은 그런 복잡한 구성 없이 Transformer 기반 단일 모델로 객체 검출을 수행하려는 시도임.
- 핵심 아이디어는 “객체 검출을 집합(set) 예측 문제로 바꾸자”는 것.
- 즉, 모델이 미리 정한 개수의 예측을 내고, 실제 객체와 일대일로 매칭하는 방식으로 단순화함.
- 결과적으로 후처리 없는 엔드투엔드 객체 검출기를 제시했고, 기존 R-CNN 계열과 비슷한 성능을 달성함.

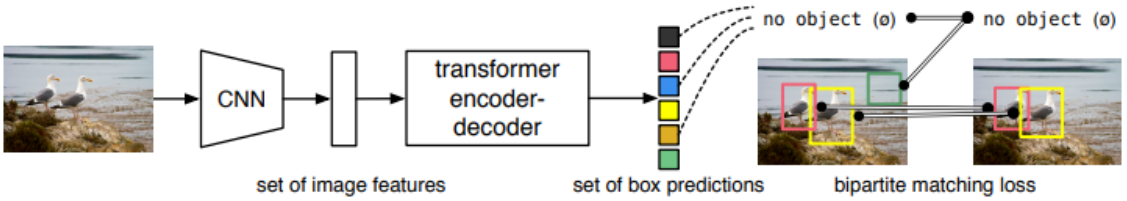
## 1. Overview

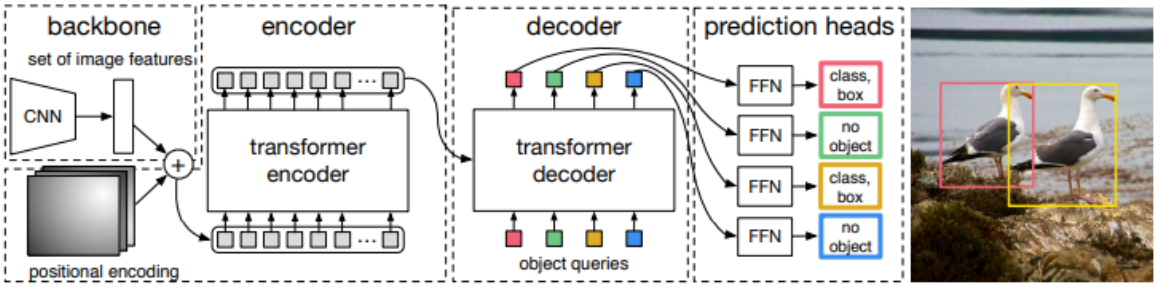
- DETR은 CNN 백본으로 특징을 뽑고, Transformer 인코더-디코더 구조로 각 객체를 직접 예측함.
- 디코더의 “object queries”가 각각 하나의 객체를 담당하며, 예측 결과는 클래스와 박스 좌표로 구성됨.
- 예측된 객체들과 실제 객체는 Hungarian 매칭 알고리즘으로 일대일 대응시켜 loss 계산함.
- 후처리(NMS)나 앵커 설정 없이, 학습만으로 중복 없는 예측을 학습할 수 있음.
- COCO 데이터셋에서 기존 검출기 수준의 성능을 보이며, 구조가 단순하고 깔끔한 게 가장 큰 장점임.

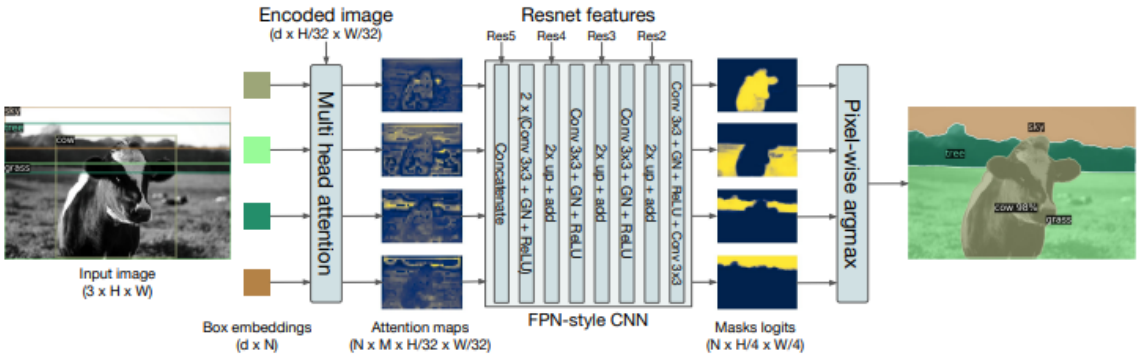
## 2. Challenges

- 수렴 속도: Transformer 구조 특성상 훈련 초기 수렴이 느림
- 소형 객체 검출 성능: 저해상도 또는 작은 객체에 대해서는 기존 방법 대비 약점 존재
- 고정 예측 수의 한계: N이 너무 크면 불필요한 연산, 너무 작으면 객체 누락 가능성
- 연산 복잡도: Transformer self-attention의 쿼리/키 매칭 연산 비용 증가
- 위치 인코딩의 한계: Transformer가 공간 정보를 잃지 않도록 위치 인코딩 설계 민감
- 추가 후처리 제거로 인한 유연성 저하 가능성: NMS 등 알고리즘이 없기 때문에 중복 예측 제어나 예외 경우 처리 어려움
- 기존 방법들과의 비교: 기존 앵커 기반 접근, NMS 후처리 방식 등이 가진 강점을 포기하면서 얻는 손실 위험

## 3. Method






- 모델 아키텍처
- 백본 + 피처 맵 생성
  - ResNet (50 또는 101) 사용, 마지막 레이어의 피처 맵을 추출
  - 채널 축소를 위한 1×1 컨볼루션 적용
  - 공간 → 시퀀스 변환 (H×W → HW 길이 시퀀스)
  - 위치 인코딩 추가 (sine/cosine 방식)
- Transformer 인코더
  - self-attention + feed-forward 레이어 반복
  - 입력은 피처 + 위치 인코딩
  - 공간 간 전역 문맥 관계 학습
- Transformer 디코더
  - Object queries (학습 가능한 임베딩)의 입력
  - self-attention, cross-attention, feed-forward 계층
  - cross-attention을 통해 인코더의 피처와 연관
  - 쿼리별로 객체 특성 추출
- 출력 헤드 (Prediction heads)
  - 각 디코더 출력(query)에 대해
  - 클래스 소프트맥스 (C+1 클래스, + “no object”)
  - 박스 좌표 예측 (center_x, center_y, width, height) — 정규화된 값
  - 박스 loss는 L1 + generalized IoU 조합 적용
- 매칭 및 Loss 계산

    - Hungarian algorithm으로 예측 집합과 실제 객체 집합 간 최적 매칭
    - 매칭된 쌍에만 loss 부여
    - unmatched 예측은 클래스 “no object”로 처리하여 classification loss만 적용
- 학습 세부 사항
    - 백본(ResNet)는 낮은 학습률 사용
    - Dropout, weight decay, augmentation 적용
    - 예측 개수 N (예: 100) 고정
    - gradient clipping 사용
    - 학습 에폭 수 조정 등 하이퍼파라미터 실험 포함

# 4. Experiments

- 데이터셋 및 설정
    - 사용 데이터셋: COCO 2017 (train / val)
    - 평가 지표: mAP (mean Average Precision), AP50, AP75 등
    - 비교 대상: Faster R-CNN 계열 등
    - 백본: ResNet-50, ResNet-101
    - 예측 수 N = 100 (기본)
    - Ablation study: 쿼리 수 변화, 레이어 수 변화, loss 구성 변화 등
- 주요 실험 및 결과
    - DETR (ResNet-50) → COCO 기준 42 AP 수준 (Faster R-CNN 대비 경쟁력 있음)
    - 다양한 구성 실험
        - 디코더 레이어 개수, 쿼리 수 변화
        - 위치 인코딩 방식 변경
        - loss 요소별 제거 실험
    - 시각화 결과: 객체 예측 샘플 제공
    - 한계 분석: 작은 객체나 복잡한 장면에서 오류 빈도 증가

- 일반화 가능성 실험: panoptic segmentation 확장 모듈 제안 (추가 헤드)
- 비교: DETR은 복잡한 후처리 없이 단순 구조로 동등한 성능을 보인다는 점 강조

# 5. Results

| 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** |

| #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 |

| Model | Backbone | PQ | SQ | RQ | $PQ^{th}$ | $SQ^{th}$ | $RQ^{th}$ | $PQ^{st}$ | $SQ^{st}$ | $RQ^{st}$ | AP |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PanopticFPN++ | R50 | 42.4 | 79.3 | 51.6 | 49.2 | 82.4 | 58.8 | 32.3 | 74.8 | 40.6 | 37.7 |
| UPSnet | R50 | 42.5 | 78.0 | 52.5 | 48.6 | 79.4 | 59.6 | 33.4 | 75.9 | 41.7 | 34.3 |
| UPSnet-M | R50 | 43.0 | 79.1 | 52.8 | 48.9 | 79.7 | 59.7 | 34.1 | 78.2 | 42.3 | 34.3 |
| PanopticFPN++ | R101 | 44.1 | 79.5 | 53.3 | **51.0** | **83.2** | 60.6 | 33.6 | 74.0 | 42.1 | **39.7** |
| DETR | R50 | 43.4 | 79.3 | 53.8 | 48.2 | 79.8 | 59.5 | 36.3 | 78.5 | 45.3 | 31.1 |
| DETR-DC5 | R50 | 44.6 | 79.8 | 55.0 | 49.4 | 80.5 | 60.6 | **37.3** | **78.7** | **46.5** | 31.9 |
| DETR-R101 | R101 | **45.1** | **79.9** | **55.5** | 50.5 | 80.9 | **61.7** | 37.0 | 78.5 | 46.0 | 33.0 |

- 정량적 성능: Faster R-CNN 대비 유사한 mAP 수준
- Ablation 결과: 각 요소의 중요도 확인
  - 예: generalized IoU loss가 성능 향상에 기여 큰 것으로 나타남 Medium+2arXiv+2
  - 쿼리 수, 레이어 수 조정 시 성능 변화 있음

- 작은 객체에 대한 성능은 여전히 기존 방법보다 낮음
- 수렴에 시간 소요됨
- 구조 단순성, 직관성, 후처리 제거 등의 장점 강조
- panoptic segmentation 확장에서는 경쟁력 있는 성능 확보

## 6. Insight

- 이 논문은 객체 검출 문제에 Transformer + 집합 예측(set prediction)이라는 관점을 처음 도입했다는 점에서 매우 중요한 전환점
- 복잡한 수작업 설계 (앵커, NMS 등)를 제거하고 모델을 단순하게 유지하는 접근은 후속 연구에 강한 영향 미침
- 단점인 수렴 속도, 작은 객체 성능은 이후 연구 (예: Deformable DETR, Dynamic DETR 등)에서 개선 시도됨
- 실무 적용 시 고려할 점: 연산 비용, 실시간 요구, 작은 객체 비율이 높은 환경 등