DETR: End-to-End Object Detection with Transformers

School of Industrial and Management Engineering, Korea University

Jin Hyeok Park

KOREA UNIVERSITY

DMQA

Contents

Introduction

Object Detection

- Computer Vision의 연구 분야 중 하나
- Object Detection이란 이미지 내 물체의 위치와 분류하는 방법
- **Object Detection** = Multi-Labeled **Classification** + Bounding Box Regression(**Localization**)

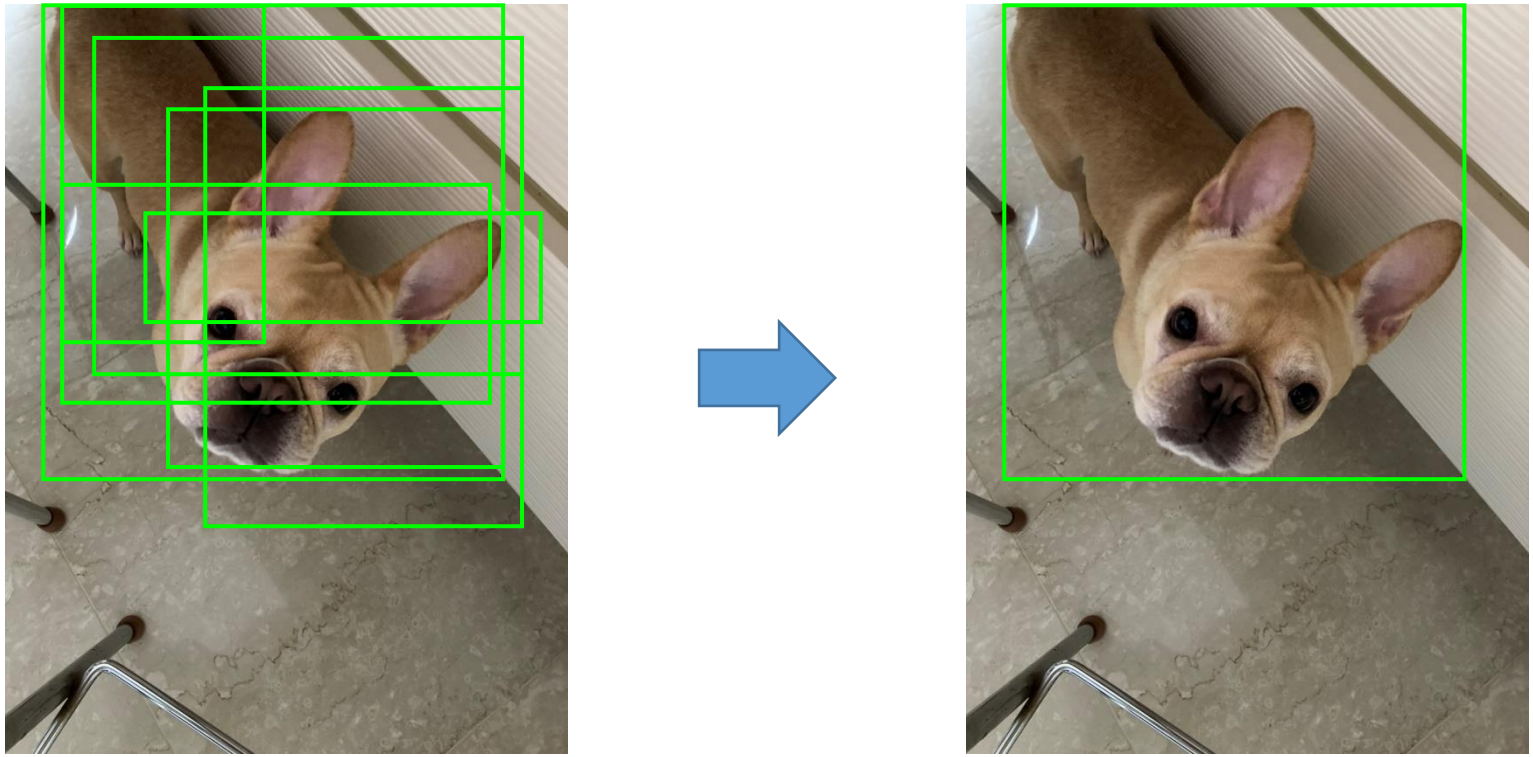
Object Detection

Thor, Captain America, Car

Introduction

Non-maximal suppression(NMS)

- Object detection에서 Object에 Bounding box를 생성함
- 동일한 Object에 여러 개의 Bounding box를 생성하는 경우가 존재함
- Non-maximal suppression은 가장 스코어가 높은 Bounding box만 남김

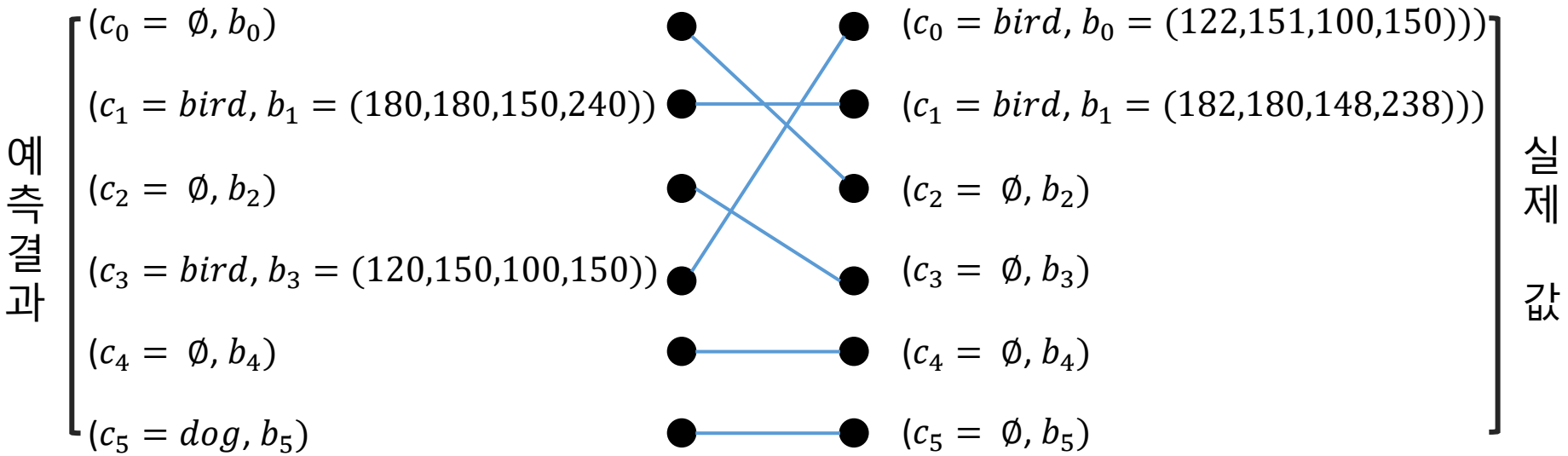


Introduction

Bipartite matching

- ❖ 이분 매칭(bipartite matching)을 통해 set prediction을 직접적으로 해결함
- ❖ 학습 과정에서 이분 매칭을 수행함으로써 인스턴스 중복 방지

출력 개수: N = 6



Overview of DETR

DETR

- Facebook AI
- 2021년 7월 21일 기준 577회 인용
- Transformer를 object detection에 적용한 최초의 연구

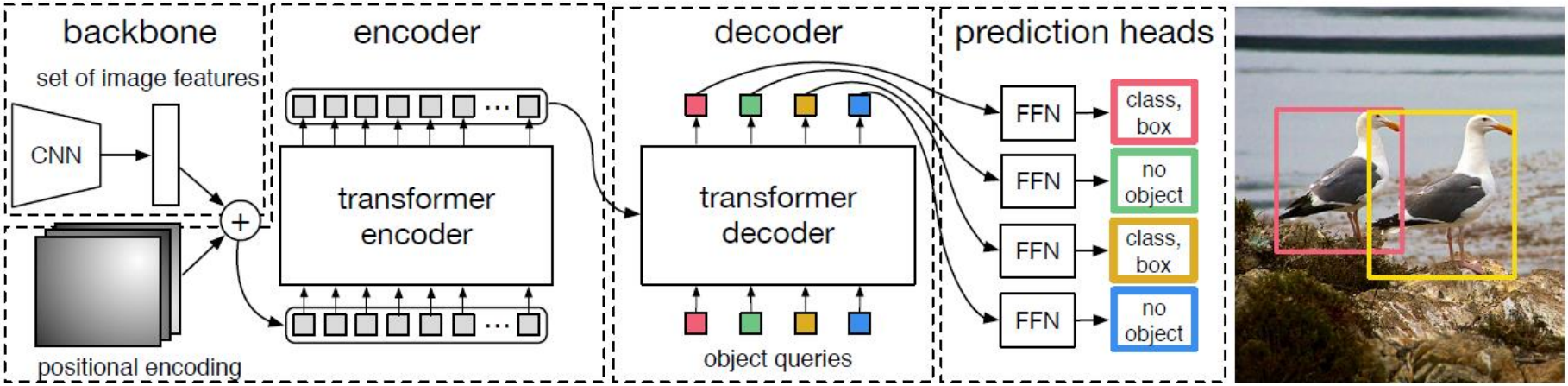
End-to-End Object Detection with Transformers

Nicolas Carion*, Francisco Massa*, Gabriel Synnaeve, Nicolas Usunier,
Alexander Kirillov, and Sergey Zagoruyko

Overview of DETR

DETR

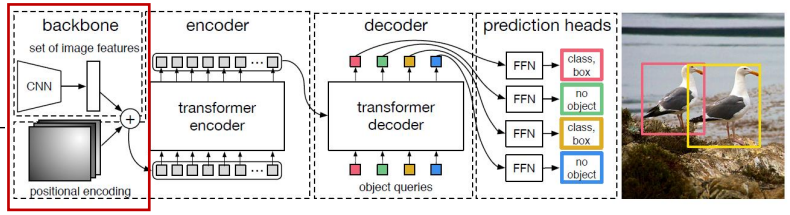
- 기존 object detection 방식은 NMS방식을 적용하거나 anchor box를 생성함
- DETR에서는 NMS방식 또는 anchor box를 생성하는 과정을 제거함
- Transformer와 bipartite matching 기법을 활용한 end-to-end로 결과 출력



<DETR architecture>

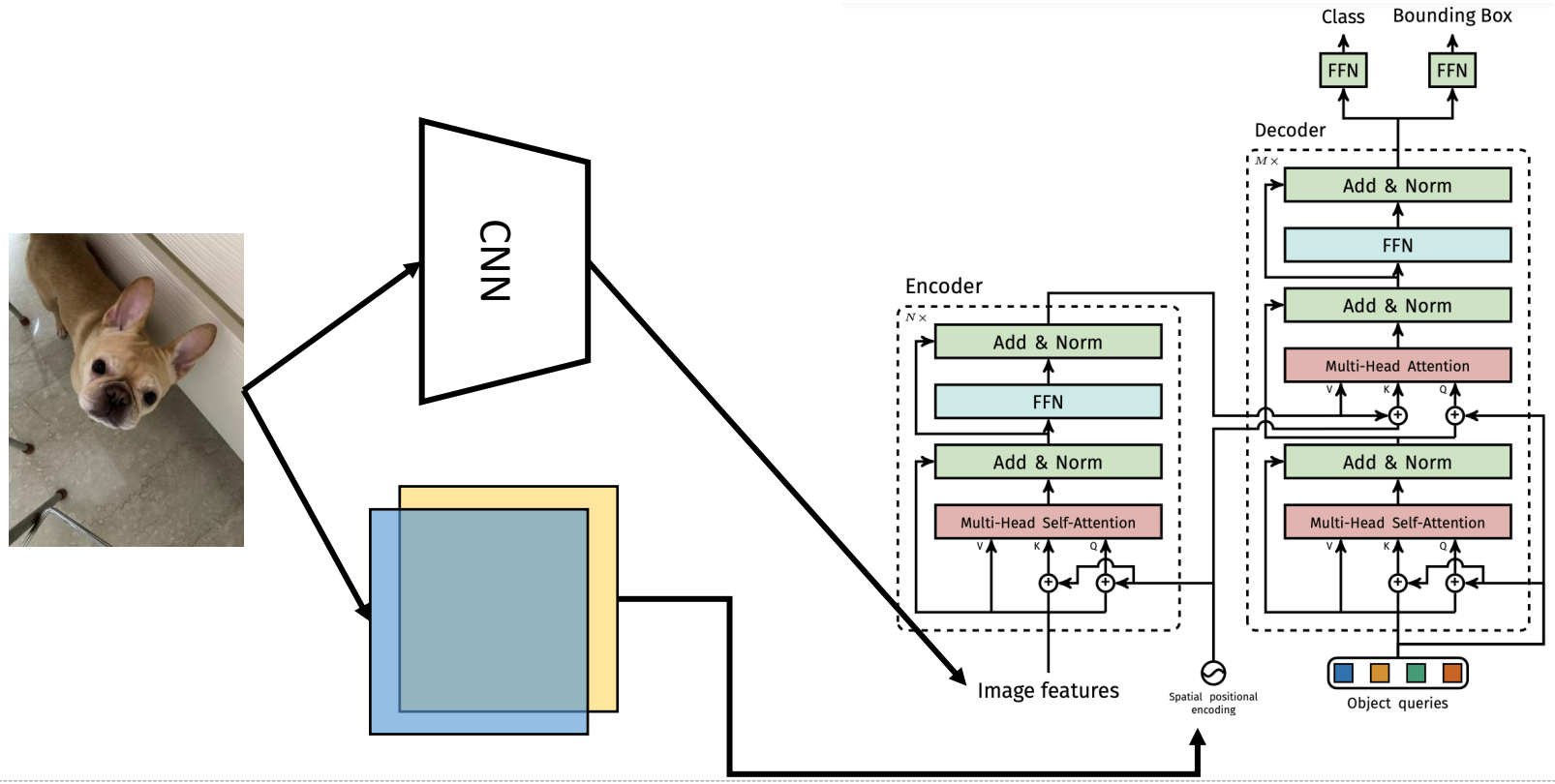
DMQA

DETR Architecture



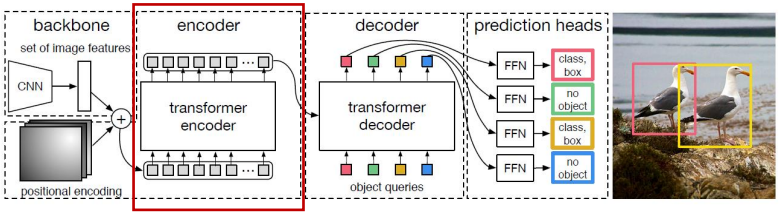
Backbone

- Backbone 구조는 3-color channel image를 input으로 받아 feature extraction(ResNet50, ResNet101)
- Backbone에서 나온 feature map을 1×1 convolution을 통해 channel을 압축함
- Feature map의 spatial domain에 해당하는 부분이 위치 정보를 상실함
- 위치 정보를 위해 position encoding과정을 추가함



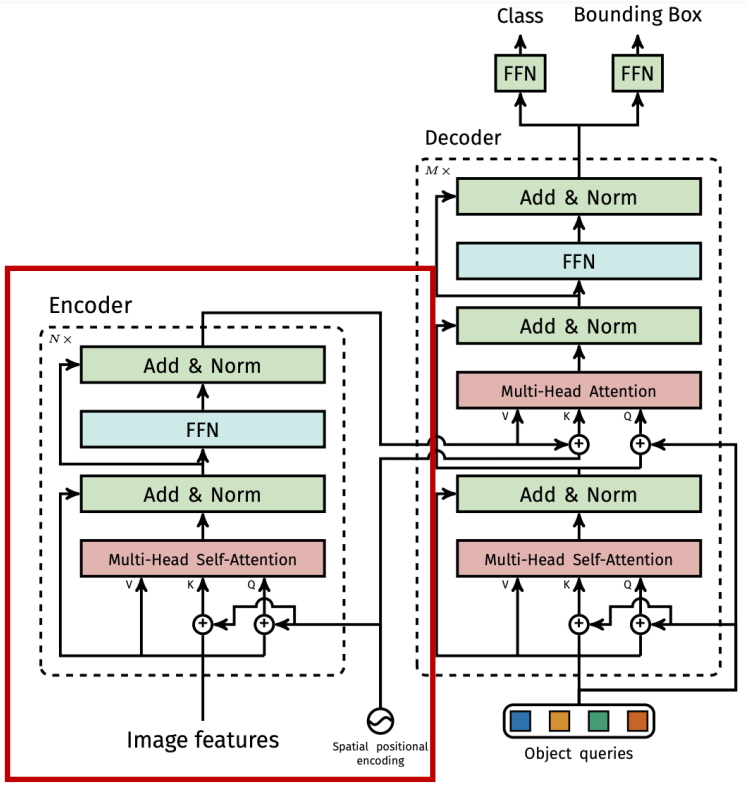
DMQA

DETR Architecture



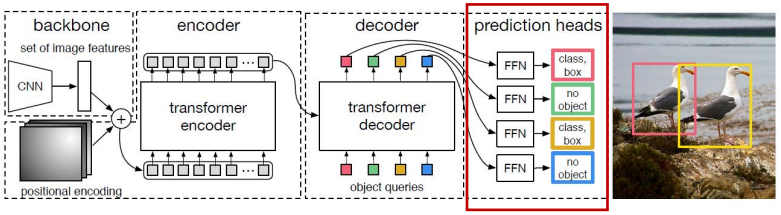
Encoder

- 이미지의 특징(feature) 정보를 포함하고 있는 각 픽셀 위치 데이터를 입력 받아 인코딩 진행
- Backbone에서 추출된 image features를 d-dimension 으로 축소한 뒤 2차원 벡터로 변환
- Transformer는 attention기반이기 때문에 순서를 부여하는 fixed position encoding을 encoder input에 적용



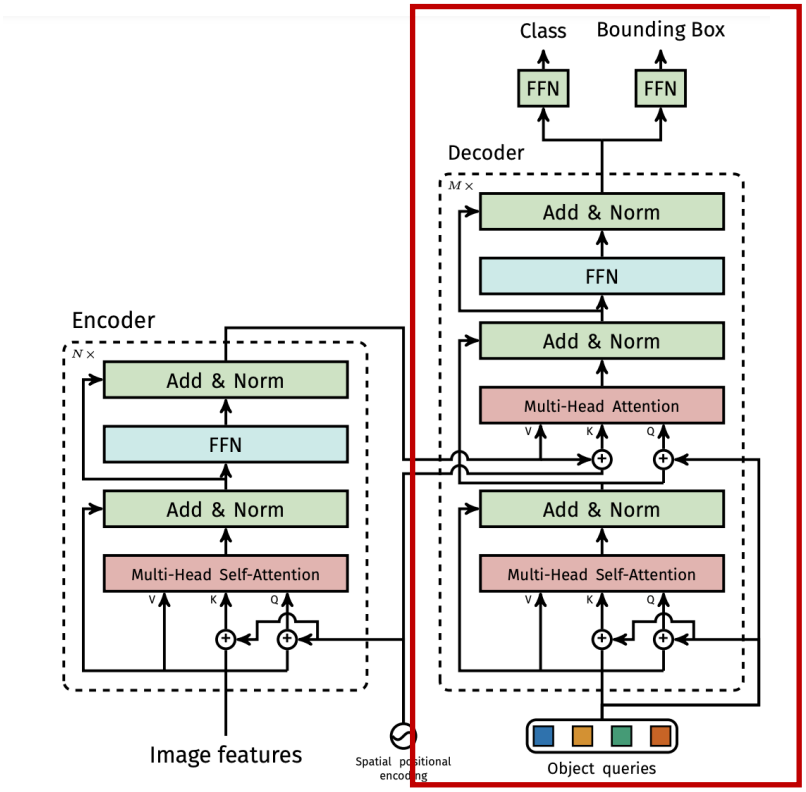
DMQA

DETR Architecture



Decoder

- N개의 object query를 초기 입력으로 받으며 인코딩 된 정보를 활용함
- 각 object query는 이미지 내 서로 다른 고유한 인스턴스를 구별함
- 디코더에서는 각 인스턴스의 클래스와 경계선을 추출함



DMQA

Result

Experiments

- COCO 2017 detection dataset을 가지고 실험 진행
- Average precision을 가지고 정량적 평가 진행
- Object detection model 중 하나인 Faster RCNN과 비교하여 우수한 성능을 보여줌
- DC5: Dilated convolution을 적용하여 실험 진행(넓은 시야 확보)

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**

Conclusion

❖ DETR: End-to-End Object Detection with Transformers

- Non-maximum suppression, anchor generation 없이 detection model을 간소화 시킴
- Object들을 하나의 set으로 두고 ground-truth objects와 bipartite matching을 통해 loss function을 계산
- CNN backbone과 encoder-decoder Transformer만을 사용한 간단한 구조

❖ DETR의 한계점

- 학습시간이 오래 걸리며 computing power가 받쳐 줘야함
- 작은 물체 탐지에 어려움이 존재


Thank You

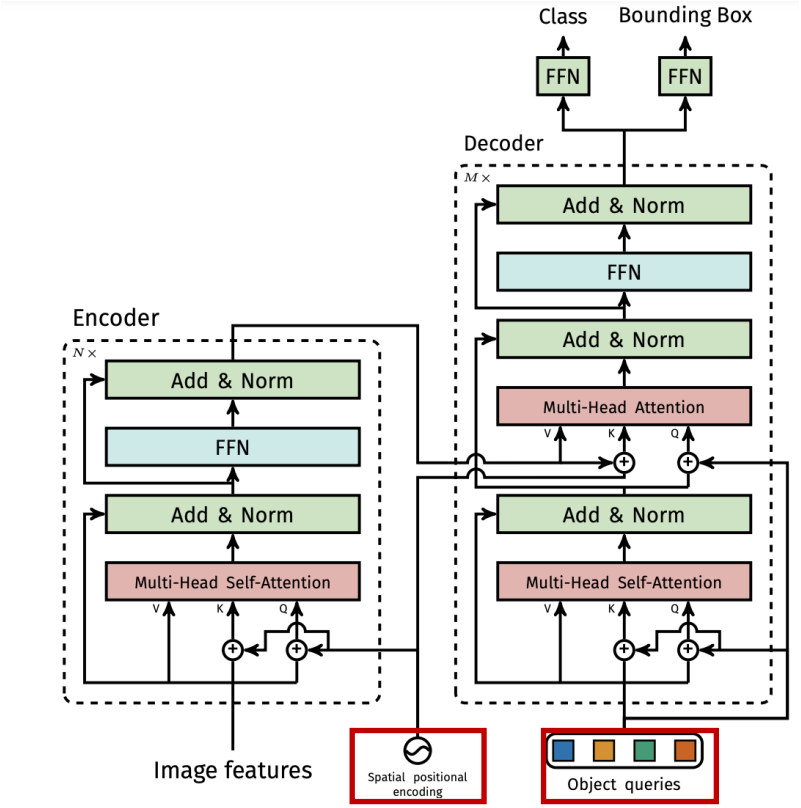
Appendix

Appendix

DETR Architecture

❖ Positional Encodings

- Spatial Position Encoding: 이미지의 위치 정보로부터 순차적 정보 추출
- Object queries: N개의 object query가 Decoder의 input으로 적용



Appendix

DETR Architecture

- $y_i = (c_i, b_i)$, c_i: target class label
- $b_i \in [0,1]^4$: bounding box 좌표
- $\hat{p}_{\sigma(i)}(c_i)$: class c_i에 속할 확률
- σ: 예측 - 정답

❖ Bipartite Matching Loss

- DETR의 decoder는 고정된 사이즈 N개의 object를 예측함
- N개의 object와 ground-truth object간의 optimal bipartite matching이 이루어짐
- L_{match}: 각 쌍의 class와 box에 대한 pair-wise matching cost
- 해당 방법은 one-to-one matching으로 duplicate 방지

$$\hat{\sigma} = \arg\min_{\sigma \in \mathfrak{S}_N} \sum_{i}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)})$$

$$L_{match}(y_i, \hat{y}_{\sigma(i)}) = -1_{\{c_i \neq \emptyset\}} \hat{p}_{\sigma(i)} + 1_{\{c_i \neq \emptyset\}} L_{box}(b_i, \hat{b}_{\sigma(i)})$$

Appendix

DETR Architecture

❖ Bipartite Matching Loss

- 가장 적은 matching cost를 가진 $\hat{\sigma}$를 찾은 후, Hungrian loss를 계산
- Class imbalance를 고려하여 class 예측에 가중치를 두기 위해 negative log-likelihood를 사용
- L_{box}: 예측값과 ground-truth의 차이($L_1\ loss$)와 함께 scale-invariant한 iou계산을 함께 적용

$$L_{Hungrian}(y, \hat{y}) = \sum_{i=1}^{N} [-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + 1_{\{c_i \neq \emptyset\}} L_{box}(b_i, \hat{b}_{\hat{\sigma}}(i))]$$

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