Object Detection Task

EfficientDet
DETR

Artificial Intelligence

Creating the Future

Dong-A University

Division of Computer Engineering &
Artificial Intelligence

References

Main

- https://theaisummer.com/cnn-architectures/

blog Sub

- https://deepbaksuvision.github.io/Modu_ObjectDetection/

PyTorch tutorial

- website : https://pytorch.org/
- Korea website : https://pytorch.kr/

github

- https://github.com/pytorch
- https://github.com/9bow/PyTorch-tutorials-kr
- torchvision : https://github.com/pytorch/vision
- https://github.com/weiaicunzai/pytorch-cifar100

SOTA of CNN Architecture

Part1

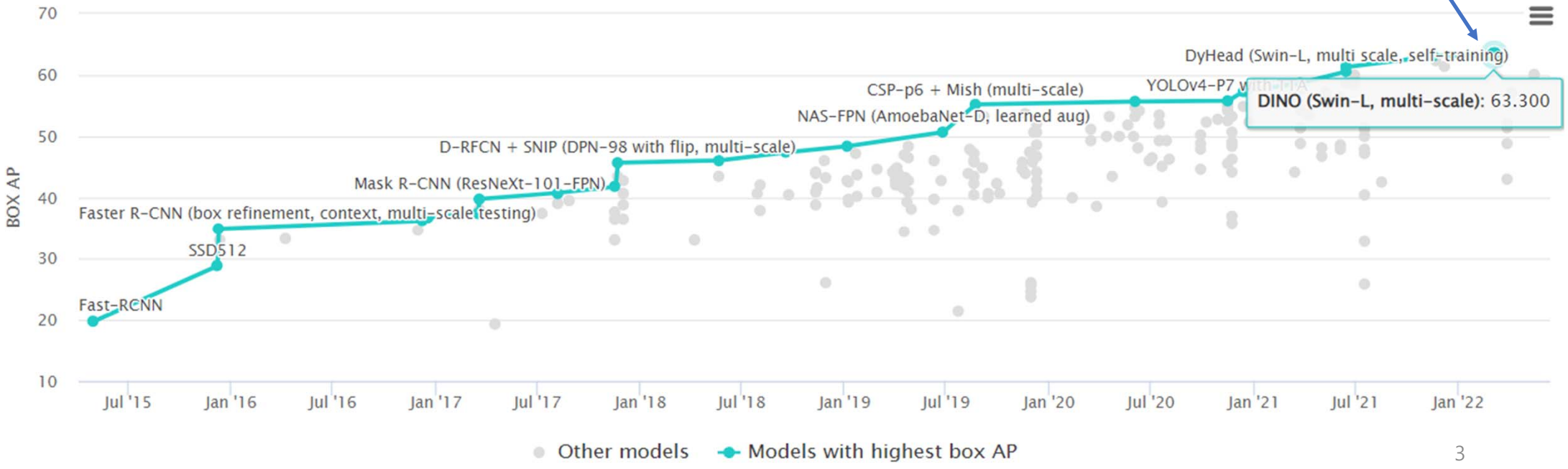
- Yolov4 / Volov5 / PP-YOLO / YOLOX / PP-YOLOE
- EfficientDet
- RegDet
- PANet++ / NAS-FPN / CSP
- SSD
- Fast R-CNN / Mask R-CNN / RetinNet

Part2

- UniverseNet
- Cascade Eff-B7 NAS-FPN
- DyHead / DINO

- DETR / Deformable DETR

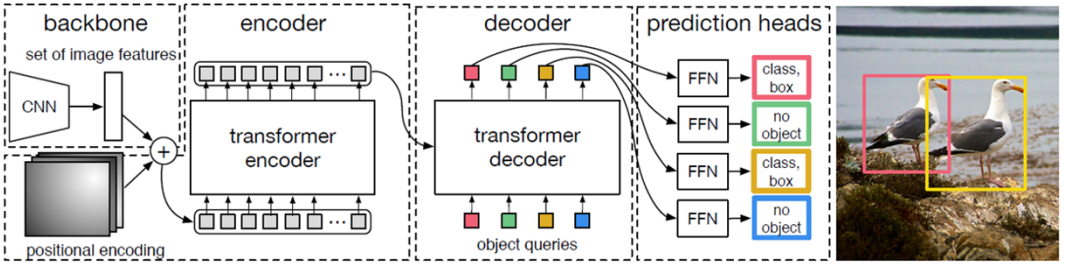
DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection



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

Facebook AI

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, “End-to-End Object Detection with Transformers,” ECCV 2020.



Facebook AI Blog : https://ai.facebook.com/blog/end-to-end-object-detection-with-transformers/

Official Code : https://github.com/facebookresearch/detr

Author’s Youtube : https://youtu.be/utxbUlo9CyY

[발췌 자료]
PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee
https://velog.io/@sjinu/논문리뷰End-to-End-Object-Detection-with-Transformers

https://keyog.tistory.com/32

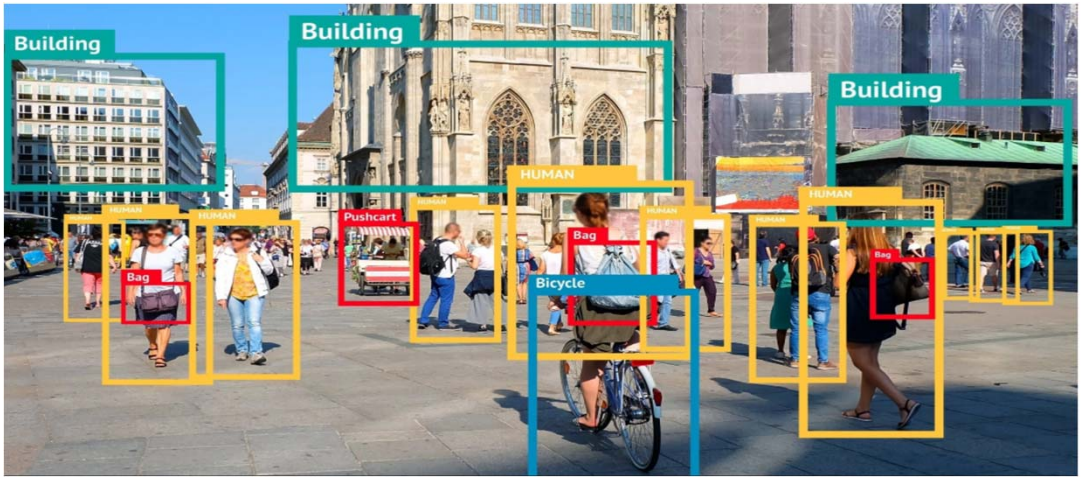
고려대학교 DSBA DETR : http://dsba.korea.ac.kr/seminar/?mod=document&uid=1784

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

[발췌 자료] 고려대학교 DSBA DETR :
http://dsba.korea.ac.kr/seminar/?mod=document&uid=1784

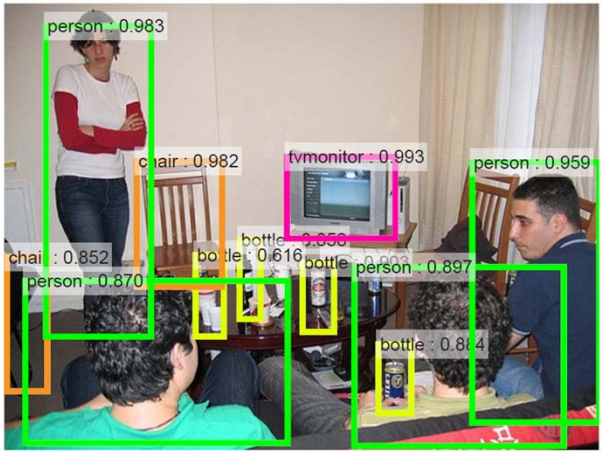
Object Detection

- 주어진 이미지에 대한 이해를 바탕으로 물체(object)를 탐지한 후, 해당 물체의 클래스와 위치를 예측하는 task
- YOLO로 대표되는 1-stage detector와 Faster R-CNN으로 대표되는 2-stage detector로 크게 나눌 수 있으며, 이들의 차이는 detector 앞단에 후보 지역을 추천하는 RPN(Region=Proposal=Network) 존재 여부
- 일반적으로 1-stage detector는 추론 속도가 빨라 real-time=task에 주로 사용되는 반면, 정확도 측면에서 2-stage=detector에 비해 다소 뒤떨어짐



How to Run Object Detection YOLOX on PyTorch with Docker on NVIDIA® Jetson™ Modules?

https://www.forecr.io/blogs/ai-algorithms/how-to-run-object-detection-yolox-on-pytorch-with-docker-on-nvidia%C2%AE-jetson%E2%84%A2-modules

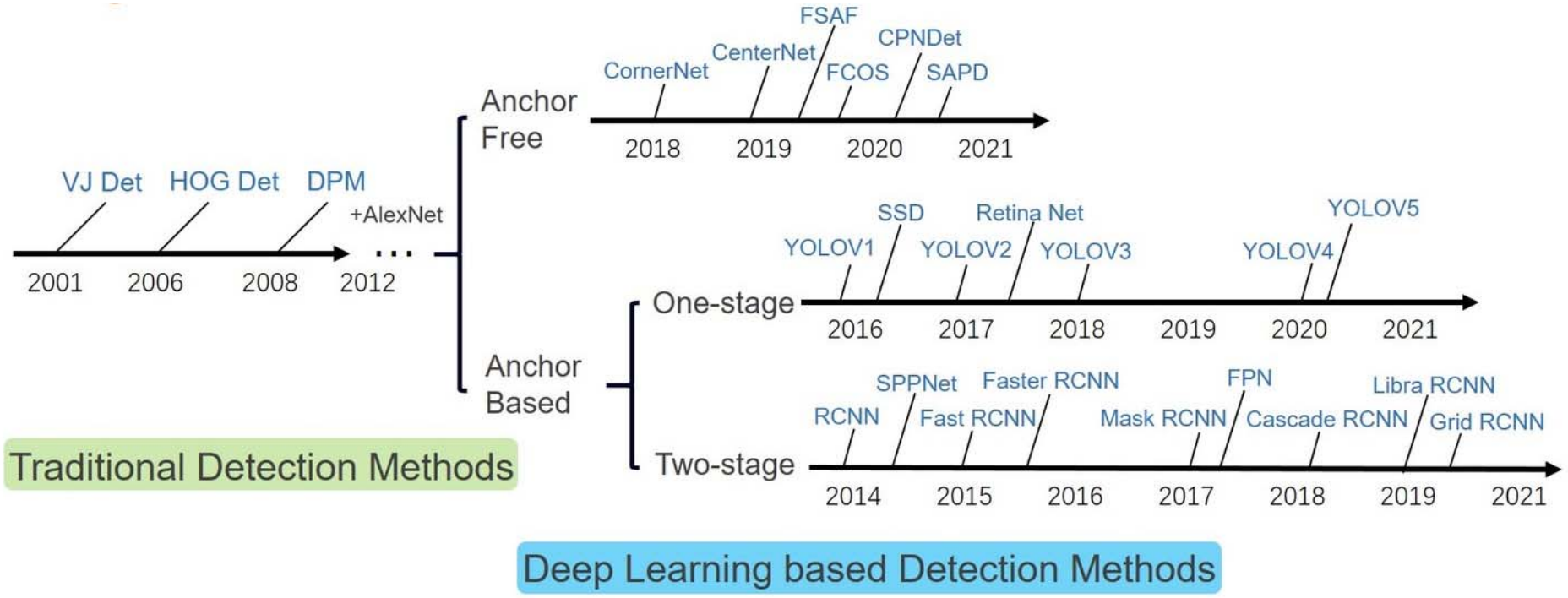


Faster R-CNN

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

[발췌 자료] 고려대학교 DSBA DETR :
http://dsba.korea.ac.kr/seminar/?mod=document&uid=1784

Object Detection Milestones



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

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Abstract

- Present a new method that views **object detection as a direct set prediction problem**.
- Our approach **streamlines the detection pipeline**, effectively **removing the need for many hand-designed components like a NMS(non-maximum suppression) procedure or anchor generation that explicitly encode our prior knowledge about the task**.
- The main ingredients of DEtection TRansformer or DETR, are **a set-based global loss that forces "unique predictions" via bipartite matching, and a transformer encoder-decoder architecture.**
- **Given a fixed small set of learned object queries, DETR reasons about the relations of the objects and the global image context to directly output the final set of predictions in parallel.**
- The new model is conceptually simple and does not require a specialized library, unlike many other modern detectors.
- DETR demonstrates accuracy and run-time performance on par with **the well-established and highly-optimized Faster RCNN baseline** on the challenging COCO object detection dataset. Moreover, DETR can be easily generalized to produce ***panoptic segmentation*** in a unified manner.

DETR의 핵심적인 마인드

- **Transformer**와 **이분 매칭(Bipartite-matching)** 기반 새로운 detection 구조
- Object detection task가 "복잡한 라이브러리를 최소화 하고, Classification 처럼 간단하게 행해져야 한다" 는 것임.
  - ✓ 기존의 Faster R-CNN, YOLO와 같은 전통적인 object detection 모델은 NMS 등 수 많은 proposals을 추려내는 과정을 거쳐야 하지만, 본 연구가 제안하는 모델은 Transformer encoder-decoder와 Bipartite matching을 이용해 Unique한 예측들을 행하는, 아주 간단한 구조로 이루어져 있음
- Inference code도 50줄 내외의 짧은 코드로 이루어져 있다고 강조함
- 구조적으로 간결함에도 다른 task에 확장성이 높고(e.g. panoptic segmentation), Attention 메커니즘에 의해 전역적 정보를 이용함에 따라 **큰 물체 탐지에 대해서 Faster R-CNN에 비해 더 높은 성능**을 보여줌

✓ NMS (Non-maximum suppression) : NMS는 하나의 객체에 중복된 prediction을 제거함.

✓ Panoptic segmentation : 일반적인 시맨틱 segmentation은 class별로 segmentation을 하게 되는데, panoptic segmentation은 class별 segmentation에서 더 나아가 인스턴스별로 segmentation 하는 것

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

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Object Detection

- Predict a set of bounding boxes with labels, given an image

- Goal of Object detection는 관심 있는 각 **object에 대해 a set of bounding boxes & category labels을 예측**하는 것이다. 현재 많은 detector들은 이러한 **set prediction problem**을 간접적으로 다룬다.
  - ✓ 예를 들면, **surrogate regression**과 **classification problem**을 수 많은 **proposals** (Faster R-CNN(2015), Cascade R-CNN(2019)), **anchors** (Focal loss for dense object detection(2017)), 또는 **window centers** (Objects as points(2019), Fully convolutional one-stage object detection(2019))통해 정의한다.

Classical Approach to Detection

- Popular approach: detection := classification of boxes
- Requires selecting a subset of candidate boxes
- Regression step to refine the predictions
- Typically non differentiable

- 위 모델들의 성능은 거의 **겹치는 예측들을 Postprocessing**하거나, **anchor set을 디자인**하거나, **heuristics하게 target boxes를 anchor에 할당**하는 데(Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection(2019))에 크게 의존한다.
- 우리는 위와 같은 과정을 간소화 하기 위해서 **surrogate task를 하지 않는 direct set prediction을 수행하는 방법**을 제안한다. 이 end-to-end 철학은 기계 번역이나 대화 인식과 같은 굉장히 복잡하게 구조화된 예측 태스크에서의 진보를 이끌었지만, 아직 object detection에는 없었다. 기존의 시도들은 사전 지식의 다른 형태를 추가하거나(End-to-end people detection in crowded scenes, Learning non-maximum suppression 등), 경쟁력 있는 성능을 보여주지 못했다. 본 연구에서는 이 Gap을 줄이고자 한다.

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

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

DETR in a nutshell

- Direct set prediction formulation
- No geometric priors(NMS, anchors, …)
- Fully differentiable
- Competitive performance
- Extensible to other tasks

Reframing the Task of Object Detection

- DETR casts the object detection task as an image to set problem
- Given an image, the model must predict an unordered set of all the objects present , each represented by its class , along with a tight bounding box surrounding each one.
- This formulation is particularly suitable for Transformers

학습 과정을 '**direct set prediction problem**'으로 다룬다.

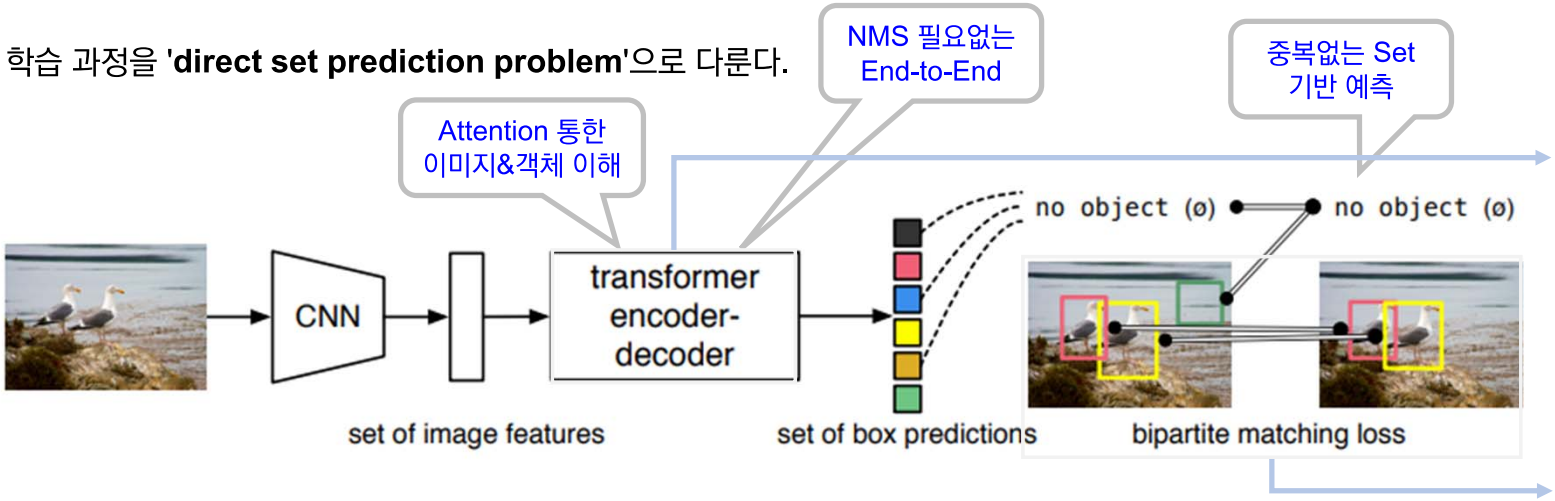


Fig. 1: DETR directly predicts (in parallel) the final set of detections by combining a common CNN with a transformer architecture. During training, bipartite matching uniquely assigns predictions with ground truth boxes. Prediction with no match should yield a "`no object`" (∅) class prediction.

첫번째; **sequence prediction**에 쓰이는 유명한 모델 중 하나인 **transformer**의 **encoder-decoder** 구조를 채택하였다. **transform**의 **self-attention mechanism**(sequence 내 elements 들 사이의 모든 pairwise 상호작용을 모델링하는 매커니즘)은 이 구조가 **removing duplicate prediction**과 같은 **set prediction**의 제약을 특히 다루기 쉽게 만들어 주었다.

두번째, DETR 모델은 **한 번에 모든 object**를 예측하기 위해, 예측 object와 ground-truth object 사이의 양자간 매칭(**bipartite matching**) 을 수행하는 **a set loss function**(여러 개을 통해 end-to-end로 학습된다. 특히, spatial anchor(?)나 non-maximal suppression과 같은 사전지식을 인코딩해야 하는 여러 수작업 요소들을 버림으로써 detection pipeline을 간소하게 했다.

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

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

DETR's feature compared to most work on direct set prediction

- Main features of DETR are the **conjunction of 1) the bipartite matching loss and 2) transformers with (non-autoregressive) parallel decoding** (BERT(2019), Parallel wavenet(2017), Mask-predict(2019) 등 기계번역 연구들)
- 대부분 detection method들과는 다르게, **DETR은 customized layer를 필요로 하지 않으며, 따라서 기본적인 CNN이나 transformer classes를 포함한 어떠한 framework에서도 reproduce될 수 있다**.
- 기존의 (visual 관련 분야의) direct set prediction 연구들은 RNN을 통해 autoregressive decoding에 중점을 두었다. 우리 모델의 **matching loss function은 ground truth object의 'Unique 예측'을 할당하므로 예측된 object들의 permutation(순서, 순열)에는 변하지 않기 때문에, 병렬적으로 autoregressive decoding task를 제거할 수 있다.**
- 성능평가 또한 진행했는데, Faster R-CNN은 계속해서 많은 수작업을 겪었기 때문에, 논문 발간 당시보다 성능이 굉장히 좋아진 상태였음에도 불구하고, 우리의 DETR 모델은 그와 비슷한 성능을 보였다(on COCO dataset). 더 정확히는, **DETR**은 (아마 Transformer의 non-local computations 때문에) **large object에 대한 성능이 좋으나, small objects에 대한 성능은 그리 좋지 않았다**. 우리는 이런 문제가 Faster R-CNN에 적용된 FPN의 개발과 같은 방식으로 개선될 수 있을 거라 믿는다.
- Training Settings은 기존의 object detector들과는 여러 관점에서 다르다. 새로운 모델은 **추가적으로 긴 학습 스케줄을 필요로** 하며, **transformer에 있는 보조적인 decoding loss 또한 사용**해야 한다.
- DETR의 Design Ethos은 쉽게 더욱 복잡한 Tasks로도 확장될 수 있다. 본 연구에서는 pre-trained DETR의 top을 기반으로 학습된 simple segmentation head가 **Panoptic Segmentation**(최근에 유명세를 얻은 어려운 pixel-level recognition task)에서 경쟁력 있는 모델을 성능 상 앞질렀다는 것을 보여줄 것이다.

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

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Transformers

- Transformers are a deep learning architecture that has gained popularity in recent years.
- They rely on a simple yet powerful mechanism called attention, which enables AI models to selectively focus on certain parts of their input and thus reason more effectively
- Transformers have been widely applied on problems with sequential data, and have also been extended to tasks as diverse as speech recognition symbolic mathematics , and reinforcement learning.
- But, perhaps surprisingly, computer vision has not yet been swept up by the Transformer revolution

Object Detection

- 최근 object detection 방법으로는 2-stage detector의 경우 proposal을 이용하고, 1-stage detector의 경우 Anchor를 이용하거나 grid를 통해 object의 중심을 찾는다.
- 최근 연구(본 글에서는 anchor와 anchor-free의 격차를 연구한 논문을 말한다)에서는 **1-stage detector는 최초로 object를 찾아낼 때, anchor 또는 grid이용 과정에 크게 의존한다고 증명했다. DETR은 이런 과정을 제거하고, anchor 대신에 input 이미지에서 절대적인 box를 예측**한다.

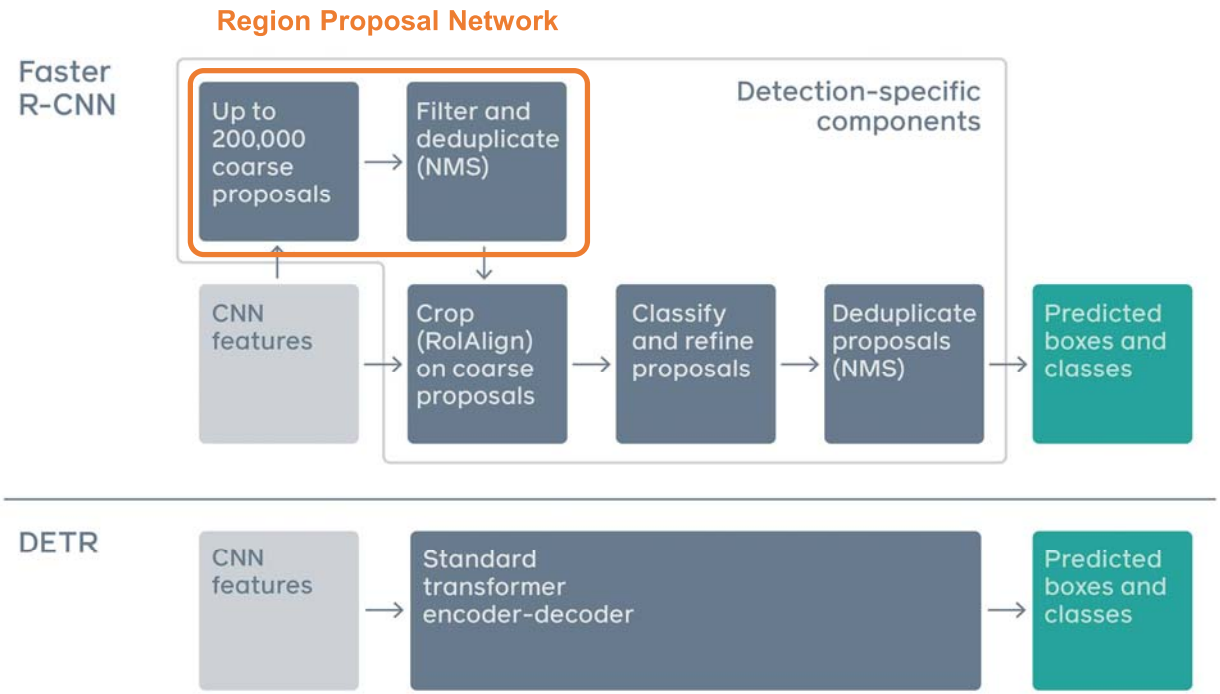
➢ Set-based loss

- 몇몇 detector는 **bipartite loss**를 사용하지만, 이것은 convolution 이나 fully-connected layer 의 서로 다른 prediction의 연관성을 수작업(수작업이라 표현되는 이유는 k-means같은 군집화 전략으로 anchor size를 직접 생성하기 때문)의 NMS post-processing로 성능 향상을 시킬 뿐이라고 한다.
- **direct set losses를 사용하면, 더 이상 NMS과정이 필요없다고** 한다. 논문의 저자들이 prior정보를 줄이는 방법을 찾는동안 몇몇의 detector들은 아직도 proposal box를 수작업으로 추가하고 있다고 한다.

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

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Streamlined Detection Pipeline



➢ Traditional two-stage detection systems, such as Faster R-CNN

- Predict object bounding boxes by filtering over a large number of coarse candidate regions, which are generally a function of the CNN features.
- Each selected region is used in a refinement step, which involves cropping the CNN features in the location defined by the region, classifying each region independently, and refining its location.
- Finally, a non-maximum suppression step is applied to remove duplicate boxes.

➢ DETR

- Simplifies the detection pipeline by leveraging a standard Transformer architecture to perform the (potentially non-differentiable) operations that are traditionally specific to object detection.

- Faster R-CNN은 RPN(Region Proposal Network)과 NMS(Non-Maximum Suppression)을 거쳐 최종적인 후보 위치를 예측한 후, 해당 지역들에 대해 다시 한번 detection을 수행하는 복잡한 pipeline을 가지고 있음
- DETR은 이미지에 관하여 직접적이고 절대적인(absolute) 방식으로 box set을 예측함으로써 NMS와 같은 hand-crafted 과정을 사용하지 않아 end-to-end 구조를 구축할 수 있음

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

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

DETR

Two ingredients are essential for direct set predictions in detection:

(1) **a set prediction loss** that forces unique matching between predicted and ground truth boxes

(2) **an architecture** that predicts (in a single pass) a set of objects and models their relation

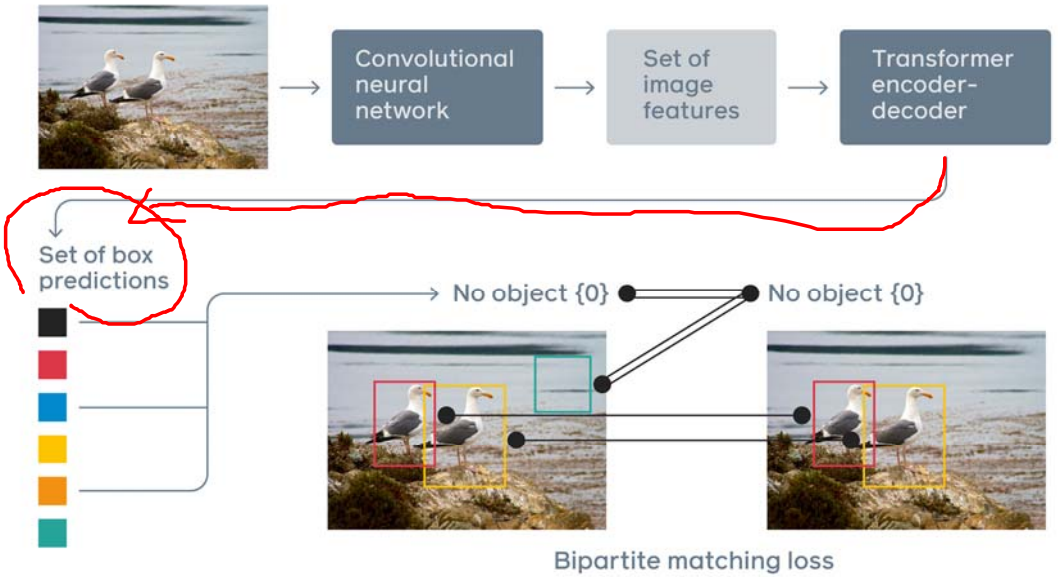


Fig. 1. DETR directly predicts (in parallel) the final set of detections by combining a common CNN with a Transformer architecture.

During training, bipartite matching uniquely assigns predictions with ground truth boxes. Predictions with no match should yield a “no object” class prediction.

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Object Detection Set Prediction Loss

- **DETR infers a fixed size set of N predictions in a single pass through the decoder**, where N is set to be significantly large than the typical number of objects in an image.
  - ✓ One of the main difficulties of training is to *score predicted objects (class, position, size) with respect to the ground truth*.
  - ✓ Our **loss** produces an ***optimal bipartite matching between predicted and ground truth objects***, and then ***optimize object-specific (bounding box) losses***.
- Let us denote by y the ground truth set of objects, and $\hat{y} = \{\hat{y}_i\}_{i=1}^{N}$ the set of N predictions.
- y also as a set of size N padded with $\emptyset$ (no object)

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

[발췌 자료] 고려대학교 DSBA DETR :
http://dsba.korea.ac.kr/seminar/?mod=document&uid=1784

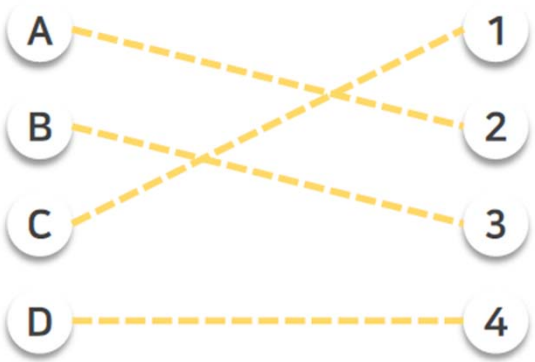
1) Bipartite matching cost (feat. Hungarian Algorithm)

- **Direct set prediction**에서 중요한 두 가지 요소는
  ① 예측과 Ground truth 사이에 중복이 없는 1:1 매칭을 가능케 해야 하며,
  ② 한 번의 추론에서 object set을 예측하고 그들의 관계를 모델링 할 수 있어야 한다.
- 이를 만족시키는 object detector를 학습시키기 위해 DETR은 Hungarian 알고리즘을 활용한 이분 매칭을 통해 loss를 정의

Bipartite matching

- Q 엔지니어 A~D가 1~4의 일을 할당 받아야 한다. 아래 작업 능률표를 기반으로 할 때 어떤 할당이 가장 최선인가?

	A	B	C	D
1	12	13	1	8
2	6	6	15	7
3	16	5	15	12
4	2	3	12	16

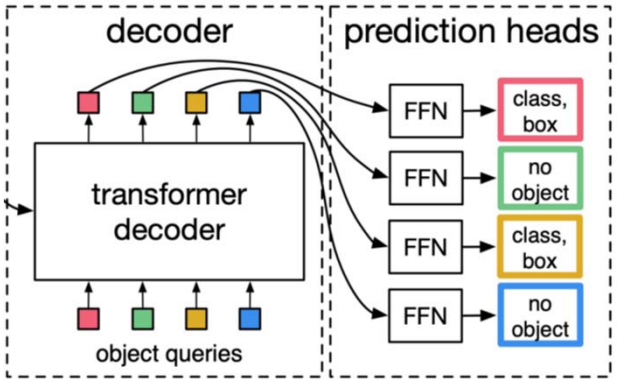


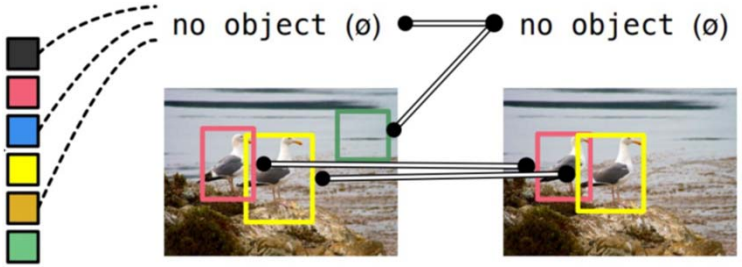
2 DETR: End-to-End Object Detection with Transformers

[발췌 자료] 고려대학교 DSBA DETR :
http://dsba.korea.ac.kr/seminar/?mod=document&uid=1784

1) Bipartite matching cost (feat. Hungarian Algorithm)

- Object query마다 예측된 결과물(① Class 유무, ② Box 좌표)과 ground truth set 간 Hungarian algorithm 기반 매칭 수행
- 이 때 매칭의 기준으로 pair-wise matching cost인 $\mathcal{L}_{match}$를 활용하며, $\mathcal{L}_{match}$를 최소화하는 최적의 순열($\hat{\sigma}$)를 찾음
- 이는 Anchor의 비교 방식과 유사하나, anchor는 동일한 ground truth object와의 중복 예측을 허용하는 반면
- DETR은 set prediction과 ground truth 간 일대일 매칭 수행하여 중복을 배제





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

$$\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) = \underbrace{-\mathbb{1}_{\{c_i \neq \varnothing\}} \hat{p}_{\sigma(i)}(c_i)}_{\text{Class 예측 Cost}} + \underbrace{\mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\sigma(i)})}_{\text{Box 좌표 예측 Cost}}$$

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

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

1) Bipartite matching cost

- To find a bipartite matching between these two sets, we search for a permutation of N elements, $\sigma \in \mathfrak{S}_N$ with the lowest cost

$$\hat{\sigma} = \arg\min_{\sigma \in \mathfrak{S}_N} \sum_{i}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}), \qquad (1)$$

  - ✓ Each Element i of ground truth set, $y_i = (c_i, b_i)$, where c_i is the target class label (which may be ∅) and $b_i = [0,1]^4$ is a vector that defines ground truth box center coordinates and its height and width relative to the image size.

- $\mathcal{L}_{match} = (y_i, \hat{y}_{\sigma(i)})$ is a pair-wise matching cost between ground truth y_i and a prediction $\hat{y}_{\sigma(i)}$ with index $\sigma(i)$ and this optimal assignment is computed efficiently with Hungarian algorithm.
- $\mathcal{L}_{match}$ takes into account both the class prediction and the similarity of predicted and ground truth boxes

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

- For the prediction with index $\sigma(i)$, $\hat{p}_{\sigma(i)}(c_i)$ is a probability of class c_i and $\hat{b}_{\sigma(i)}$ is the predicted box.

- 이렇게 matching을 찾는 과정은 사실 modern detector인 proposal(RPN)이나 anchors(FPN)를 ground truth objects에 매칭하는 heuristic assignment rules과 같은 역할을 한다. 다른 점은, duplicates가 없는 direct set prediction을 위한 1-1 매칭을 찾아야 한다는 것이다.

- 하나의 요소가 N요소의 순열에 포함되는 것을 이분 매칭을 통해 찾았을때, cost가 가장 낮게 된다.
- ground truth에 N의 크기가 될 수 있게 ∅(no object)가 추가되고, transformer의 decoder가 예측하는 객체의 class가 ground truth 객체에 포함될 때, loss가 낮아진다.

- 다른 detector와 다르게 중복되는 prediction이 나오지 않는다.
- Object와 ∅ 사이의 matching cost는 prediction에 의존하지 않는다. 즉, 이 경우에서 cost는 상수이다. Matching cost에 log-probabilities 대신 확률 $\hat{p}_{\sigma(i)}(c_i)$를 사용함. 이는 class prediction term을 $\mathcal{L}_{box}(\cdot,\cdot)$와 상응하게 만들어 준다(성능 향상).

- 이 후 Hungarian loss(이분 그래프에서 모든 정점에 대한 potential의 합이다.)로 모든 쌍을 매칭하며, linear combination of a negative log-likelihood 를 통해 class와 box의 loss를 정의 한다. (다음 페이지)

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee, & 고려대학교 DSBA DETR

2) Loss Function – Hungarian Loss

- Compute the loss function, the *Hungarian loss* for all pairs matched in the previous step
- A linear combination of a negative log likelihood for class prediction and a box loss

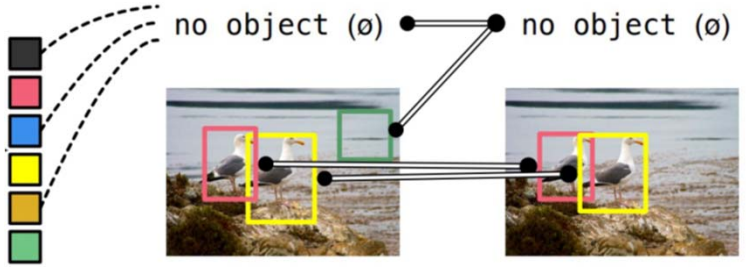
$$\mathcal{L}_{Hungarian}(y, \hat{y}) = \sum_{i=1}^{N} [-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \emptyset\}} \mathcal{L}_{box}(b_i, \hat{b}_{\sigma(i)})]$$

Class 예측 / Box 좌표 예측

Box Loss와의 균형을 위해 이분 매칭과 달리 Log term 사용

- $\hat{\sigma}$ is the optimal assignment computed by Eq. (1)
- In practice, the log prob. term is down weighted when $c_i = \emptyset$ by a factor of 10 to account for class imbalance.

- 앞선 Hungarian algorithm으로 찾은 '최적의 예측 순열 $\hat{\sigma}$'를 이용해 학습을 위한 loss 계산
- 일반적인 object detection task에서의 loss와 유사하게 정의(Class 예측 Loss + Box 예측 Loss)되나,
- bounding box loss에 **GIoU**를 추가하여 **box의 스케일에 둔감**한 loss 정의
- 설정된 slot의 개수(N=100)에 비해 object의 개수가 훨씬 적으므로(평균 7개) 예측 클래스=ø인 경우가 훨씬 많음 (Imbalanced)
- 이를 해결하기 위해 Faster-RCNN의 sub-sampling과 유사한 효과를 위해 클래스 ø인 경우 weight를 10배 낮추어 학습



[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee, & 고려대학교 DSBA DETR

Bounding Box Loss

- Unlike many detectors that do box predictions as a Δw.r.t. some initial guesses, DETR make box prediction directly
- The most commonly used ℓ_1 loss will have different scales for small and large boxes even if their relative errors are similar.
- To mitigate this issue, DETR use a linear combination of the ℓ_1 loss and **GIoU** (generalized IoU) loss that is scale invariant.

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

$$\mathcal{L}_{\text{iou}}(b_{\sigma(i)}, \hat{b}_i) = 1 - \left(\frac{|b_{\sigma(i)} \cap \hat{b}_i|}{|b_{\sigma(i)} \cup \hat{b}_i|} - \frac{|B(b_{\sigma(i)}, \hat{b}_i) \setminus b_{\sigma(i)} \cup \hat{b}_i|}{|B(b_{\sigma(i)}, \hat{b}_i)|}\right)$$

GIoU(Generalized Intersection over Union) loss

- 일반적인 detector에서의 bounding box loss는 예측과 ground truth 간의 상대적(Δ)인 좌표 및 크기를 이용하여 정의 (*e.g. Anchor에서 정의된 box를 얼마나 움직이고, 얼마나 키워야 ground truth에 가까워지는가*)
- 반면 DETR의 경우 절대적인 bounding box의 좌표 및 크기를 direct 하게 예측하므로 큰 물체와 작은 물체 loss를 계산 시 일반적인 L1 loss 외에 scale 보정이 필요함 → GIoU(Generalized Intersection over Union) 사용

Box Loss에 대한 ablation study 결과

Table 4: Effect of loss components on AP. We train two models turning off ℓ_1 loss, and GIoU loss, and observe that ℓ_1 gives poor results on its own, but when combined with GIoU improves AP_M and AP_L. Our baseline (last row) combines both losses.

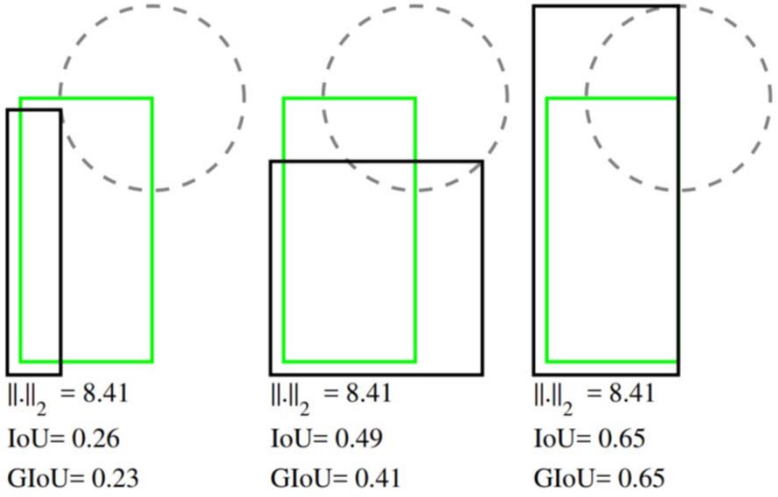
class	ℓ_1	GIoU	AP	Δ	AP_{50}	Δ	AP_S	AP_M	AP_L
✓	✓		35.8	-4.8	57.3	-4.4	13.7	39.8	57.9
✓		✓	39.9	-0.7	**61.6**	0	**19.9**	43.2	57.9
✓	✓	✓	**40.6**	-	**61.6**	-	**19.9**	**44.3**	**60.2**

→ GIoU가 모델의 성능을 개선하는 데에 중요한 역할을 수행

DETR: End-to-End Object Detection with Transformers

[발췌 자료] 고려대학교 DSBA DETR :
http://dsba.korea.ac.kr/seminar/?mod=document&uid=1784

GIoU

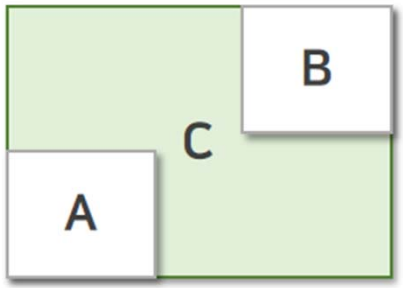


일반적으로 **IoU**을 높이기 위해 L_n **Loss**를 사용하나 이는 실제로 **IoU**를 개선하는 데에 큰 관련이 없다!

+

Box가 겹치지 않으면 떨어진 정도를 반영할 수 없다!

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



- ✓ 두 영역이 겹치지 않는(IoU=0) 경우에도 loss를 정의해 학습에 이용할 수 있음
- ✓ $\mathcal{L}_{Box} = 1 - GIoU$

Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression, H Rezatofighi et al., 2019

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

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

DETR Architecture

Extract a feature representation

Make the final detection prediction

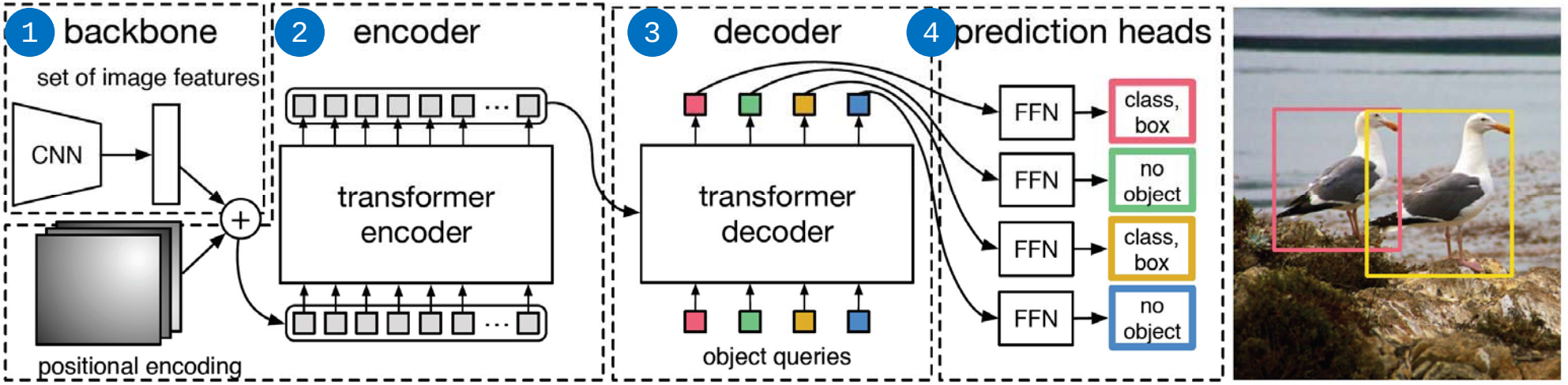


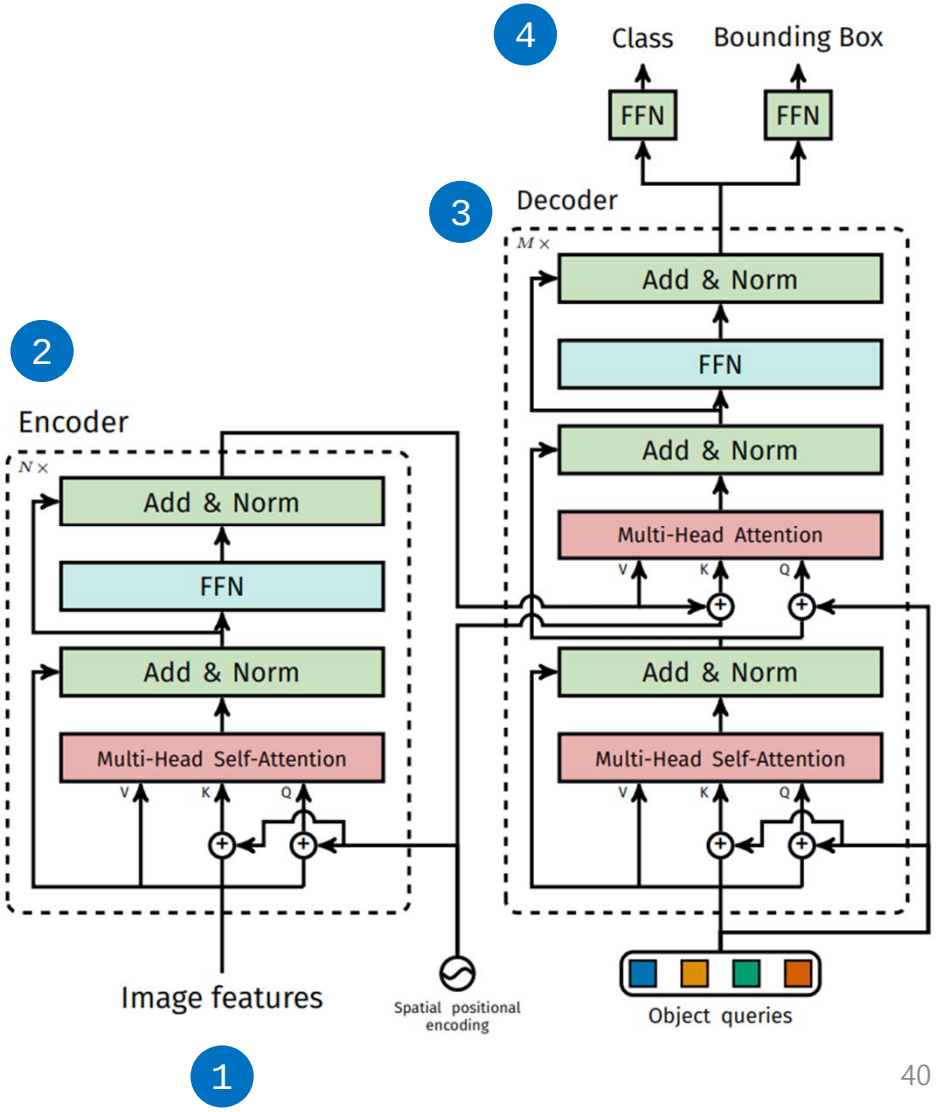
Fig. 2: DETR uses a conventional CNN backbone to learn a 2D representation of an input image. The model flattens it and supplements it with a positional encoding before passing it into a **transformer encoder**. A **transformer decoder** then takes as input a small fixed number of learned positional embeddings, which we call **object queries**, and additionally attends to the encoder output. We pass each output embedding of the decoder to a shared feed forward network (FFN) that predicts either a detection (class and bounding box) or a “no object" class.

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

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Architecture of DETR's Transformer

- The detailed description of the transformer used in DETR, with **positional encodings passed at every attention layer**
- Image features from the CNN backbone are passed through the transformer encoder, together with **spatial positional encoding that are added to queries and keys at every multihead self-attention layer**.
- Then, the decoder receives **queries (initially set to zero)**, **output positional encoding (object queries)**, and **encoder memory**, and **produces the final set of predicted class labels and bounding boxes** through multiple multihead self-attention and decoder-encoder attention. The first self-attention layer in the first decoder layer can be skipped.

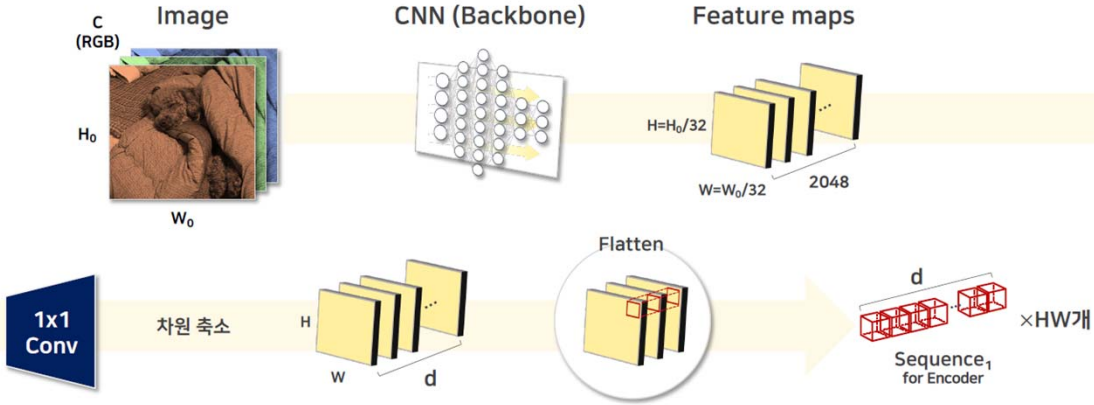


2 DETR: End-to-End Object Detection with Transformers

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

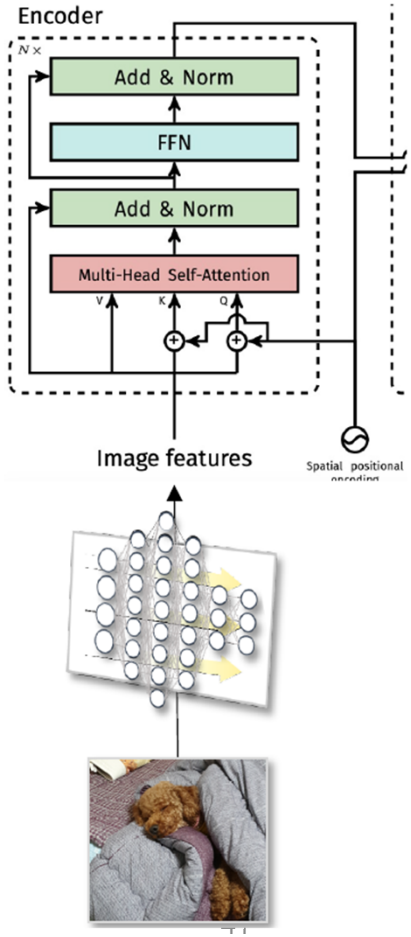
1 Backbone : Image to Input

- Starting from the initial image $x_{img} \in \mathbb{R}^{3 \times H_0 \times W_0}$ (with 3 color channels), conventional CNN backbone generates a lower resolution activation map $f \in \mathbb{R}^{C \times H \times W}$
- Typical value DETR use are C=2048 and $H, W = \frac{H_0}{32}, \frac{W_0}{32}$



2 Transformer Encoder

- 1x1 convolution reduces the channel dimension of $f \in \mathbb{R}^{C \times H \times W}$ from C to d. Creating a new feature map $z_0 \in \mathbb{R}^{d \times H \times W}$, d is smaller than C
- The encoder expects a sequence as input hence the spatial dimensions of z_0 is collapsed into one dimension, resulting in a $d \times HW$ feature map.
- Since the transformer architecture is permutation invariant, DETR supplement it with fixed positional encodings that are added to the input of each attention layer.



- Encoder는 attention mechanism을 기반으로 feature map의 pixel과 pixel간의 관계를 학습
- Locality 중심의 CNN과 다르게 global한 정보를 학습함으로써 이미지를 이해하고, 특히 object detection - task에 맞게 학습됨으로써 이미지 내 object의 위치, 관계 등 학습하게 됨

Experiment – Encoder self-attention

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

[발췌 자료] 고려대학교 DSBA DETR :
http://dsba.korea.ac.kr/seminar/?mod=document&uid=1784

2 Transformer Encoder

- Transformer의 encoder는 **permutation invariant** 하기 때문에 토큰 간 차이를 나타낼 수 있도록 **positional encoding**이 필요
- DETR에서는 **2D fixed sine positional encoding**을 사용 (ViT 논문에서는 1D와 2D 간 성능 차이는 유의미하지 않음)
- 모든 **encoder layer**마다 **positional encoding**을 **query, key**에 더하여 입력으로 사용

1D positional encoding (Standard Transformer 1))

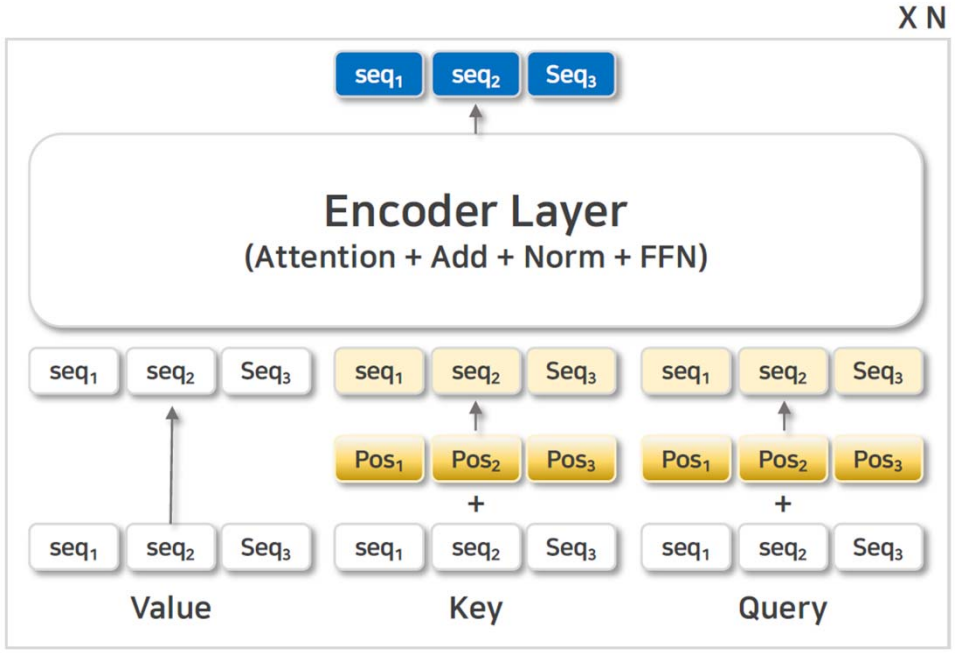
```
sinusoid_table[:, 0::2] = np.sin(sinusoid_table[:, 0::2])  # dim 2i
sinusoid_table[:, 1::2] = np.cos(sinusoid_table[:, 1::2])  # dim 2i+1
return torch.FloatTensor(sinusoid_table).unsqueeze(0)
```

https://github.com/jadore801120/attention-is-all-you-need-pytorch

2D positional encoding (DETR 2))

```
pos_x = x_embed[:, :, :, None] / dim_t
pos_y = y_embed[:, :, :, None] / dim_t
pos_x = torch.stack((pos_x[:, :, :, 0::2].sin(), pos_x[:, :, :, 1::2].cos()), dim=4).flatten(3)
pos_y = torch.stack((pos_y[:, :, :, 0::2].sin(), pos_y[:, :, :, 1::2].cos()), dim=4).flatten(3)
pos = torch.cat((pos_y, pos_x), dim=3).permute(0, 3, 1, 2)
return pos
```

https://github.com/facebookresearch/detr

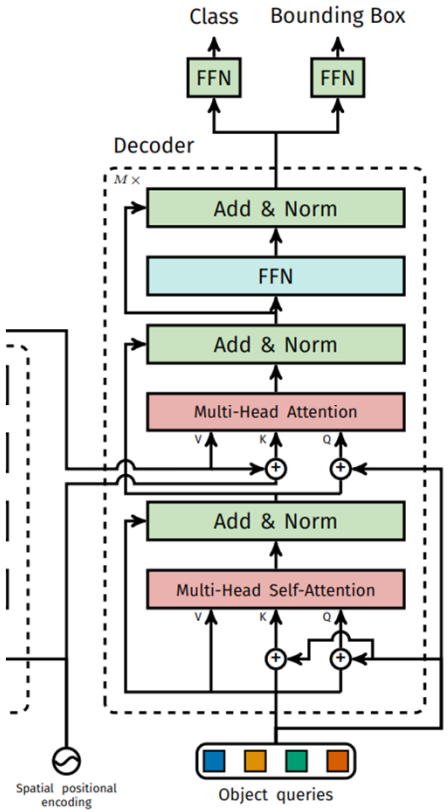


[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee, & 고려대학교 DSBA DETR

3 Transformer Decoder

- The decoder follows the standard architecture of the transformer, transforming N embeddings of size *d* using multi headed self and encoder decoder attention mechanisms.
- The difference with the original trans former is that our model decodes the N objects in parallel at each decoder layer
- Since the decoder is also permutation invariant, the N input embeddings must be different to produce different results
- These input embeddings are learnt positional encodings that we refer to as ***object queries***, and similarly to the encoder, they are added to the input of each attention layer

- Decoder의 역할은 어떠한 입력값을 받아 이미지 내에 존재하는 object의 클래스 및 위치를 출력하는 것
- Permutation invariant한 transformer의 특성 때문에 입력값이 서로 달라야 서로 다른 출력값(물체 예측)을 낼 수 있으므로,
- 학습 가능한 positional encoding(object query)을 랜덤하게 초기화 하여 입력값으로 사용 (*단 decoder은 모든 것이 set이므로, "positional embedding"이라는 것은 의미가 없음*)



④ **FFN**
Feed forward network를 통해 head들을 결합

③ **Encoder-Decoder Attention**
Encoder의 결과물과 query slot 간 attention을 통해 어느 query가 어떤 위치에서 object를 찾을 수 있을 지 학습

② **Decoder Self-Attention**
Self attention을 통해 query slot 간 관계 학습
(첫번째 self attention은 의미가 없으므로 제거해도 성능 차이 없음)

① **Object query 입력**

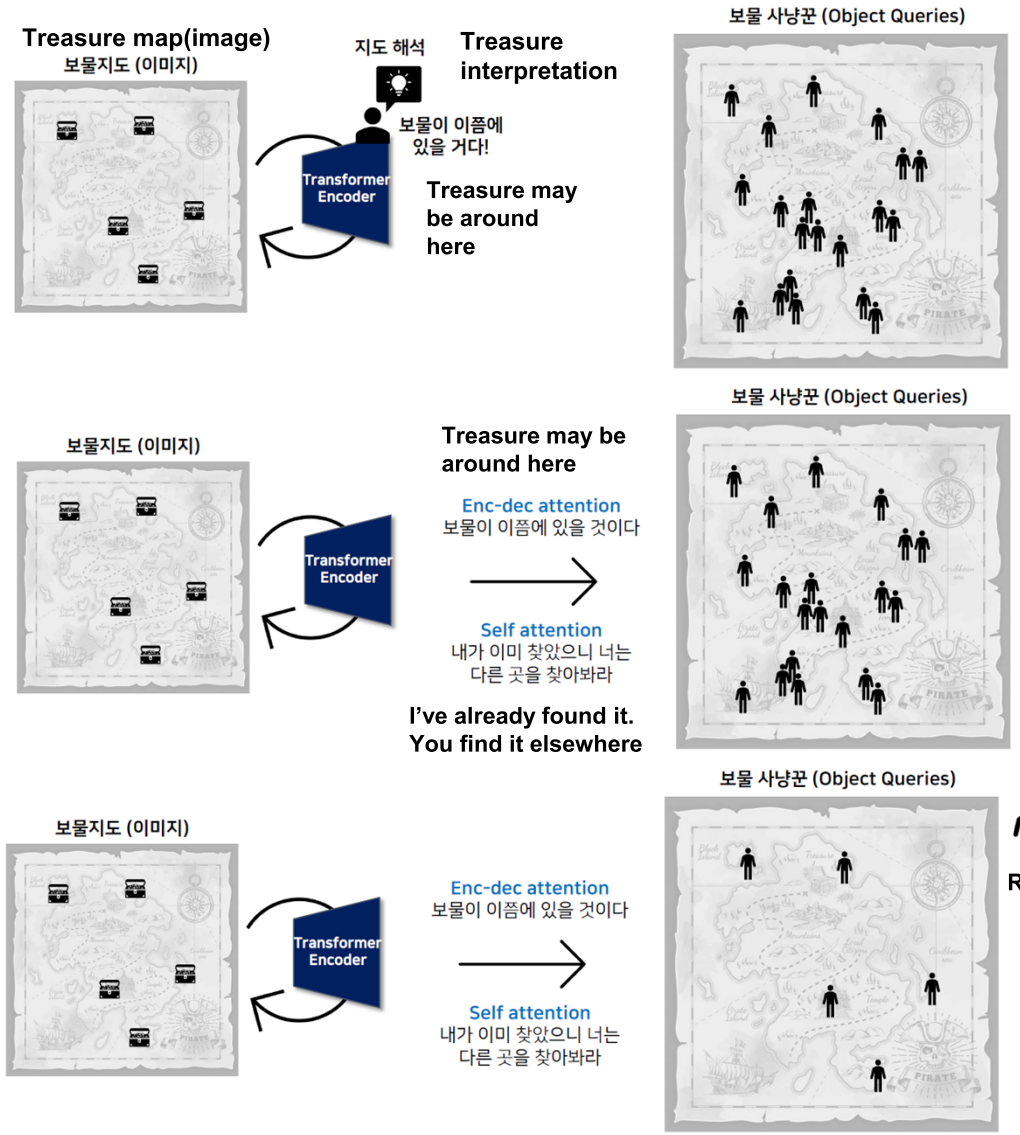
2 DETR: End-to-End Object Detection with Transformers

[발췌 자료] 고려대학교 DSBA DETR :
http://dsba.korea.ac.kr/seminar/?mod=document&uid=1784

3 Transformer Decoder

➢ Object Queries

- Object query는 곧 정보를 담기 위한 그릇(slot)으로 생각할 수 있고,
- Decoder의
  ① Encoder-decoder attention을 통해 이미지의 어느 부분을 위주로 봐야할 지 (물체가 어느 위치에 있을 확률이 높을 지),
  ② Self-attention을 통해 자신들의 역할을 어떻게 분배하여 최적의 일대일 매칭을 수행할 수 있을 지를 학습하게 됨

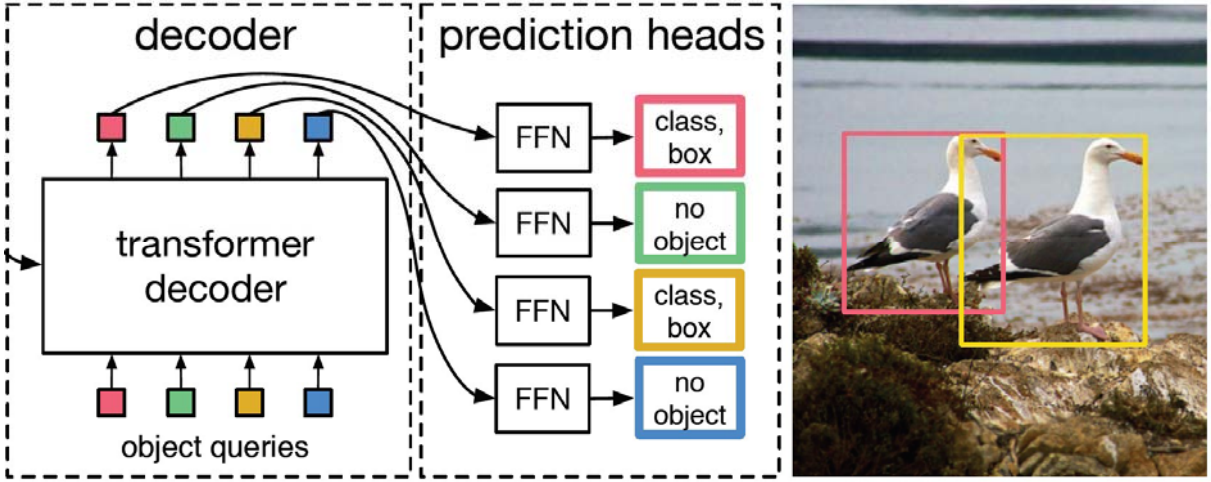


It turns out experimentally that *it will tend to reuse a given slot to predict objects in a given area of the image*

[발췌 자료] 고려대학교 DSBA DETR :
http://dsba.korea.ac.kr/seminar/?mod=document&uid=1784

3 Transformer Decoder

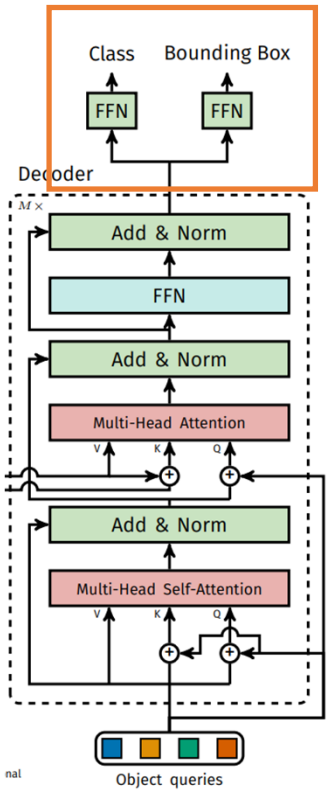
- Decoder의 각 슬롯들은 이미지/물체에 대한 이해(global reasoning)와 더불어 각자의 관계(역할)을 학습하여 슬롯 개수(N) 만큼의 임베딩 값을 출력, 즉 전체 이미지를 하나의 context로 이용
- 각 임베딩 값을 FFN에 태워 특정 슬롯이 예측한 물체의 유무 + 물체의 위치를 출력하며, NLP와 같이 순서적으로 관계가 존재하지 않으므로 auto-regressive 하지 않은 set-prediction을 수행



[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

4 Prediction feed-forward networks (FFN) and Auxiliary Decoding Losses

- The final prediction is computed by a 3-layer perceptron with ReLU and hidden dimension d
- The FFN predicts the normalized center coordinates, height and width of the box.
- Authors add prediction FFNs and Hungarian loss after each decoder layer. All predictions FFNs share their parameters.



[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

A.6 PyTorch inference code

```
import torch
from torch import nn
from torchvision.models import resnet50

class DETR(nn.Module):

    def __init__(self, num_classes, hidden_dim, nheads,
                 num_encoder_layers, num_decoder_layers):
        super().__init__()
        # We take only convolutional layers from ResNet-50 model
        self.backbone = nn.Sequential(*list(resnet50(pretrained=True).children())[:-2])
        self.conv = nn.Conv2d(2048, hidden_dim, 1)
        self.transformer = nn.Transformer(hidden_dim, nheads,
                                          num_encoder_layers, num_decoder_layers)
        self.linear_class = nn.Linear(hidden_dim, num_classes + 1)
        self.linear_bbox = nn.Linear(hidden_dim, 4)
        self.query_pos = nn.Parameter(torch.rand(100, hidden_dim))
        self.row_embed = nn.Parameter(torch.rand(50, hidden_dim // 2))
        self.col_embed = nn.Parameter(torch.rand(50, hidden_dim // 2))

    def forward(self, inputs):
        x = self.backbone(inputs)
        h = self.conv(x)
        H, W = h.shape[-2:]
        pos = torch.cat([
            self.col_embed[:W].unsqueeze(0).repeat(H, 1, 1),
            self.row_embed[:H].unsqueeze(1).repeat(1, W, 1),
        ], dim=-1).flatten(0, 1).unsqueeze(1)
        h = self.transformer(pos + h.flatten(2).permute(2, 0, 1),
                             self.query_pos.unsqueeze(1))
        return self.linear_class(h), self.linear_bbox(h).sigmoid()

detr = DETR(num_classes=91, hidden_dim=256, nheads=8, num_encoder_layers=6, num_decoder_layers=6)
detr.eval()
inputs = torch.randn(1, 3, 800, 1200)
logits, bboxes = detr(inputs)
```

- Listing 1: DETR PyTorch inference code. For clarity it uses learnt positional encodings in the encoder instead of fixed, and positional encodings are added to the input only instead of at each transformer layer. Making these changes requires going beyond PyTorch implementation of transformers, which hampers readability.

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

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

DETR (ECCV2020) : https://www.youtube.com/watch?v=utxbUlo9CyY **1:40 ~ 2:36**

End-to-End Object Detection with Transformers

Nicolas Carion
Facebook AI

Francisco Massa
Facebook AI

Gabriel Synnaeve
Facebook AI

Nicolas Usunier
Facebook AI

Alexander Kirillov
Facebook AI

Sergey Zagoruyko
Facebook AI

FACEBOOK AI

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Experiments

- Experiments on COCO 2017 detection and panoptic segmentation dataset. There are 7 instances per image on average , up to 63 instances in a single image in training set.
- Two different backbones are used: a ResNet-50(DETR) and a ResNet-101(DETR R101).
- Dilation to the last stage of the backbone also used for increasing the feature resolution: DETR-DC5 and DETR-DC5-R101 (dilated C5 stage)
- This modification increases the resolution by a factor of 2, thus improving performance for small object, at the cost of a 16x higher cost in the self attentions of the encoder, leading to an overall 2x increase in computational cost.

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Experiments : DETR vs Faster R-CNN

Table 1: Comparison with Faster R-CNN with a ResNet-50 and ResNet-101 backbones on the COCO validation set.

We use torchscript Faster R-CNN and DETR models to measure FLOPS and FPS. Results without R101 in the name correspond to ResNet-50.

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**

Faster RCNN과의 최대한 합리적인 비교를 위해, 기본적인 Faster RCNN 모델에 아래와 같은 요소를 추가

① Bounding box loss function에 g-IOU Loss를 추가

② DETR 학습 시와 동일한 crop augmentation을 추가

③ 더욱 긴 training schedule(x3)을 이용하여 학습

크기가 큰 물체에 대해서는 Faster RCNN 대비 높은 성능을 달성했으나, 크기가 작은 물체에 대해서는 낮은 성능을 기록

The top section shows results for Faster R-CNN models in Detectron2 [50],

The middle section shows results for Faster R-CNN models with GIoU [38], random crops train-time augmentation, and the long 9x training schedule.

DETR models achieve comparable results to heavily tuned Faster R-CNN baselines, having lower AP_S but greatly improved AP_L.

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Deformable DETR (ICLR2021) https://arxiv.org/abs/2010.04159

Deformable DETR: Deformable Transformers for End-to-End Object Detection

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, Jifeng Dai

DETR has been recently proposed to eliminate the need for many hand-designed components in object detection while demonstrating good performance. However, it suffers from slow convergence and limited feature spatial resolution, due to the limitation of Transformer attention modules in processing image feature maps. To mitigate these issues, we proposed Deformable DETR, whose attention modules only attend to a small set of key sampling points around a reference. Deformable DETR can achieve better performance than DETR (especially on small objects) with 10 times less training epochs. Extensive experiments on the COCO benchmark demonstrate the effectiveness of our approach. Code is released at this https URL.

Method	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	params	FLOPs	Training GPU hours	Inference FPS
Faster R-CNN + FPN	109	42.0	62.1	45.5	26.6	45.4	53.4	42M	180G	380	26
DETR	500	42.0	62.4	44.2	20.5	45.8	61.1	41M	86G	2000	28
DETR-DC5	500	43.3	63.1	45.9	22.5	47.3	61.1	41M	187G	7000	12
DETR-DC5	50	35.3	55.7	36.8	15.2	37.5	53.6	41M	187G	700	12
DETR-DC5+	50	36.2	57.0	37.4	16.3	39.2	53.9	41M	187G	700	12
Deformable DETR	50	43.8	62.6	47.7	26.4	47.1	58.0	40M	173G	325	19
+ iterative bounding box refinement	50	45.4	64.7	49.0	26.8	48.3	61.7	40M	173G	325	19
++ two-stage Deformable DETR	50	46.2	65.2	50.0	28.8	49.2	61.7	40M	173G	340	19

Table 1: Comparision of Deformable DETR with DETR on COCO 2017 val set. DETR-DC5+ denotes DETR-DC5 with Focal Loss and 300 object queries

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Number of Encoder Layers

- Without encoder layers, overall AP drops by 3.9 points, with a more significant drop of 6.0 AP on large objects
- By global scene reasoning, the encoder is important for disentangling objects.

Table 2: Effect of encoder size. Each row corresponds to a model with varied number of encoder layers and fixed number of decoder layers. Performance gradually improves with more encoder layers.

#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

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Encoder Self Attention

Back to Encoder

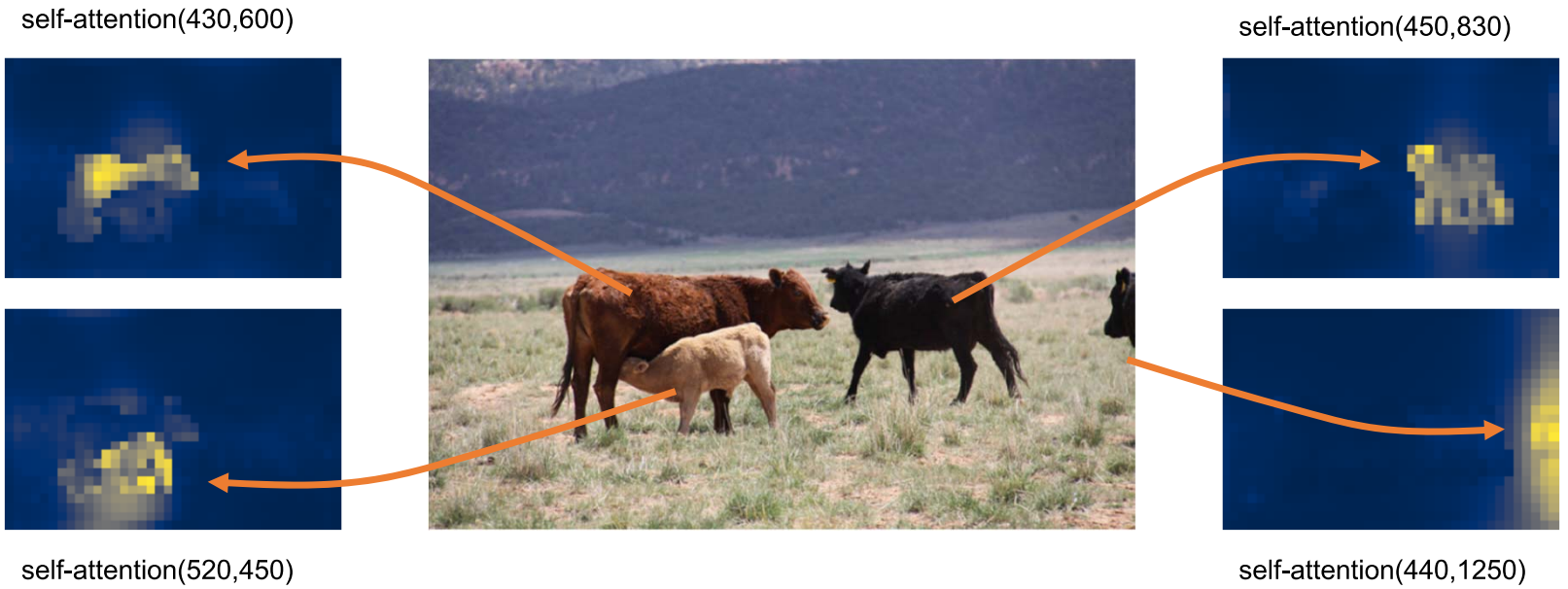


Fig. 3: Encoder self-attention for a set of reference points. The encoder is able to separate individual instances. Predictions are made with baseline DETR model on a validation set image.

- Encoder를 사용하지 않는 경우 약 4~5 AP 감소가 있으며, 특히 큰 물체(APL)에 대한 detection 성능이 크게 저하됨
- DETR의 attention mechanism이 이미지 내 물체를 분리(disentangle)하는 데에 굉장히 중요함을 의미
- **Attention map을 통해 Encoder에서부터 이미 물체를 어느 정도 구분하고 있음을 알 수 있고,**
- 따라서 학습된 임베딩을 **Key와 Value로 사용하는 Decoder가 detection** 하는 과정을 좀 더 쉽게 만들어 줌

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Number of Decoder Layers

- A single decoding layer of the transformer is not able to compute any cross correlation between the output elements, and thus it is prone to making multiple predictions for same object . → NMS improves performance

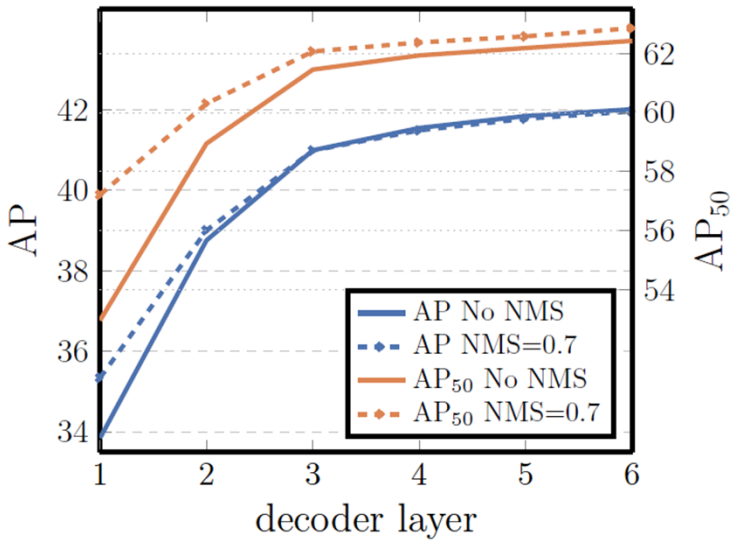


Fig. 4: AP and AP50 performance after each decoder layer. A single long schedule baseline model is evaluated. DETR does not need NMS by design, which is validated by this figure. NMS lowers AP in the final layers, removing TP predictions, but improves AP in the first decoder layers, removing double predictions, as there is no communication in the first layer, and slightly improves AP50.

- Decoder layer가 깊어질수록 예측 정확도가 높아지며, 특히 object들 간의 관계를 더욱 잘 학습하므로 NMS(Non-Maximum=Suppression)을 추가하여도 효과가 없음
- Decoder가 물체를 탐지하는 학습 과정을 통해 각 object들의 말단(extremities)에 어텐션을 크게 주도록 학습되며,
- object가 겹치는 경우(occluded) 잘못된 attention을 주지 않도록 학습됨을 확인할 수 있음
- 즉 encoder가 global=attention을 통해 이미지 내의 물체를 잘 나눈 후, decoder는 클래스 및 물체의 Boundary를 잘 추출하도록 attention을 주게 됨

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Decoder Attention

- Decoder attention is fairly local, meaning that is mostly attends to object extremities such as heads or legs.

- Decoder가 Object를 Detection하는 Training 과정을 통해 각 object들의 extremities(말단)에 Attention을 크게 주도록 학습되며,
- Object가 겹치는 경우(occluded) 잘못된 attention을 주지 않도록 학습됨을 확인할 수 있음
- 즉 encoder가 global attention을 통해 이미지 내의 물체를 잘 나눈 후, decoder는 클래스 및 물체의 Boundary를 잘 추출하도록 attention 주게 됨

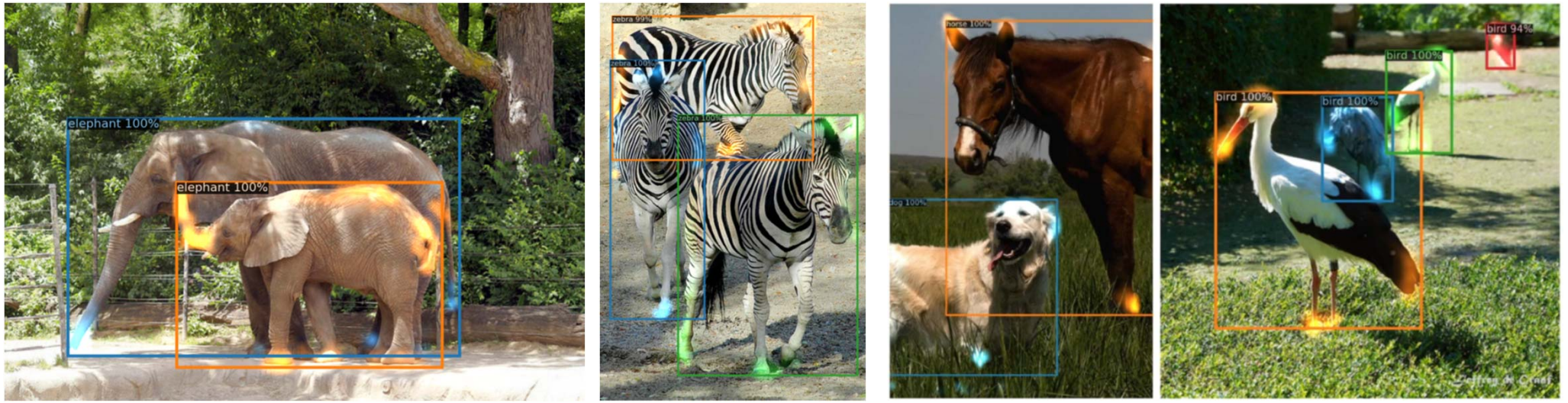


Fig. 6: Visualizing decoder attention for every predicted object (images from COCO val set). Predictions are made with DETR-DC5 model. Attention scores are coded with different colors for different objects. Decoder typically attends to object extremities, such as legs and heads. Best viewed in color.

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Importance of Positional Encodings

Table 3: Results for different positional encodings compared to the baseline (last row), which has fixed sine pos. encodings passed at every attention layer in both the encoder and the decoder. Learned embeddings are shared between all layers. Not using spatial positional encodings leads to a significant drop in AP. Interestingly, passing them in decoder only leads to a minor AP drop. All these models use learned output positional encodings.

spatial pos. enc. encoder	spatial pos. enc. decoder	output pos. enc. decoder	AP	Δ	AP_{50}	Δ
none	none	learned at input	32.8	-7.8	55.2	-6.5
sine at input	sine at input	learned at input	39.2	-1.4	60.0	-1.6
learned at attn.	learned at attn.	learned at attn.	39.6	-1.0	60.7	-0.9
none	sine at attn.	learned at attn.	39.3	-1.3	60.3	-1.4
sine at attn.	sine at attn.	learned at attn.	**40.6**	-	**61.6**	-

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Loss ablations

Table 4: Effect of loss components on AP. We train two models turning off l_1 loss, and GIoU loss, and observe that l_1 gives poor results on its own, but when combined with GIoU improves AP_M and AP_L. Our baseline (last row) combines both losses.

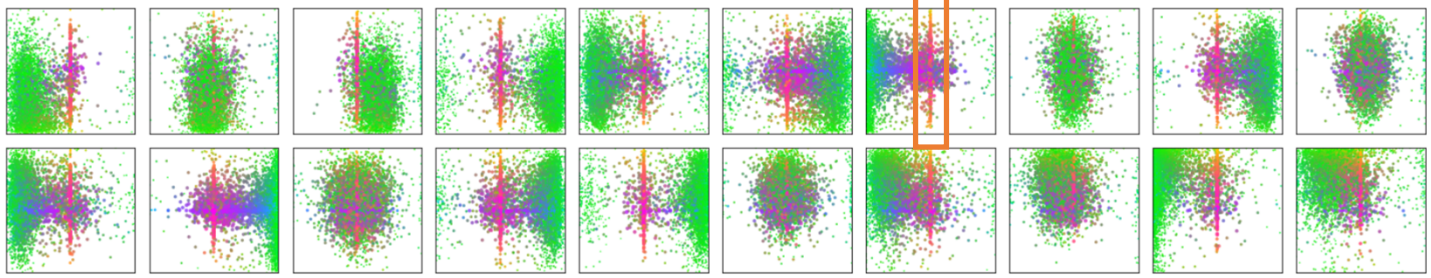
class	ℓ_1	GIoU	AP	Δ	AP_{50}	Δ	AP_S	AP_M	AP_L
✓	✓		35.8	-4.8	57.3	-4.4	13.7	39.8	57.9
✓		✓	39.9	-0.7	**61.6**	0	**19.9**	43.2	57.9
✓	✓	✓	**40.6**	-	**61.6**	-	**19.9**	**44.3**	**60.2**

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Analysis

- Decoder output slot analysis
  - DETR learns different specialization for each query slot.
- Generalization to unseen numbers of instances
  - There is no image with more than 13 giraffes in the training set.
  - This experiment confirms that there is no strong class specialization in each object query.

*수직 선은 COCO 데이터셋에 흔히 존재하는 가로로 긴 큰 물체



*COCO 데이터셋에 대한 decoder slot100개 중 20개 시각화 (빨강:수평으로 큰 Box, 파랑:수직으로 큰 Box, 초록:작은 Box)

Fig. 7: Visualization of all box predictions on all images from COCO 2017 val set for 20 out of total N = 100 prediction slots in DETR decoder. Each box prediction is represented as a point with the coordinates of its center in the 1-by-1 square normalized by each image size. The points are color-coded so that green color corresponds to small boxes, red to large horizontal boxes and blue to large vertical boxes. We observe that each slot learns to specialize on certain areas and box sizes with several operating modes. We note that almost all slots have a mode of predicting large image-wide boxes that are common in COCO dataset.

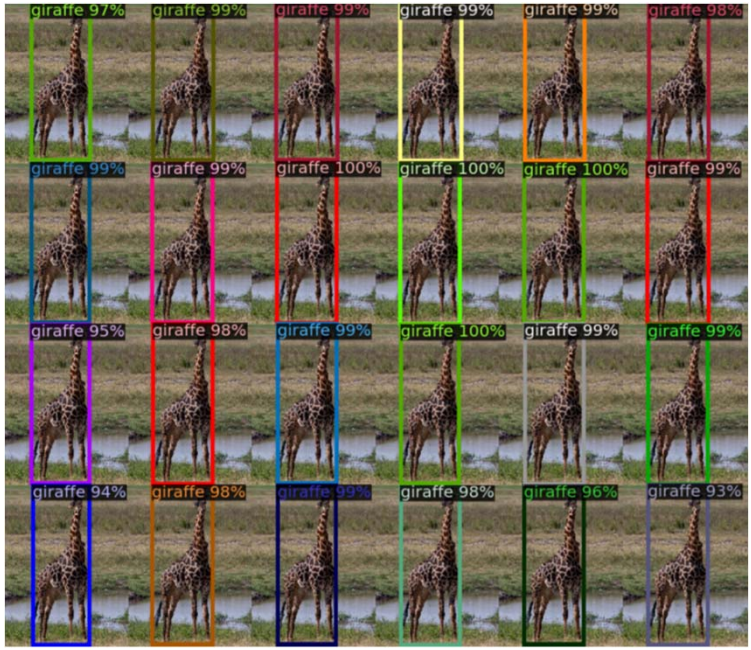


Fig. 5: Out of distribution generalization for rare classes. Even though no image in the training set has more than 13 giraffes, DETR has no difficulty generalizing to 24 and more instances of the same class.

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Increasing the Number of Instances

- While the model detects all instances when up to 50 are visible, it then starts saturating and misses more and more instances. Notably, when the image contains all 100 instances, the model only detects 30 on average, which is less than if the image contains only 50 instances that are all detected.

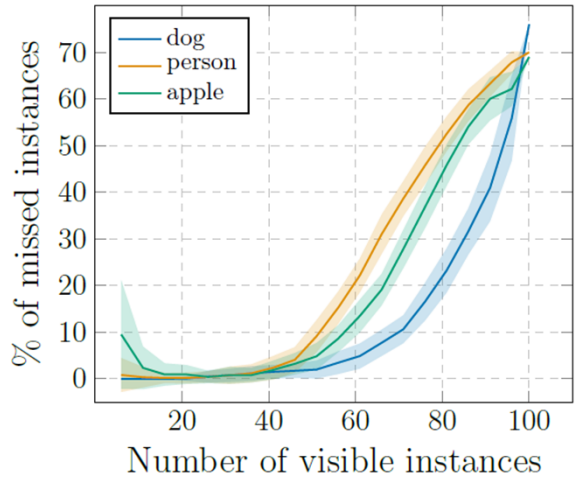
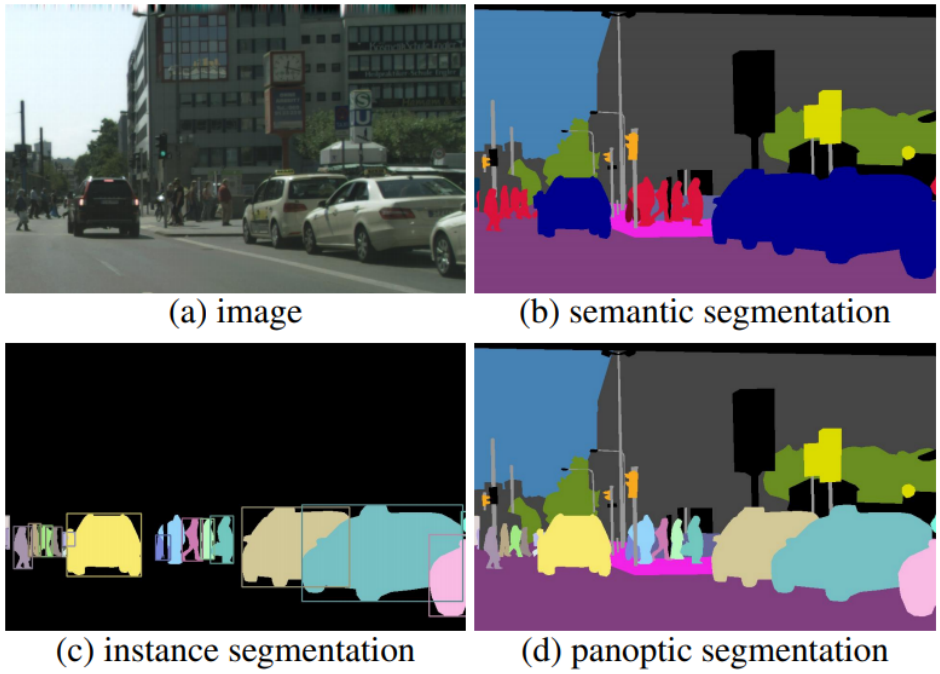


Fig. 12: Analysis of the number of instances of various classes missed by DETR depending on how many are present in the image. We report the mean and the standard deviation. As the number of instances gets close to 100, DETR starts saturating and misses more and more objects

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

DETR for Panoptic Segmentation

- Similarly to the extension of Faster R CNN to Mask R CNN, DETR can be naturally extended by adding a mask head on top of the decoder outputs.



(a) image (b) semantic segmentation

(c) instance segmentation (d) panoptic segmentation

Panoptic Segmentation, A Kirillov et. al., 2018

- Panoptic segmentation은
  - ① 이미지 내의 모든 픽셀을 사전에 정해진 **class**로 분류하는 **semantic segmentation**과,
  - ② 동일한 **class**내에서도 서로 다른 객체를 구분하는 **instance segmentation**을 합친 개념
- Faster R-CNN에 mask head를 더해 Mask R-CNN 구조를 구축했듯이, "**DETR의 decoder 결과물에 mask head를 추가하여 segmentation task까지 확장할 수 있다**!"

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

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

DETR for Panoptic Segmentation

① DETR의 decoder가 출력한 결과물(object queries)을 encoder에 의해 encoding 된 결과물과 attention 수행

② ①에서 얻은 attention map을 기반으로 여러 resolution의 feature map과의 연산을 통해 masked image를 얻음

③ 해당 masked image의 픽셀마다 argmax를 적용하여 특정 object로의 분류 수행

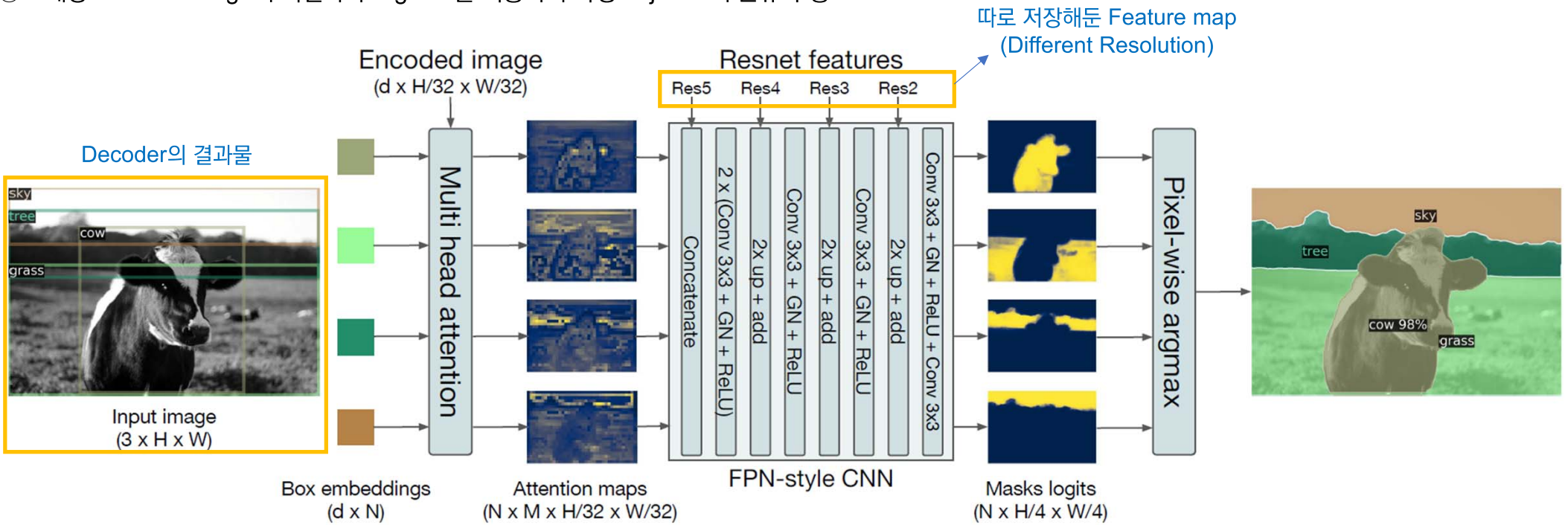


Fig. 8: Illustration of the panoptic head. A binary mask is generated in parallel for each detected object, then the masks are merged using pixel-wise argmax.

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Panoptic Segmentation Results

Table 5: Comparison with the state-of-the-art methods UPSNet [51] and Panoptic FPN [18] on the COCO val dataset We retrained Panoptic FPN with the same data augmentation as DETR, on a 18x schedule for fair comparison. UPSNet uses the 1x schedule, UPSNet-M is the version with multiscale test-time augmentations

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

- PQ(Panoptic Quality) 기준 당시 SOTA 모델(PanopticFPN)을 넘는 성능을 기록
- 특히 stuff(sky, tree 등)에 대해 타 모델 대비 훨씬 높은 성능(PQ^{st})을 기록하였으며, encoder attention의 global reasoning이 이러한 성능에 큰 영향을 미쳤을 것으로 예상

[발췌 자료] PR-284: End-to-End Object Detection with Transformers(DETR), Jin Won Lee

Panoptic Segmentation Results

Fig. 9: Qualitative results for panoptic segmentation generated by DETR-R101. DETR produces aligned mask predictions in a unified manner for things and stuff.