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

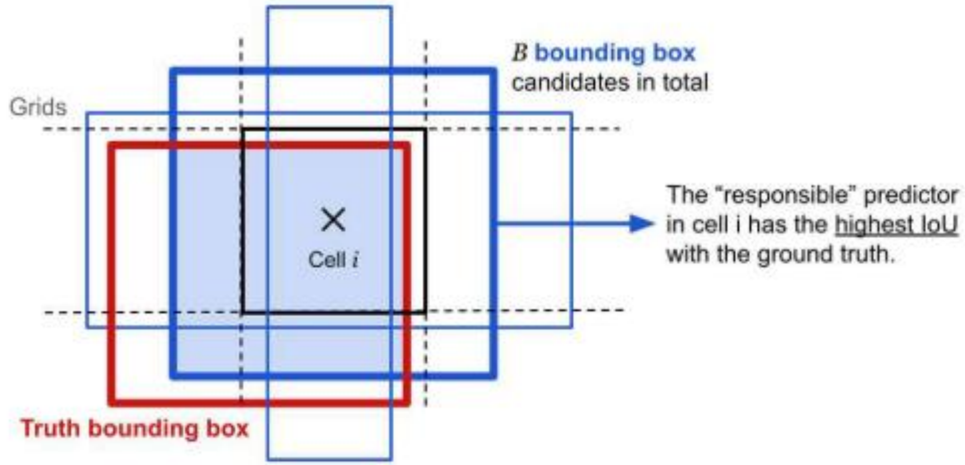
ICLR 2021

발표자: 김태성

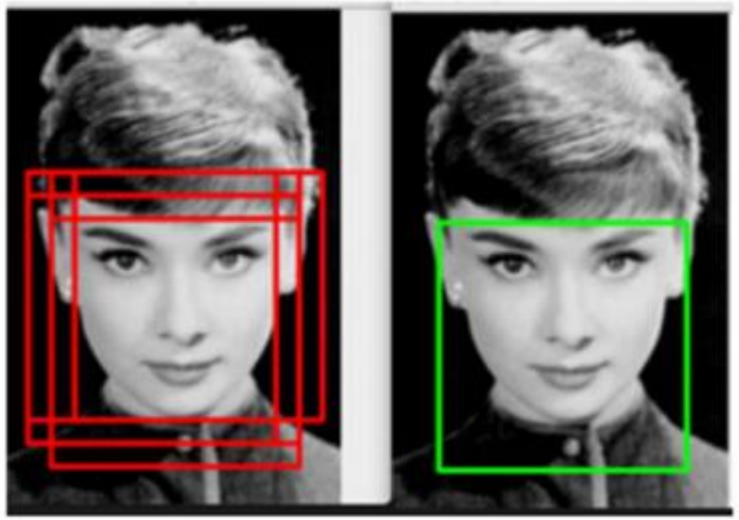
Paper link: https://openreview.net/forum?id=gZ9hCDWe6ke
Github link: https://github.com/fundamentalvision/Deformable-DETR

Introduction

- DETR (End-to-End Object **De**tection with **Tr**ansformers)
  - DETR은 다양한 hand-designed components들 없이 완전히 end-to-end 로 학습하여 MS COCO dataset에 Faster RCNN과 비슷한 성능을 내는 것에 성공함.



Anchor box

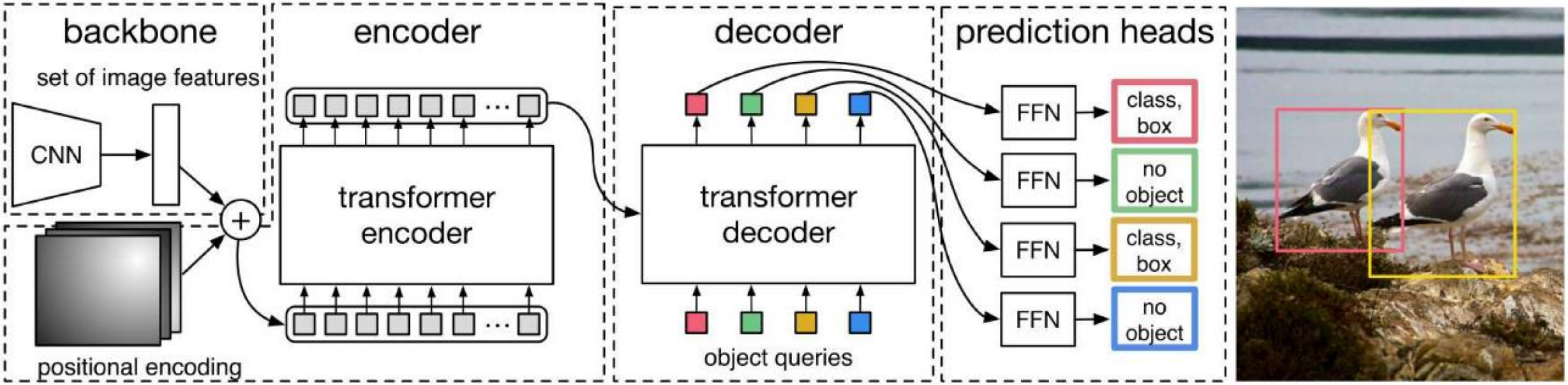


Non-maximum suppression

https://github.com/awesome-davian/Paper-study/blob/master/paper-list(2020).md

Introduction

- DETR (End-to-End Object **De**tection with **Tr**ansformers)



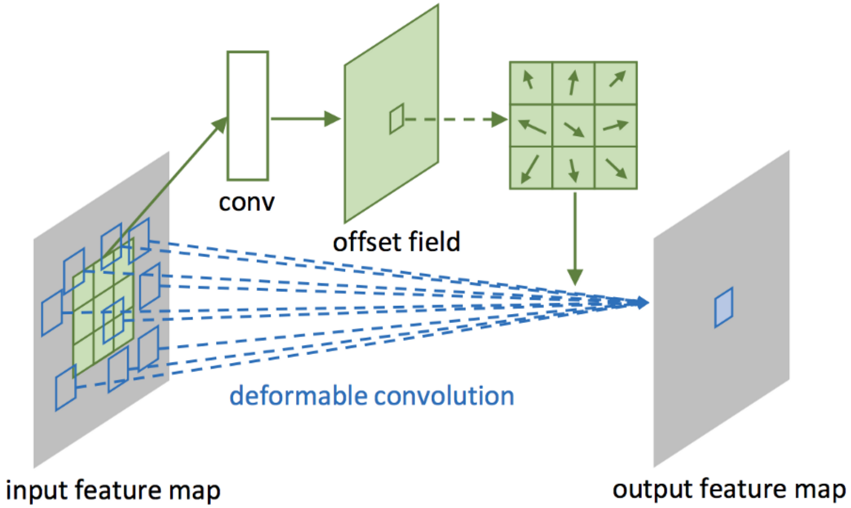
Introduction

- DETR 의 문제점
  1. Training time
     - MS COCO dataset에 대해 500 에포크 이상 학습해야 함. (약 2~3일 소요)
     - 이는 Faster R-CNN 보다 10~20배 느린 속도임.

  2. Low performance on small objects
     - 기존의 Convolutional neural network 들은 high-resolution feature 로부터 정보를 얻는 것이 가능했음
     - 하지만 DETR의 경우, attention 계산을 위해 pixel 수의 제곱에 비례하는 computation 이 필요하므로, high-resolution featur를 사용하는 것이 거의 불가능함.

Introduction

- Deformable DETR
  - 두가지 모델의 장점을 합친 모델임
    - Deformable Conv
      - Sparse spatial sampling
    - DETR
      - Relation modeling between pixels



Deformable Conv

- Deformable attention
  - Feature pyramid network 없이도 multi-scale feature 로부터 정보를 얻을 수 있음

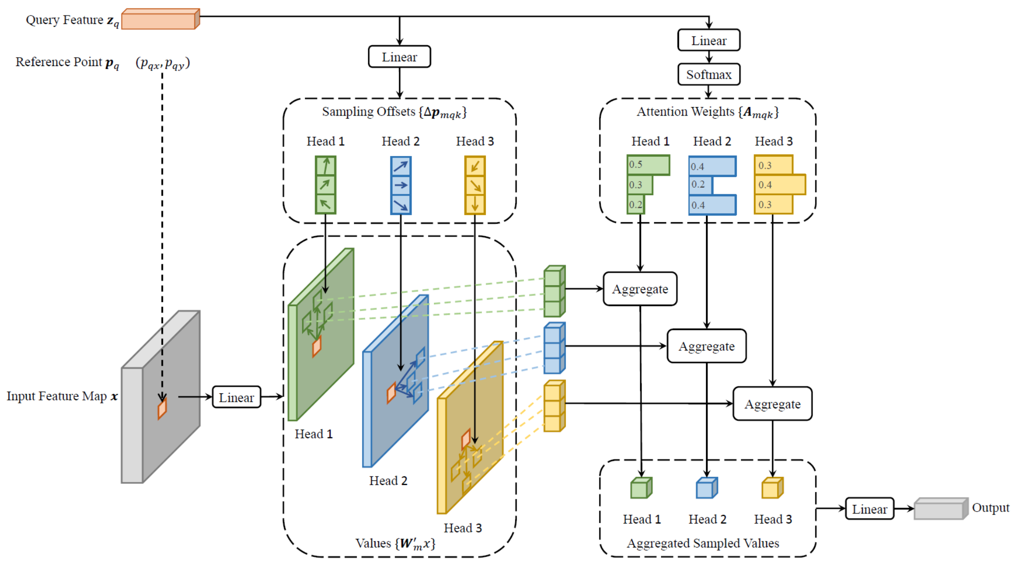
Introduction

- Deformable DETR
  - DETR과 비교해 10배 빠른 수렴 속도
  - DETR과 비교해 MS COCO dataset에 대해 더 뛰어난 성능
  - 새롭게 제안한 two-stage Deformable DETR을 사용하면 더 높은 성능을 보임

Method

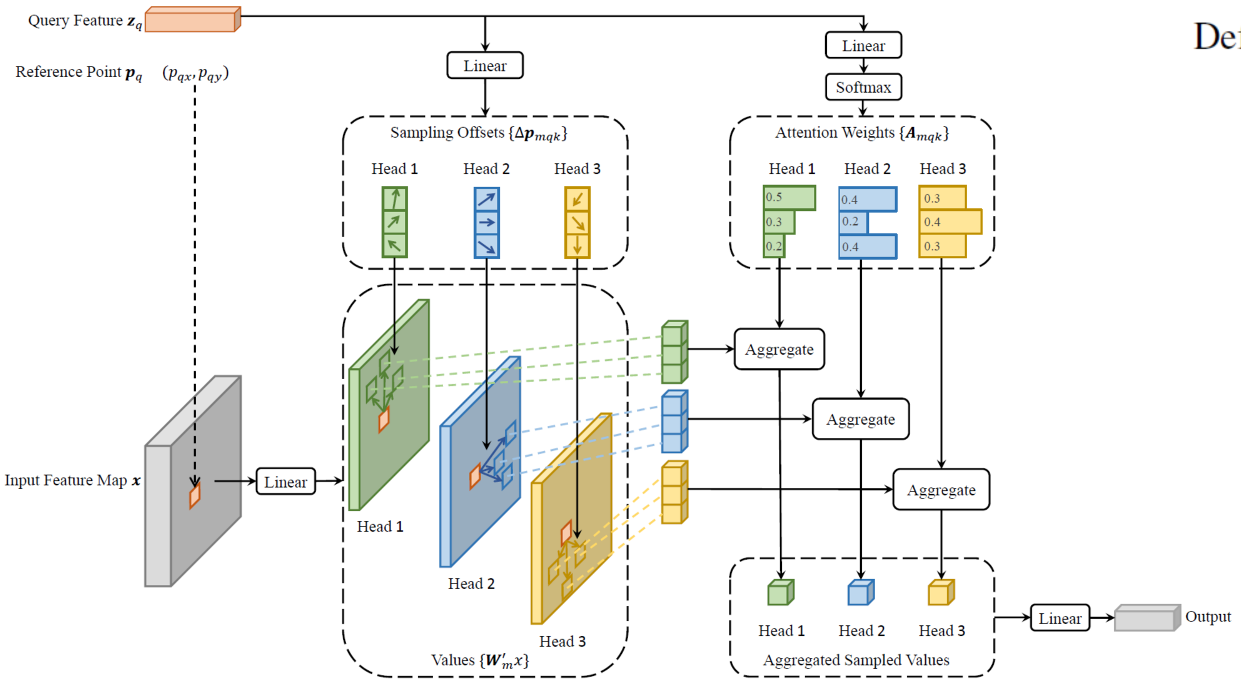
- Deformable Attention Module
  - Transformer attention 을 이미지에 적용 시 생기는 문제점은, spatial 정보를 잊게 된다는 점임

➢이를 해결하기위해 Deformable attention module 제안



Method

- Deformable Attention Module



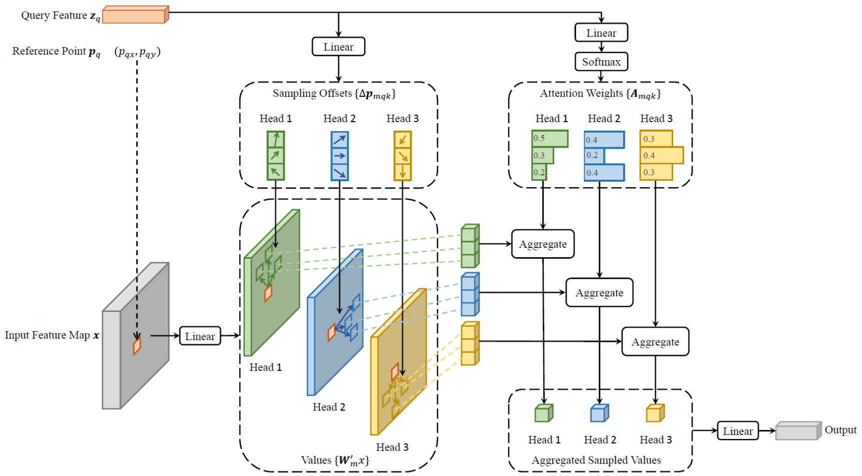
M: head 개수
K: sampled key 개수 (sampled pixel 개수)
Q: query 개수

$$\mathrm{DeformAttn}(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \Big[\sum_{k=1}^{K} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk}) \Big]$$

1. Input feature map → Multi-head values
2. Query feature → offsets
3. Query feature → attention weights
4. Weighted sum (Aggregate)
5. Linear

Method

- Deformable Attention Module



M: head 개수
K: sampled key 개수 (sampled pixel 개수)
Q: query 개수

$$\mathrm{DeformAttn}(z_q, p_q, x) = \sum_{m=1}^{M} W_m \Big[\sum_{k=1}^{K} A_{mqk} \cdot W'_m x(p_q + \Delta p_{mqk})\Big]$$

- 기존 DETR attention module보다 Computation complexity 측면에서 이득임

Offset, attention weight 계산: $O(3N_qCMK)$

DeformAttn 계산: $O(N_qC^2 + N_qKC^2 + 5N_qKC)$

※Deformable DETR의 경우, Encoder: $N_q = HW >> C$, N_k

Decoder: $HW >> C, N_q, N_k$

단순 attention: $O(N_qC^2 + N_kC^2 + N_qN_kC)$

※일반적인 이미지 attentnion의 경우, $N_q = N_k = HW >> C$

※DETR의 경우, Encoder: $N_k = N_q = HW >> C$,

Decoder: $N_k = HW >> C, N_q$

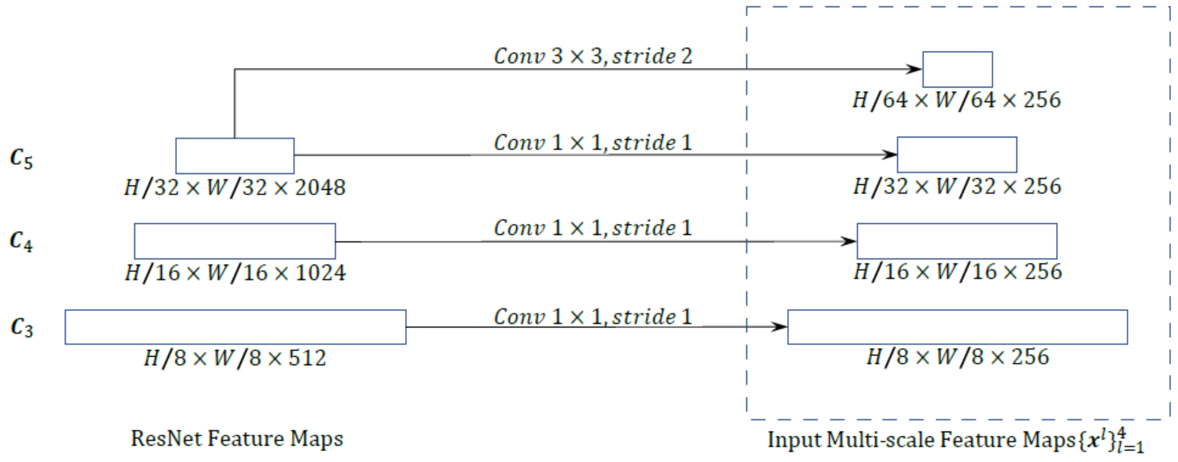
Method

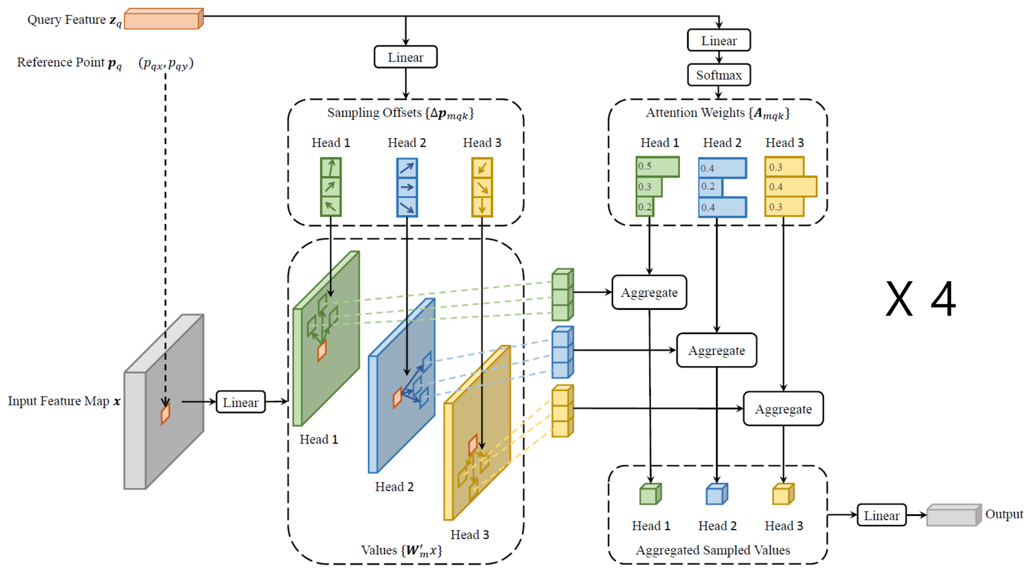
- Multi-scale Deformable Attention Module

$$\mathrm{DeformAttn}(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \Big[\sum_{k=1}^{K} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk}) \Big]$$

$$\mathrm{MSDeformAttn}(\boldsymbol{z}_q, \hat{\boldsymbol{p}}_q, \{\boldsymbol{x}^l\}_{l=1}^{L}) = \sum_{m=1}^{M} \boldsymbol{W}_m \Big[\sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}^l(\phi_l(\hat{\boldsymbol{p}}_q) + \Delta \boldsymbol{p}_{mlqk}) \Big]$$

M: head 개수
K: sampled key 개수 (sampled pixel 개수)
Q: query 개수
L: multi-scale feature map 개수
$\phi_l(\hat{\boldsymbol{p}}_q)$: Unnormalize 함수

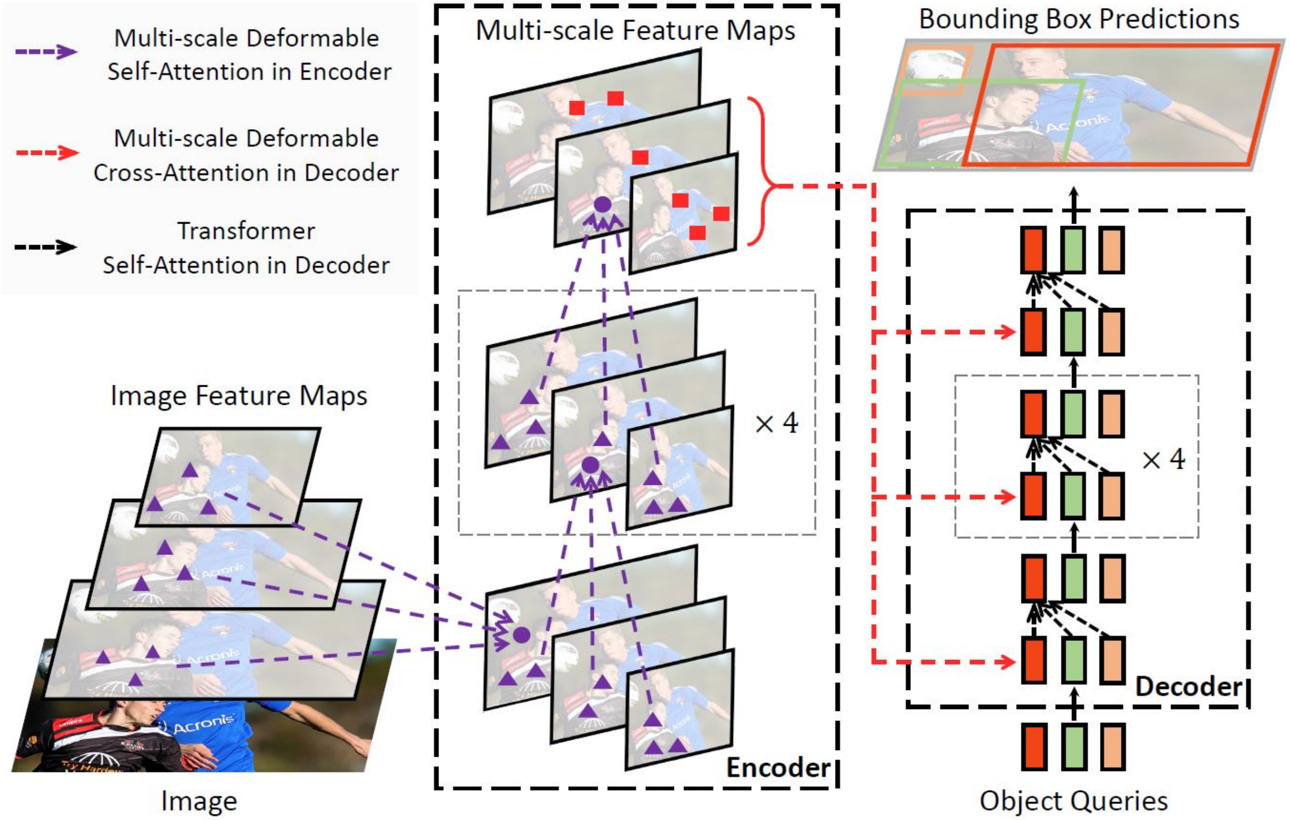




X 4

Method

- Final model



Hungarian loss[1] 사용하여 학습

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

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

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

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

1) Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In ECCV, 2020.

Method

- Additional techniques
  - Iterative refinement
    - 각 decoder 레이어가 이전 레이어의 예측 값을 수정하는 형식[1)]
  - Two-Stage Deformable DETR
    - Stage 1: Region proposal network를 사용하여 모든 pixel에 대해 bounding box 예측
      - Region proposal network에 Multi-scale deformable attention 사용 (decoder 가 없으므로 self attention만 적용)
      - DETR에서 제안한 Hungarian loss 사용하여 학습
    - Stage 2: 스코어가 높은 bbox들의 pixel coordinates를 decoder object queries 로 사용하여, Encoder-Decoder 구조의 Deformable DETR 학습

1) RAFT: Recurrent All-Pairs Field Transforms for Optical Flow, Zachary Teed and Jia Deng, ECCV 2020

Experiments

- MS COCO object detction

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.

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

- DC5: conv5 layer의 stride를 삭제하여 resolution을 증가시킨 모델
- DETR-DC5$^+$: DETR-DC5에 Focal loss 를 추가한 모델

Experiments

- Ablation study

Table 2: Ablations for deformable attention on COCO 2017 val set. "MS inputs" indicates using multi-scale inputs. "MS attention" indicates using multi-scale deformable attention. K is the number of sampling points for each attention head on each feature level.

MS inputs	MS attention	K	FPNs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
✓	✓	4	FPN (Lin et al., 2017a)	43.8	62.6	47.8	26.5	47.3	58.1
✓	✓	4	BiFPN (Tan et al., 2020)	43.9	62.5	47.7	25.6	47.4	57.7
		1	w/o	39.7	60.1	42.4	21.2	44.3	56.0
✓		1		41.4	60.9	44.9	24.1	44.6	56.1
✓		4		42.3	61.4	46.0	24.8	45.1	56.3
✓	✓	4		43.8	62.6	47.7	26.4	47.1	58.0

- FPN과 BiFPN이 성능 향상에 거의 영향을 미치지 못함을 보여줌.

Experiments

- Comparison with SOTA models

Table 3: Comparison of Deformable DETR with state-of-the-art methods on COCO 2017 test-dev set. "TTA" indicates test-time augmentations including horizontal flip and multi-scale testing.

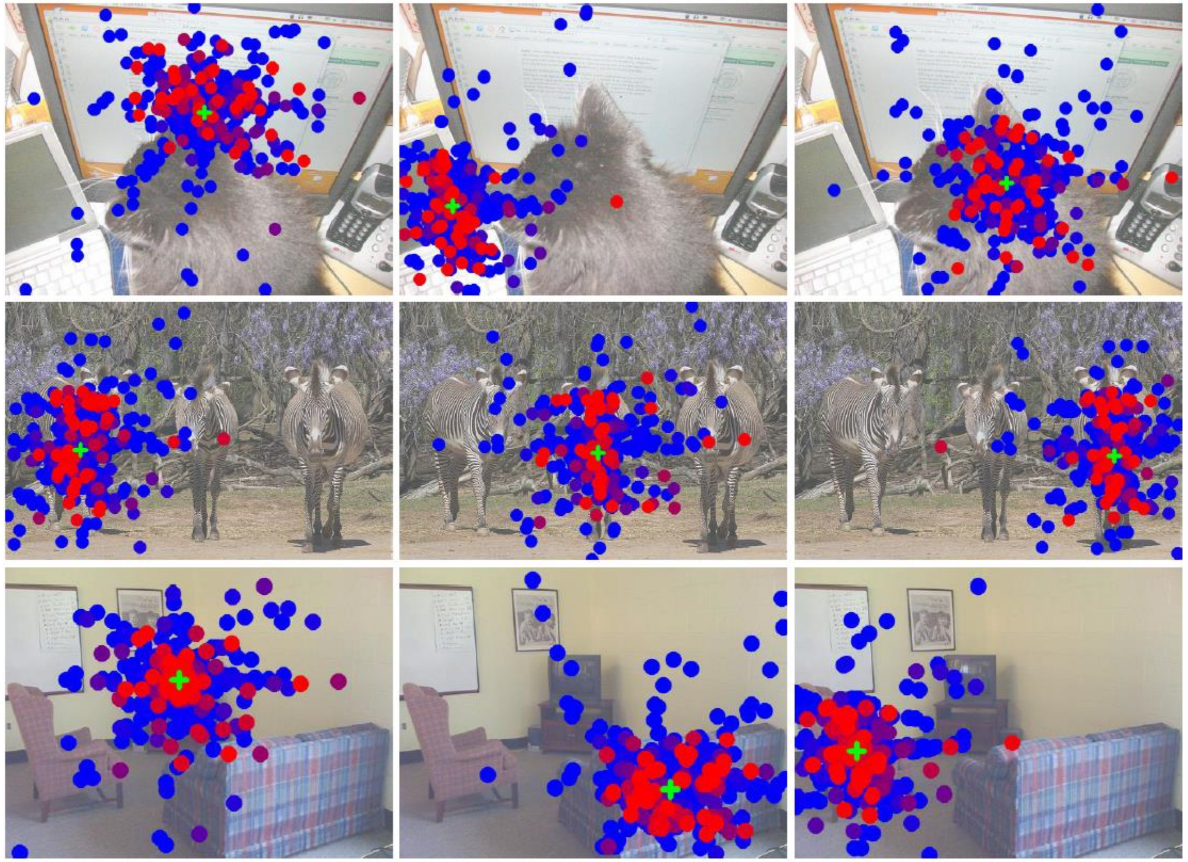
Method	Backbone	TTA	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
FCOS (Tian et al., 2019)	ResNeXt-101		44.7	64.1	48.4	27.6	47.5	55.6
ATSS (Zhang et al., 2020)	ResNeXt-101 + DCN	✓	50.7	68.9	56.3	33.2	52.9	62.4
TSD (Song et al., 2020)	SENet154 + DCN	✓	51.2	71.9	56.0	33.8	54.8	64.2
EfficientDet-D7 (Tan et al., 2020)	EfficientNet-B6		52.2	71.4	56.3	-	-	-
Deformable DETR	ResNet-50		46.9	66.4	50.8	27.7	49.7	59.9
Deformable DETR	ResNet-101		48.7	68.1	52.9	29.1	51.5	62.0
Deformable DETR	ResNeXt-101		49.0	68.5	53.2	29.7	51.7	62.8
Deformable DETR	ResNeXt-101 + DCN		50.1	69.7	54.6	30.6	52.8	64.7
Deformable DETR	ResNeXt-101 + DCN	✓	52.3	71.9	58.1	34.4	54.4	65.6

- Iterative bounding box refinement와 two-stage Deformable DETR을 모두 사용함.
- Backbone network를 ResNeXt-101, Deformable convnet v2로 교체함.
- 최종적으로, SOTA 모델들과도 비슷한 성능을 보임.

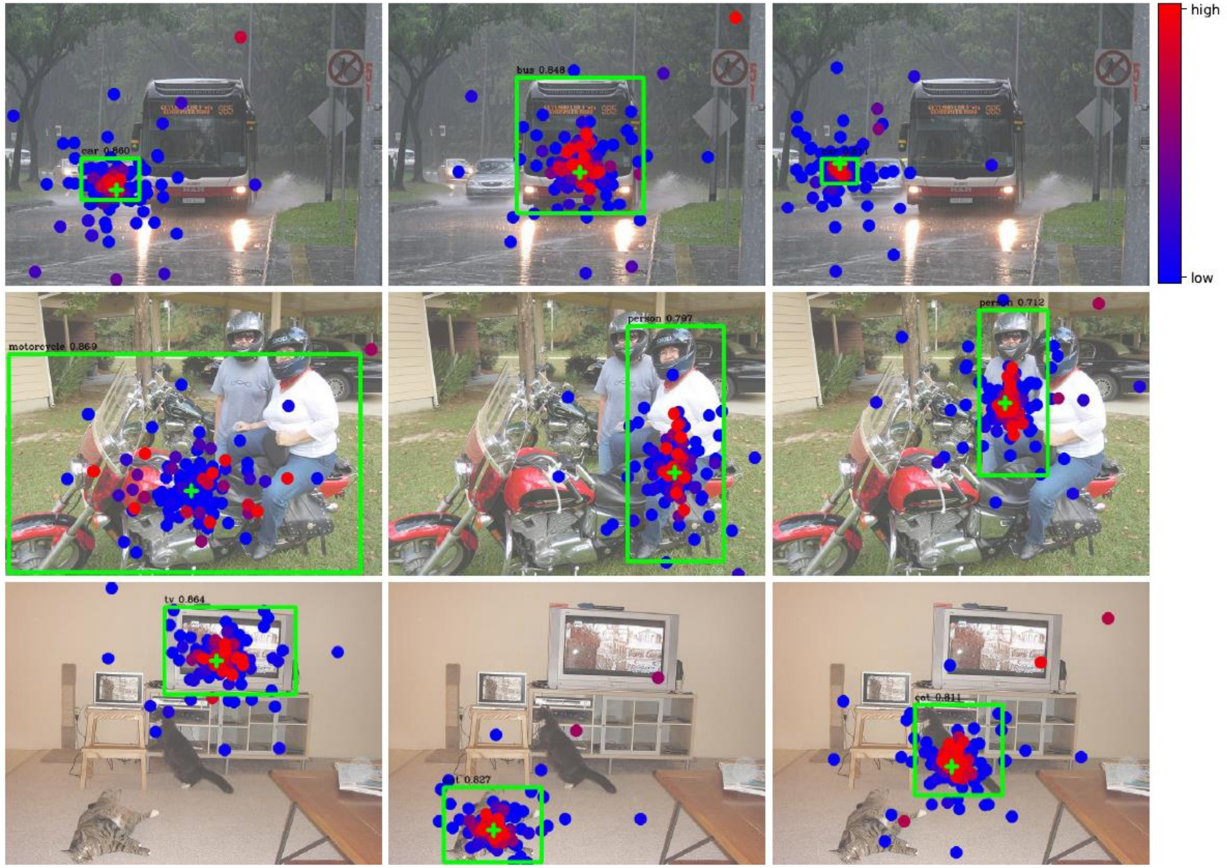
Experiments

- Attention visualization

Green cross marker: Query
blue~red points: Sampled key



(a) multi-scale deformable self-attention in encoder



(b) multi-scale deformable cross-attention in decoder

Conclusion

- 기존 DETR은 object detection 을 set prediction task로 전환하여 simplicity를 증가시키는 것에 기여하였으나, 아래와 같은 Transformer network의 단점들을 가지고 있었음
  - Slow convergence speed
  - High computational complexity
  - Low performance on small objects

- Deformable DETR은
  - Deformable convolution의 아이디어에 기반하여, attention key를 sampling하는 방법을 제안함으로써, slow convergence, high computational complexity를 해결
  - Multi-scale deformable DETR을 통해 performance 끌어올림
  - Iterative refinement, two-stage mechanism 을 통해 추가적인 성능 향상 (SOTA 근접)