DETR

(End-to-End Object Detection with Transformers)

김동원

목차

ViT

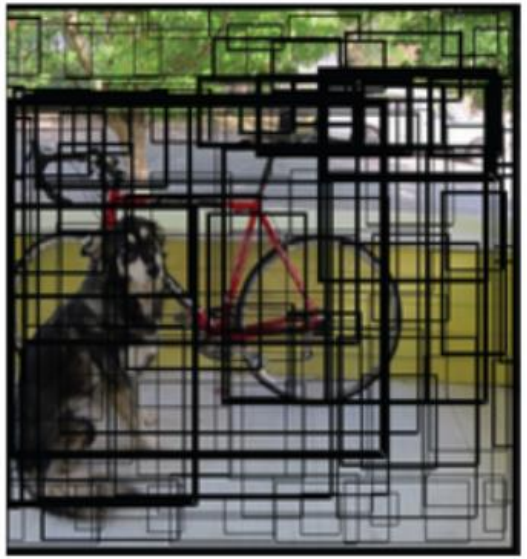
DETR(End-to-End Object Detection with Transformers)

Transformer구조를 이용하여 object detection task를 수행하기 위한 간단한 모델을 제시

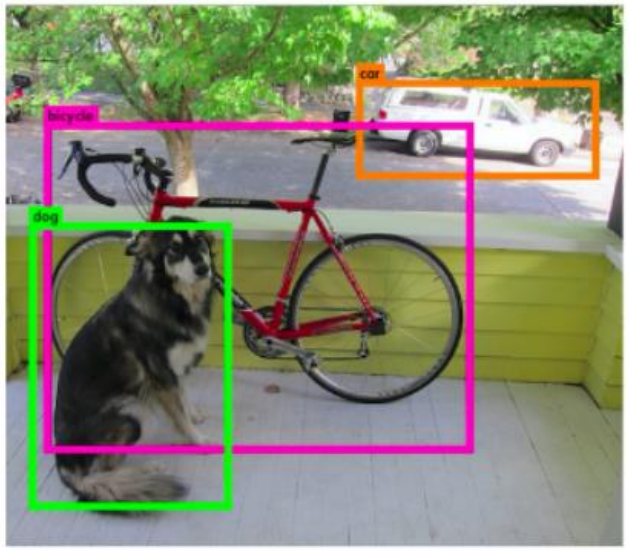
NMS를 사용하지 않아 직접적인 해결이 가능하다

NMS(non-maximum suppression)이란?

전통적 object detection 기법에서 여러 bounding box들 같은 클래스에 대해 예측한 박스들 중 가장 신뢰도가 높은 bounding box만 남기는 방법



Multiple Bounding Boxes



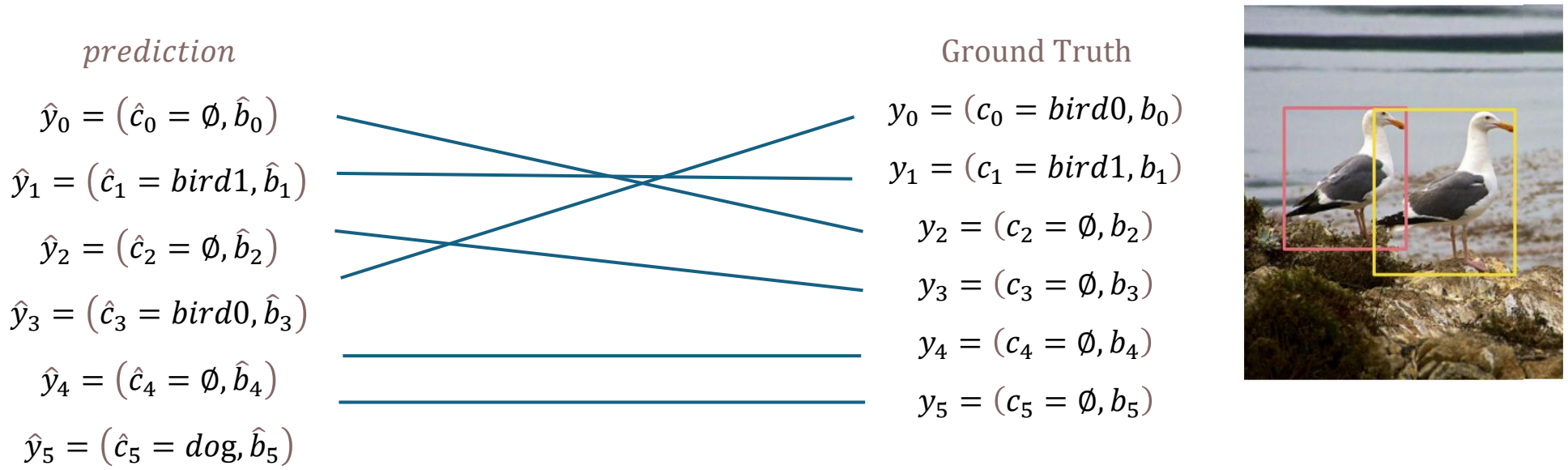
Final Bounding Boxes

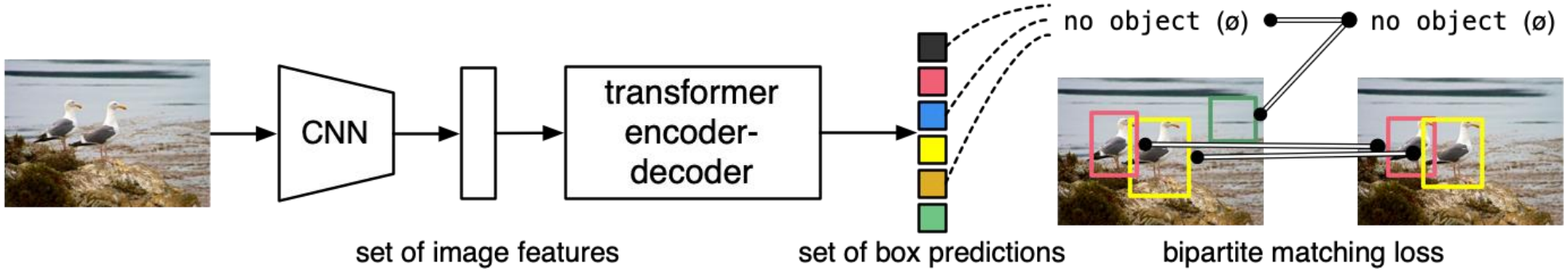
DETR(End-to-End Object Detection with Transformers)

Bipartite matching(이분매칭)

DETR에서는 정해진 개수 만큼의 (class,box)로 이루어진 slot으로 예측을 하고, 각 대상에 대해 각 하나의 중복없는 예측만 진행하고, $\emptyset(no\ object)$에 대한 예측도 포함된다

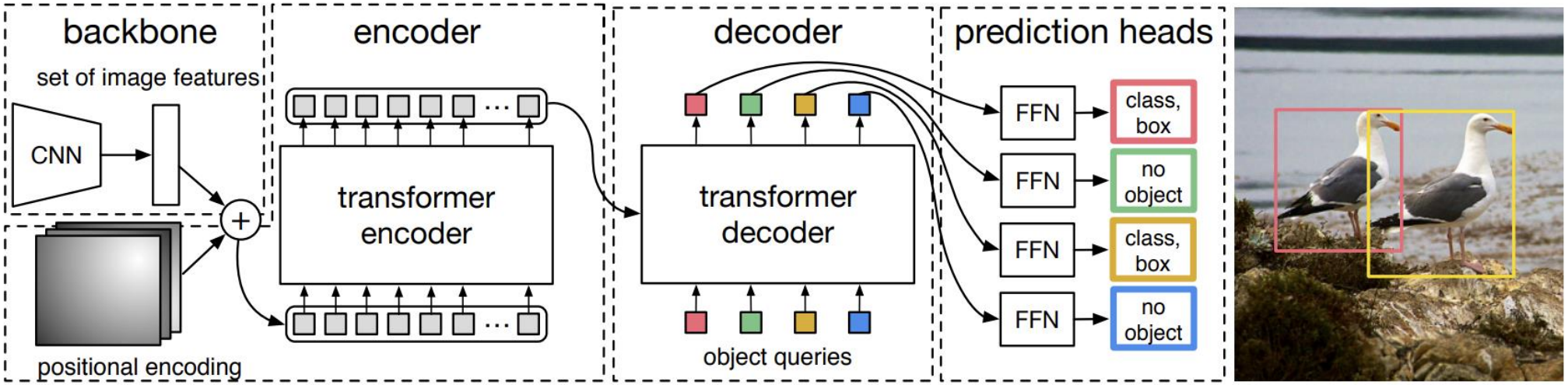
예시)





DETR(End-to-End Object Detection with Transformers)

DETR 구조



사전학습 된 CNN모델로 이미지의 피쳐맵을 추출하고, positional encoding을 추가하여 transformer encoder의 입력하고, 이미지의 전반적인 특징을 학습을 encode에서 진행하고 decoder에서 예측하고자 하는 물체의 개수만큼의 object query를 따라 출력을 하여 Feed forward neural network를 통해 class와 bounding box를 제시하게 된다.

이미지 내 물체의 존재 여부와 bounding box를 제시한다

DETR(End-to-End Object Detection with Transformers)

DETR 구조

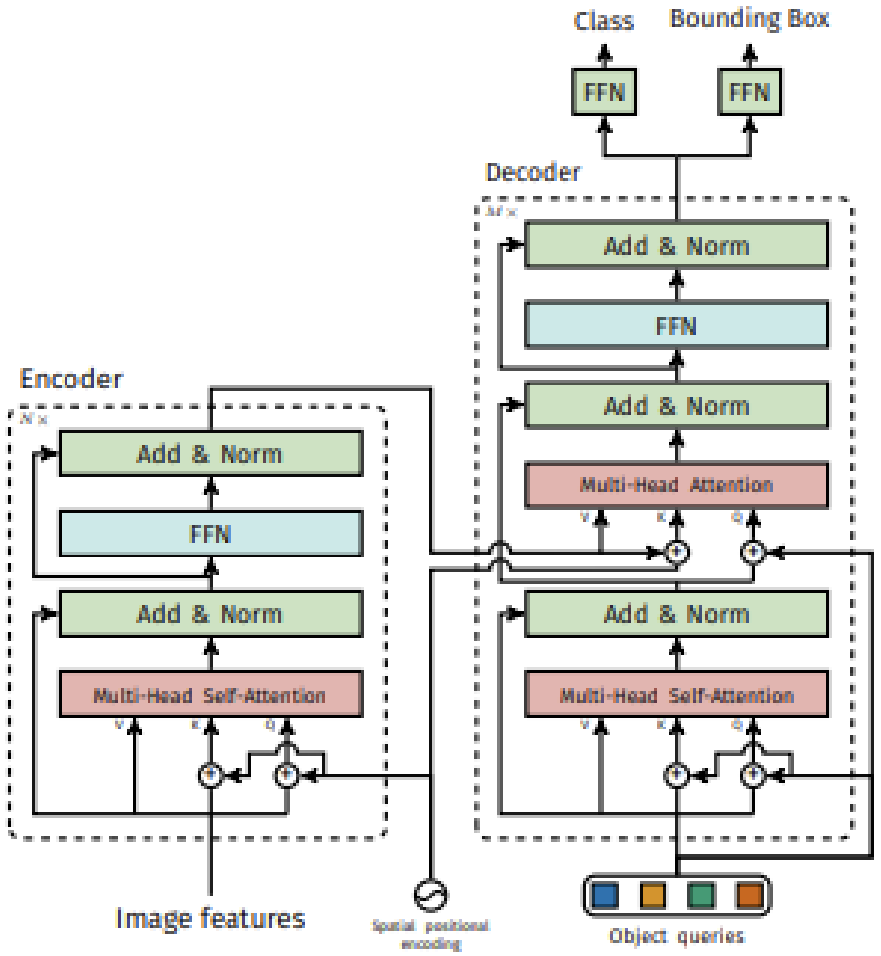


Fig. 10: Architecture of DETR's transformer. Please, see Section A.3 for details.

DETR(End-to-End Object Detection with Transformers)

Loss

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

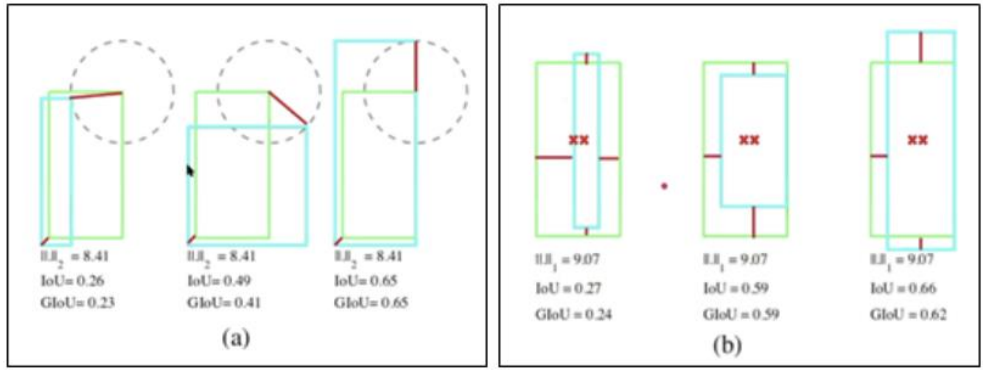
$$\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) = -\mathbb{1}_{\{c_i \neq \varnothing\}} \hat{p}_{\sigma(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\sigma(i)})$$

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

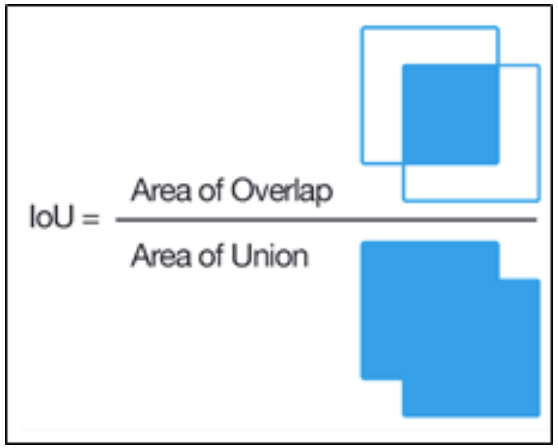
$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}}(i)) \right]$$

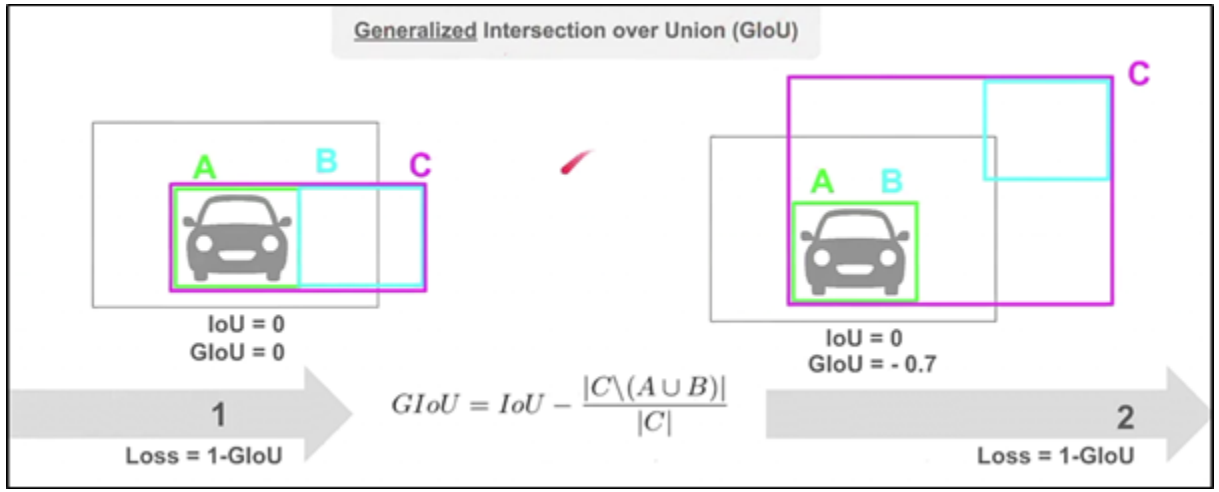
DETR(End-to-End Object Detection with Transformers)

IoU(Intersection over Union), GIoU(Generalized-IoU)



L1, L2 loss만 사용할 경우 박스의 크기가 ground truth에 비해 큰지 작은지 알지 못하기 때문에 IoU를 추가하여 사용한다.





DETR(End-to-End Object Detection with Transformers)

실험

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**

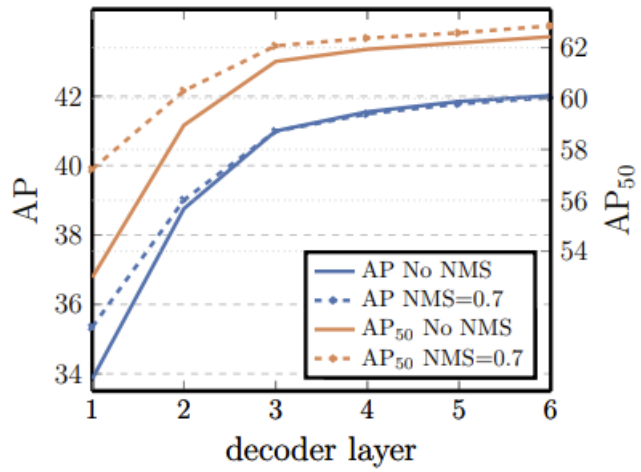
큰 이미지에 비해 작은 크기의 이미지에서는 성능이 떨어진다

#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

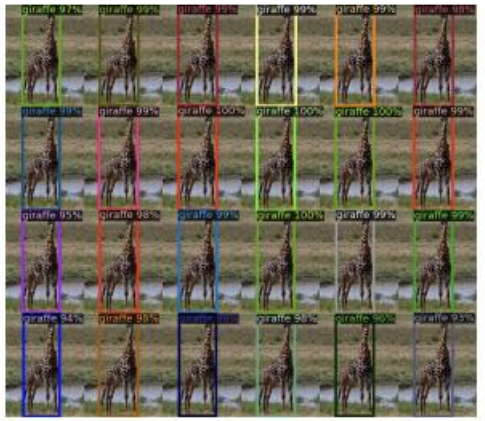
Encoder의 layer가 증가할 수록 성능이 향상되는 것을 알 수 있다.

DETR(End-to-End Object Detection with Transformers)

실험



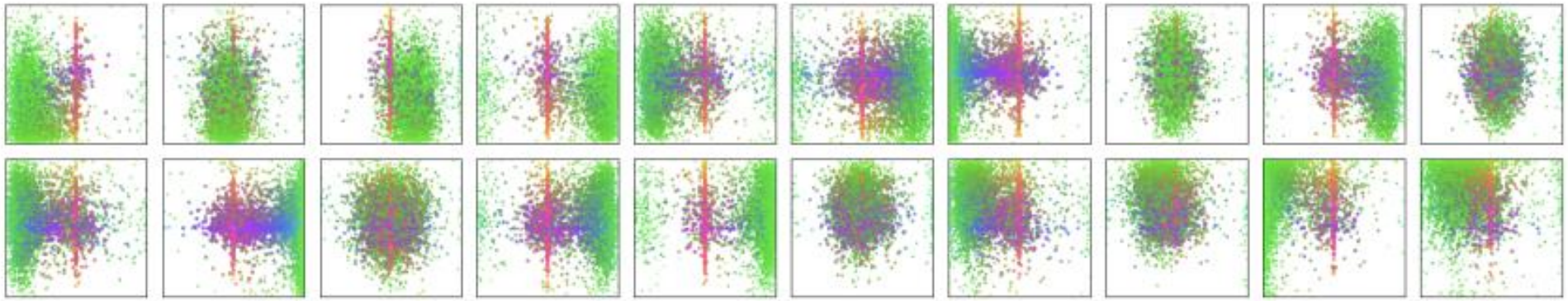
NMS를 적용시켰을 경우 성능이 약간 상승하지만
NMS를 사용하지 않아도 충분한 성능이 나온다



이미지 합성을 통해 동일한 클래스가 13개 이상 있는 이미지를 만들어서 DETR의 성능을 확인 하였고, 잘 인식하였다.

DETR(End-to-End Object Detection with Transformers)

분석



각각의 object query의 slot을 bounding box의 유형을 의미하는 그림으로 이를 보면 각각의 slot이 특정 위치의 특정 크기에 대해 학습하는 것을 알 수 있다.

참고자료link

IoU, GIoU

NMS 설명, 이미지

DETR 설명예시