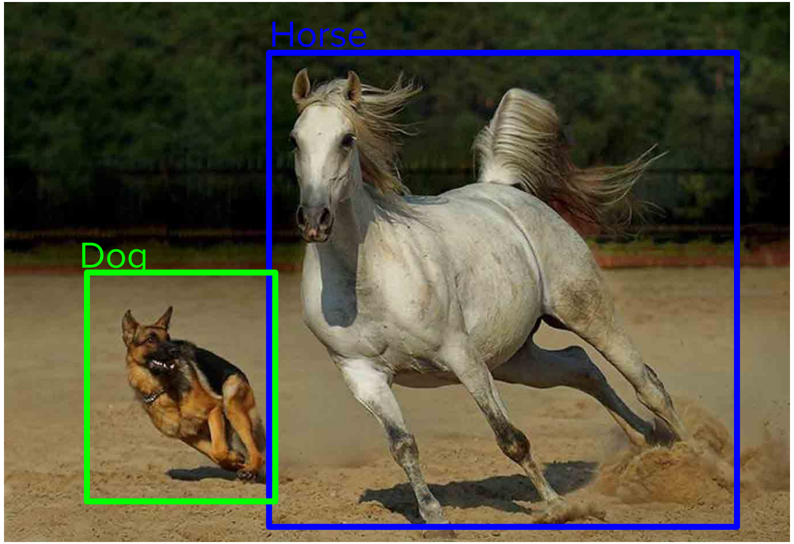
End to End Object Detection with Transformers

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko - Facebook AI
(ECCV 2020)

Object Detection

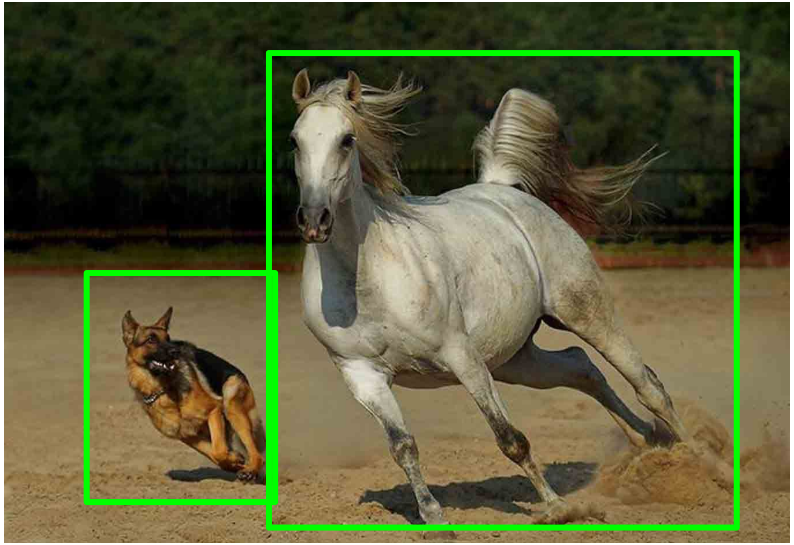
Tasks:

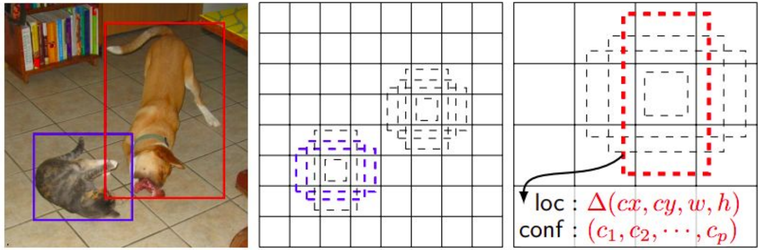
- Create bounding boxes around objects
- Classify Objects



Issues in Object Detection

- Set Prediction Problem
    - Unbounded number of distinct elements
    - Invariant to permutations
- Prior methods doesn't solve the problem directly and produce large set of proposals (10K-100K)
- Hand crafted post processing is required, such as Non Maximal Suppression
- Inductive bias and failures in OOD cases





(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

How DETR Innovates?

(Detection Transformer)

Learning set prediction directly

by

Conjunction of bipartite matching and transformers

Bipartite Matching Loss

- One to one matching with lowest cost

- Hungarian Algorithm: O(n^3)

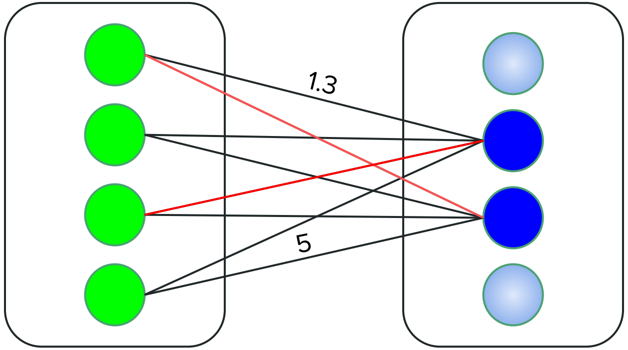
- Matching Cost

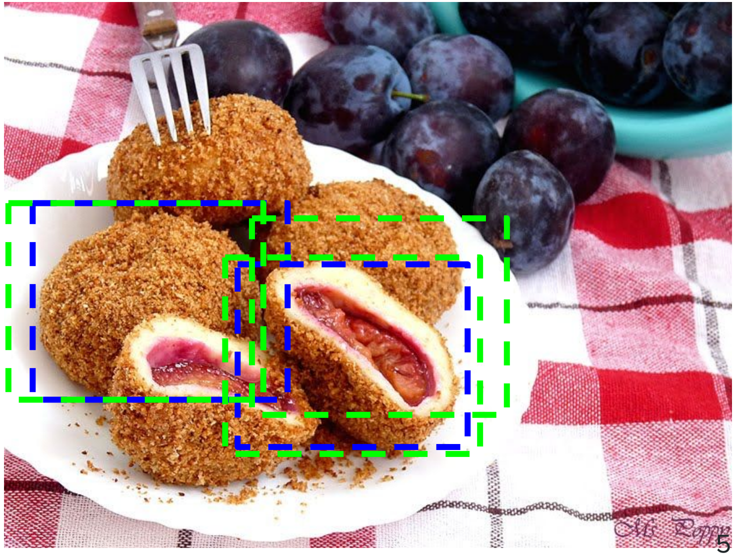
  Cost = class loss + BB loss

- Total Loss

  Loss = sum over used costs

- Can't be used in anchor based methods because of complexity O((10K)^3) ~ O(10^12)

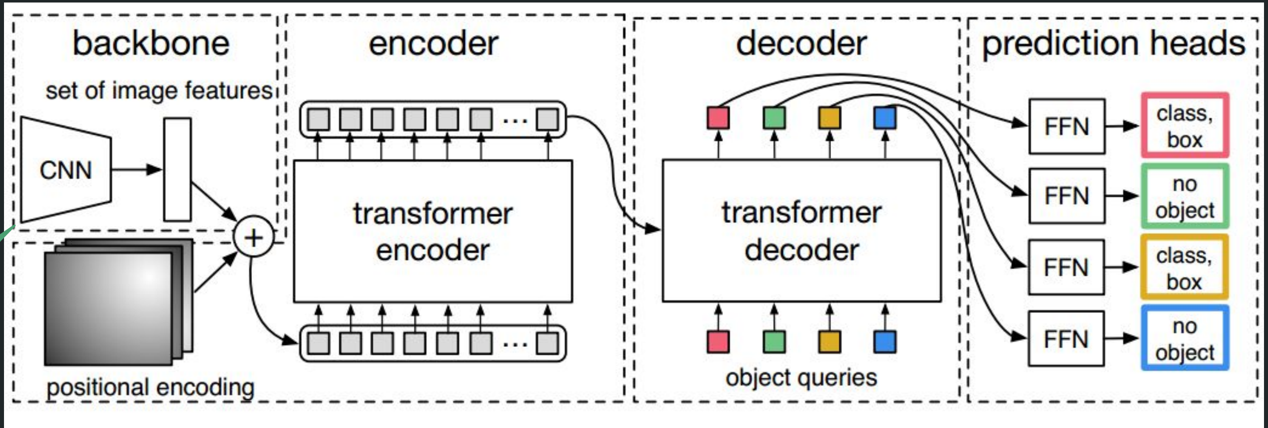




DETR Architecture

$CxH_0/32xW_0/32$ $d'xHW$ 100

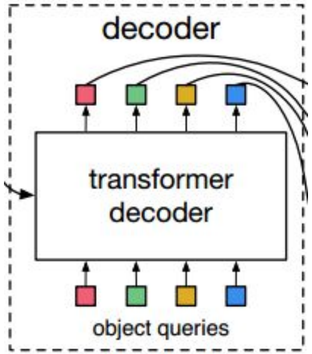
$3xH_0xW_0$



ImageNet-pretrained ResNet

$dxH_0/32xW_0/32$

Decoder Object Queries



- Transformer is permutation invariant, how can we enforce different predictions?
- In translation there is an implied order
- In set prediction implying order might result in bad results (Order Matters: Sequence to Sequence for Sets - Vinyals et al)
- Solution: Learn positional embeddings
- **Does it imply problems with OOD?**

Tweaks

- Resize of input images (Facebook Research-Detectron2)
    - Smaller size between 480 and 800 pixels
    - Longest size at most 1333 pixels

- Random crop in training (add 1 AP)

- Optimize for AP (add 2 AP) -

  At inference they replace the empty slots with the second highest scoring class. (Increasing positive examples?)

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**

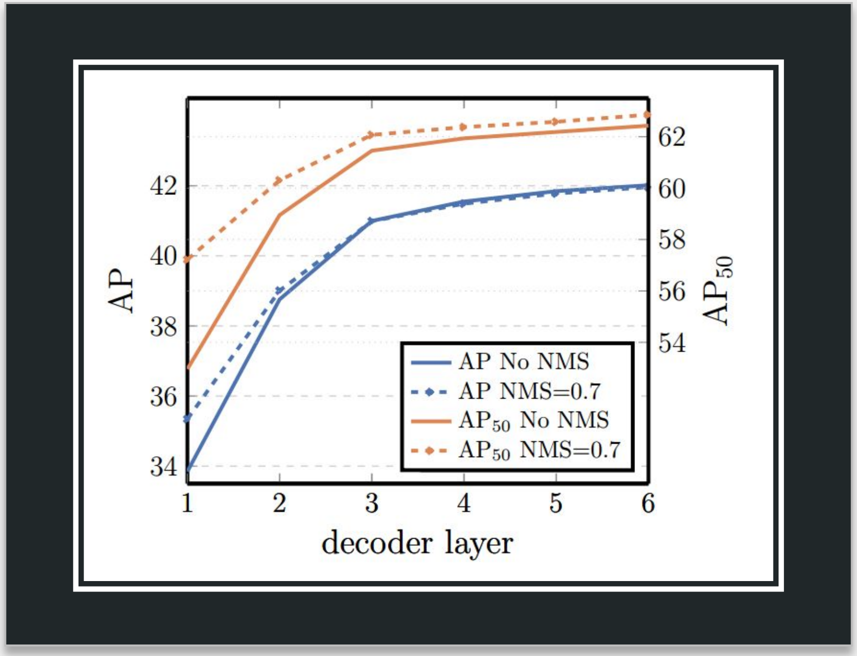
- Good results on large objects
- Not so good on small objects
- Similar amount of parameters

Is NMS needed?

- By solving set prediction directly model shouldn't need NMS

- First attention layer doesn't compute cross correlations, thus NMS helps

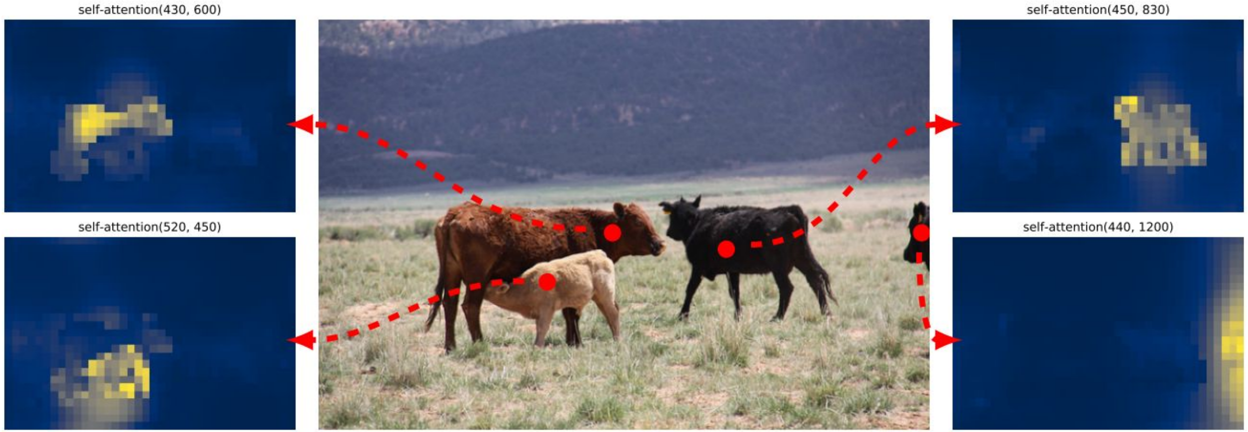
- Results agree with this assumption

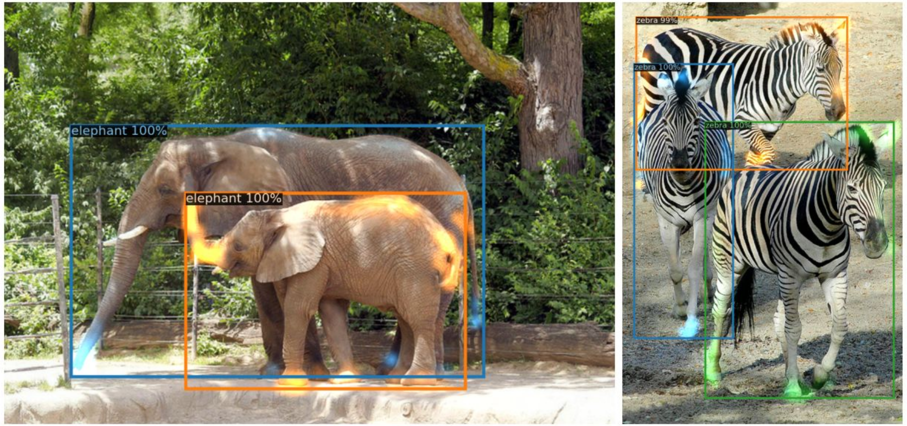


Attention Maps

- Encoder attention separates objects

- Decoder attention looks on objects extremities such as heads or legs





Spatial Encoding

Interesting results - puting spatial encoding only in decoder gives good results,

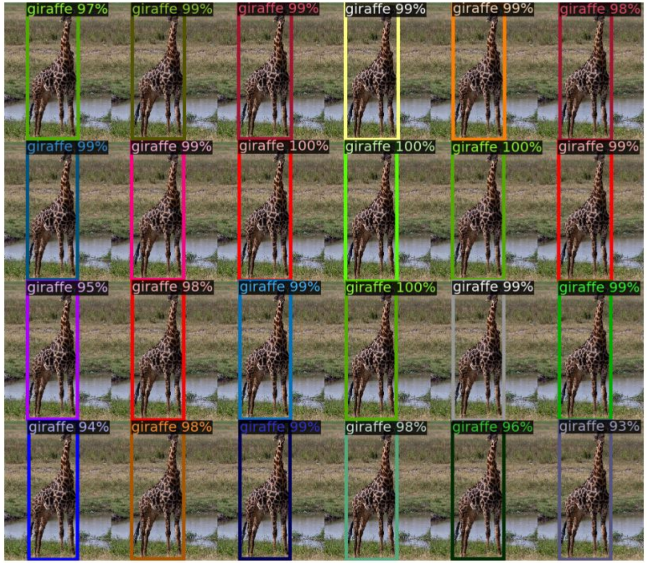
What does it say on attention at encoder?

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**	-

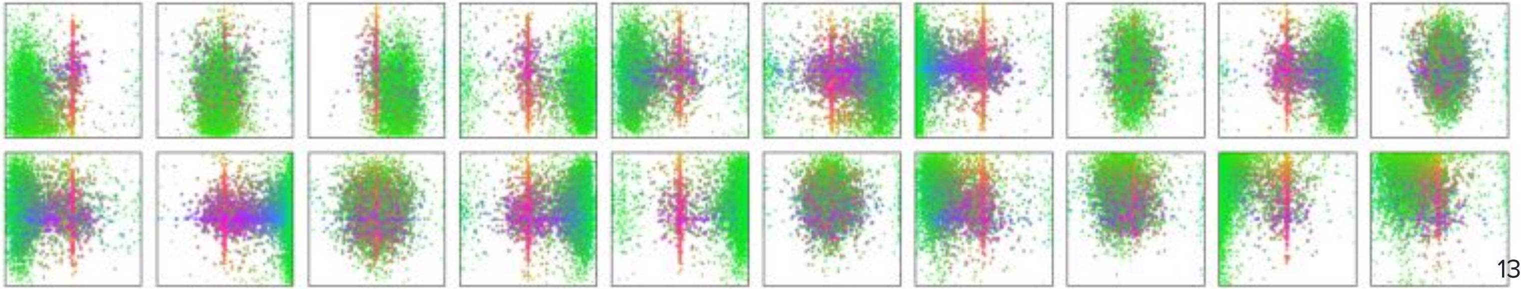
OOD & Object Queries

- 24 giraffes is OOD

- Object queries look on different places for small objects, but all look on the middle for large horizontal objects



Locations of objects identified by each of the object queries.
Green - Small objects, Red - Large horizontal objects, Blue - Large vertical objects



Weaknesses

1. Slow Convergence > 300-500 Epochs vs ~100 Epochs for Faster R-CNN
    a. In Self-Attention, attention weights go like $1/N_k$ which leads to small gradients
    b. Attention maps tend to be sparse

2. Small Objects

Possible Solution

Deformable DETR: Deformable Transformers for End to End Object Detection - Xizhou Zhu et al (ICLR 2021)

Summary

- Object Detection can be solved as a set prediction problem with transformers

- This method has room for improvement
    - Small Objects
    - Training time

Thank You

Loss Issues

- Bounding Box loss
  - Need to consider different sized boxes
  - Average over IoU and L1 loss

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

- How to deal with class imbalance?
  - Anchor based methods sometime uses Focal Loss or Sub-sampling
  - Here they just factor outputs with no class

- Auxiliary decoding loss
  - Also used in Hourglass based methods