End-to-End Object Detection with Transformers

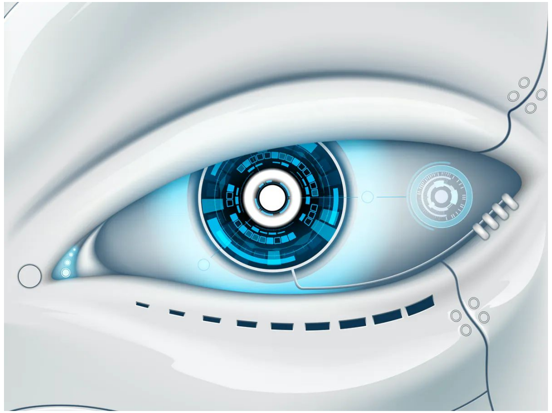
Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko

Presenter: Jay Liao

09/14/2021

Main Problem

- Robot vision use object detection to get object information in the environment
- Majority of the object detection models today use hand-designed components
  - Encodes prior knowledge about the object detection
- Prior end-to-end object detection works
  - Used other forms of prior knowledge
  - Used autoregressive decoding
  - Were not as competitive in results

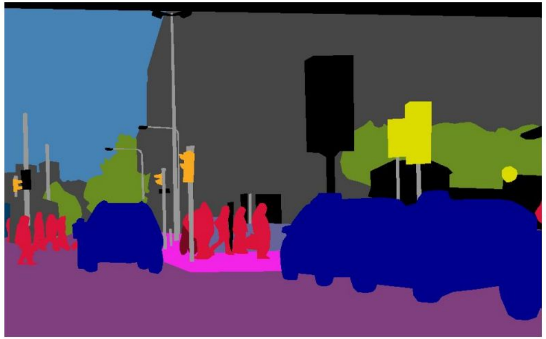


Motivation

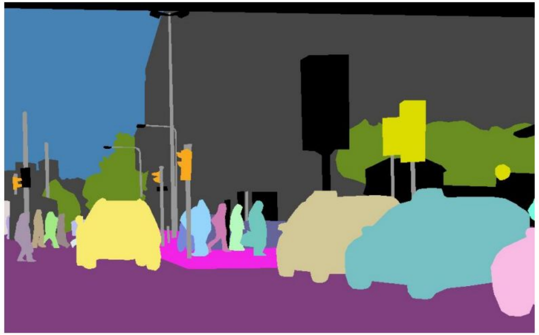
- DETR (DEtection TRansformer)
  - Try out transformer architecture for object detection
  - Transformer can predict multiple objects in parallel
- Bipartite Matching
  - Unique matching
    - Invariant to permutations of predicted objects
    - No more autoregressive decoding to avoid duplicates
  - Bypass the need for NMS or anchors

Problem Setting

- **Object Detection**: for each object in the image:
  - Identify the bounding box of the object in the image
  - Classify the object
- **Panoptic Segmentation**: Given a set of L semantic classes encoded by S := {0, …, L - 1}, for each pixel i of an image:
  - Identify l_i of the pixel, where $l_i \in S$ is the semantic class of pixel i
  - Identify z_i of the pixel, where z_i represents the pixel's instance id
    - Groups pixel of the same class into distinct segments



(b) semantic segmentation



(d) panoptic segmentation

Related Work

- Vinyals et al., 2016: Order Matters: Sequence to Sequence for Sets
    - General approach to set prediction but requires autoregressive decoding
- Zhang et al., 2019: Bridging the Gap Between Anchor-based and Anchor-free Detection via Adaptive Training Sample Selection
    - Shows that performance of object detectors using proposals or anchors are limited by the exact way those initial guesses are set
- Ren et al., 2017: End-to-End Instance Segmentation with Recurrent Attention
- Salvador et al., 2017: Recurrent Neural Networks for Semantic Instance Segmentation
    - Both used bipartite-matching loss with encoder-decoder but evaluated on small datasets, and both used autoregressive models (RNNs)

The DETR Model

Set Prediction Loss for Object Detection

- Infer a fixed-size set of N predictions, where N is much larger than the number of objects in an image
- Use Hungarian Algorithm to find a bipartite matching with the lowest matching cost:

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

- Matching cost accounts for class probability and the predicted box:

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

Set Prediction Loss for Object Detection

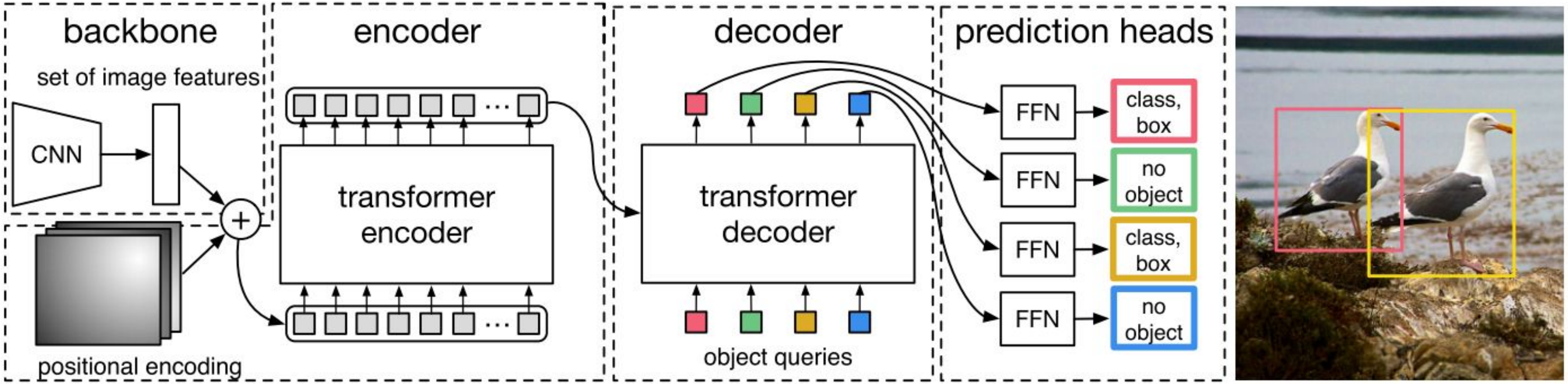
- Loss function: A Hungarian loss for all pairs matched in the previous step using NLL and bounding box loss:

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

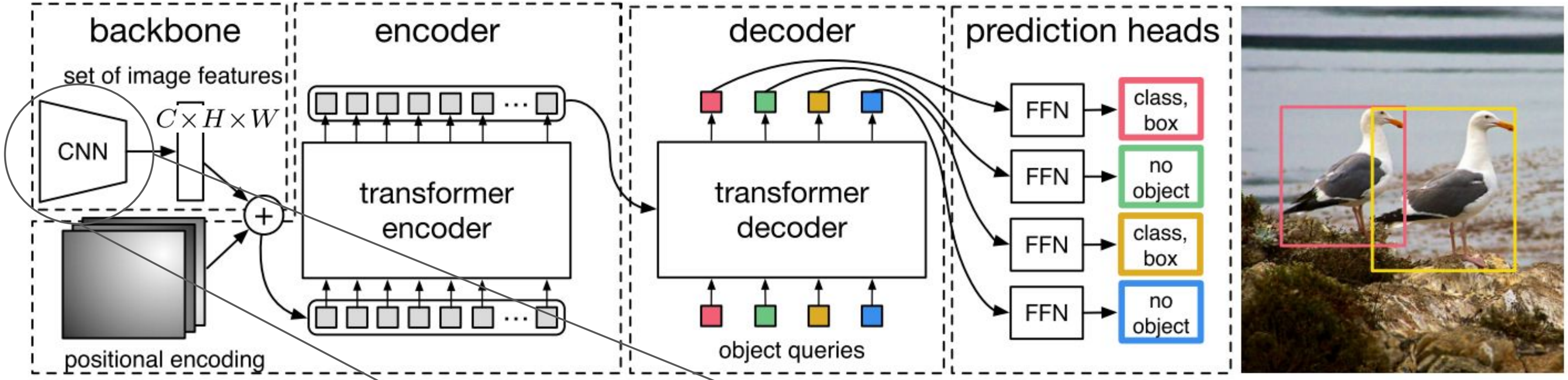
- Bounding box loss use a combination of the L1 loss and the generalized IoU loss that is scale-invariant:

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

DETR Architecture



DETR Architecture



❖ CNN Backbone:

$$f \in \mathbb{R}^{C \times H \times W}$$

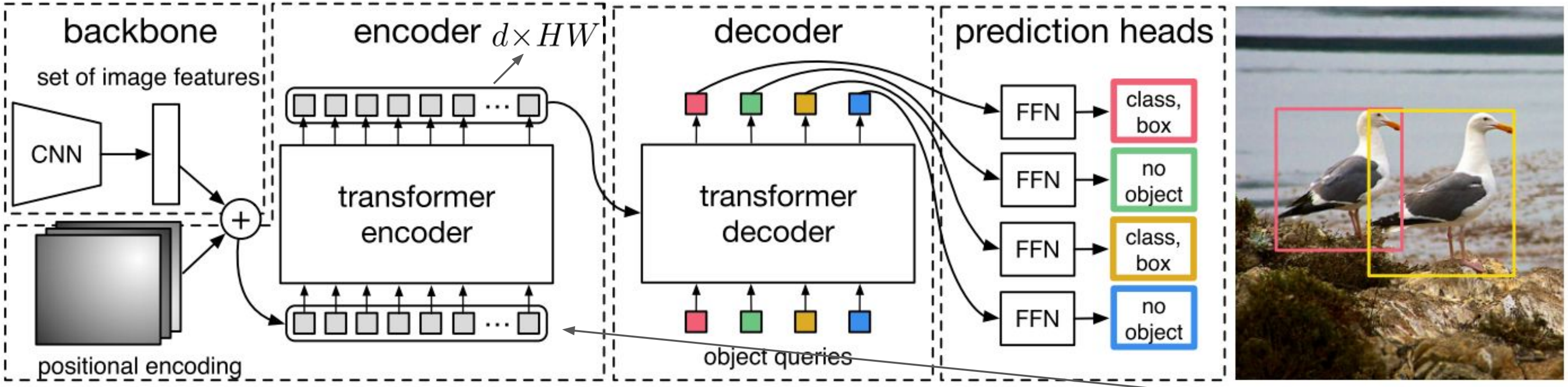
Activation Map $C \times H \times W$

$$x_{\text{img}} \in \mathbb{R}^{3 \times H_0 \times W_0}$$

Typically:

$$C = 2048 \;\; H, W = \frac{H_0}{32}, \frac{W_0}{32}$$

DETR Architecture



❖ Transformer Encoder:

- Reduce channel dimension (1x1 Convolution)
- Flatten features into a sequential feature map
- Add positional encodings to input of each attention layer
- A multi-head self-attention module and a feed forward network

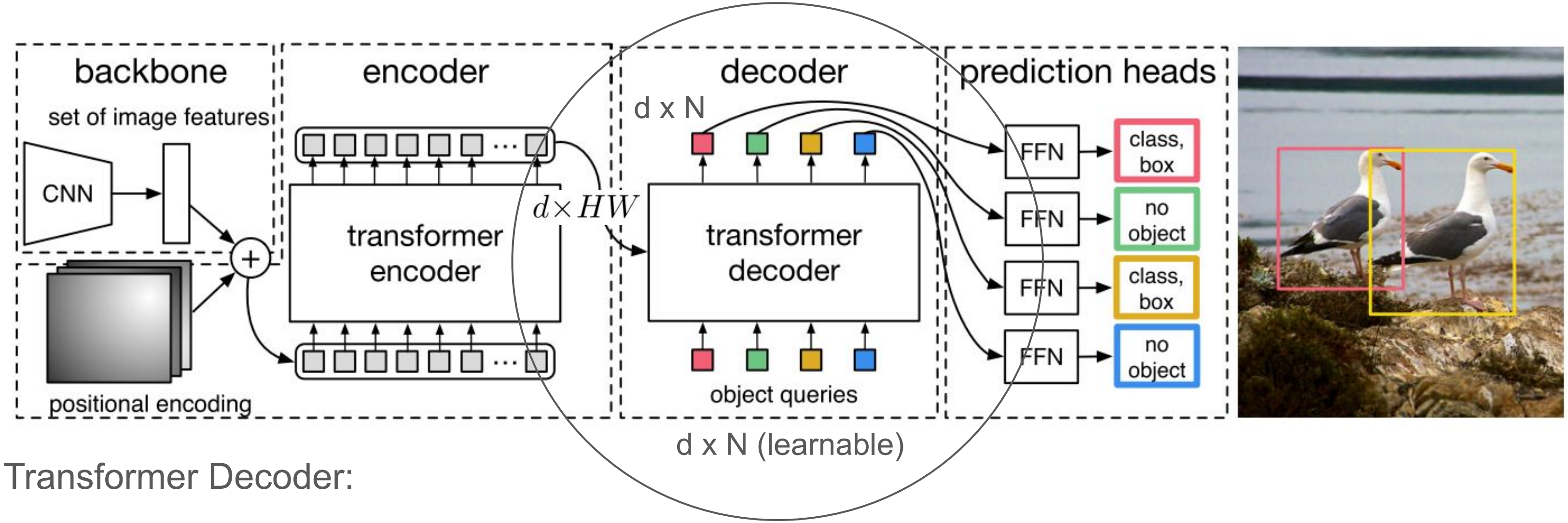
$$C\times H\times W \longrightarrow z_0 \in \mathbb{R}^{d\times H\times W}$$

$$d\times HW$$

+

Positional Encoding

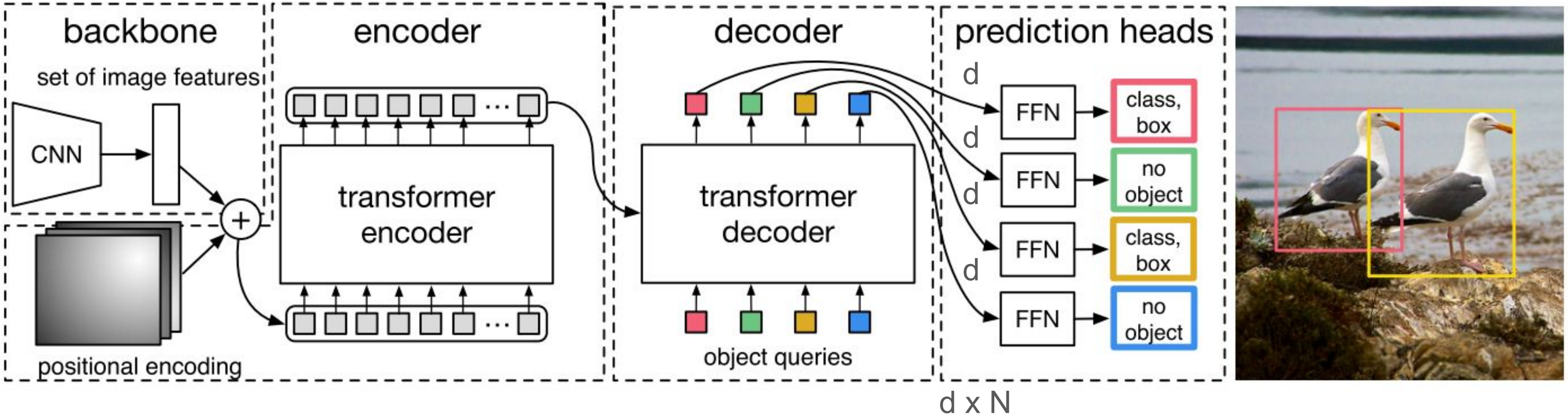
DETR Architecture



❖ Transformer Decoder:

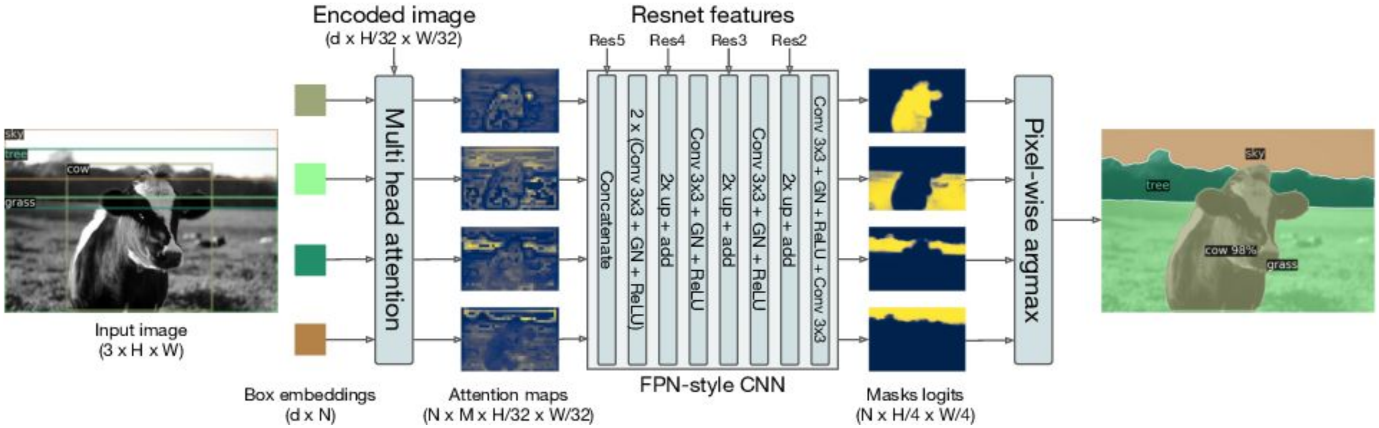
- Use encoded representations, d x HW embeddings, from encoder as key and value
- Add N learnt positional encodings (object queries) to input of each attention layer
- Transforms N object queries into N output embeddings in parallel (non-autoregressive)
- Trained with auxiliary decoding loss to improve training

DETR Architecture



- Prediction Feed-forward networks (FFNs):
  - For normalized center coordinates: 3-layer perceptron with ReLU activation, hidden dim d
  - For class prediction: a linear projection layer with softmax
  - Class prediction also predicts “no object”

Panoptic Segmentation Head

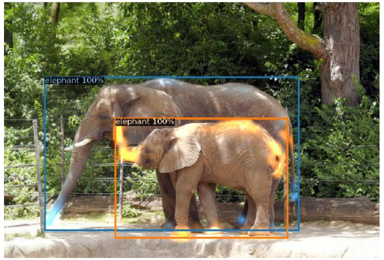


- Predicts a binary mask of the predicted bounding boxes
  - Compute multi-head attention heatmap of decoder output over encoder output
  - Use an FPN-like architecture to increase the resolution of the mask
  - Mask is supervised independently using DICE/F-1 loss and Focal loss

Experimental Setup

Tasks and Datasets

- Object Detection and Panoptic Segmentation
- COCO 2017 detection and panoptic segmentation datasets
    - 118k training images and 5k validation images
    - Each image is annotated with bounding boxes and panoptic segmentation
    - Average 7 instances per image; up to 63 instances in a single image in training dataset
    - Panoptic annotations of 53 stuff categories and 80 things categories



Detection Baselines

- Faster R-CNN
  - Features explored: Dilated C5 (DC5), Feature Pyramid Network (FPN), and ResNet-101 backbone with FPN (R101-FPN)
  - Stronger Faster R-CNN baselines:
    - Longer training (like for transformers)
    - Same random crop augmentation
    - Add generalized IoU to the box loss
- Can DETR perform comparably to ResNet under similar settings?

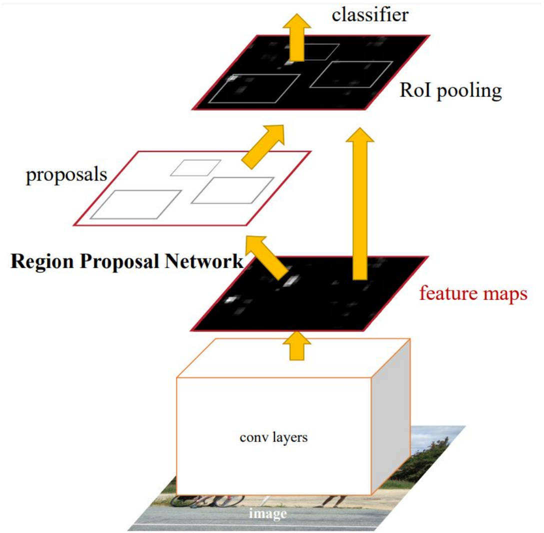


Figure 2 of [Ren et al., 2016]

Detection Metrics

- AP (Average Precision)
  - Precision: True Positives / (True Positives + False Positives)
    - Correct predictions out of all predictions
  - Recall: True Positives / (True Positives + False Negatives)
    - Correct predictions out of all objects in ground truth
  - Average Precision: area beneath the precision-recall curve

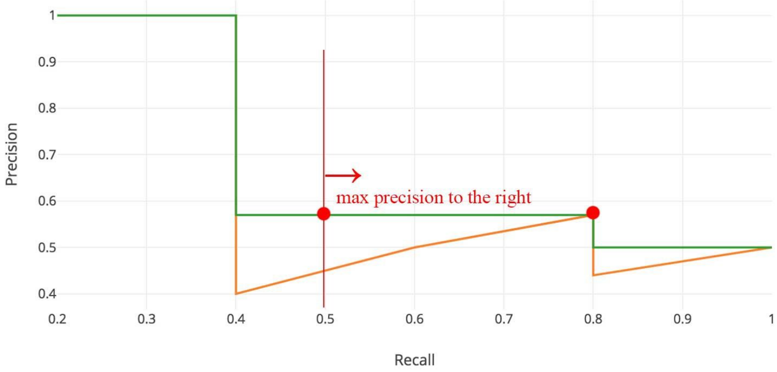


Image Source

Detection Metrics

- AP (Average Precision)
  - Intersection-over-Union (IoU): Area of Overlap / Area of Union
    - Measures how much bounding box prediction intersects with ground truth
  - AP: Average AP at IoU = 50%, 5%, and 95%.
  - AP_{50}: Only bounding box with IoU = 50% is counted as true positive
    - Similarly for AP_{75}
  - AP_S, AP_M, AP_L: AP based on objects of different sizes
    - Refer to: https://cocodataset.org/#detection-eval

Panoptic Segmentation

- Baselines
  - UPSNet
  - Panoptic FPN
    - Same data augmentation as DETR
    - Longer training schedule

Panoptic Segmentation Metrics

- Mask AP for things classes
- Panoptic Quality
  - PQ^{th}: PQ for things
  - PQ^{st}: PQ for stuff

$$\mathrm{PQ} = \frac{\sum_{(p,g)\in TP} \mathrm{IoU}(p,g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}$$

$$\mathrm{PQ} = \underbrace{\frac{\sum_{(p,g)\in TP} \mathrm{IoU}(p,g)}{|TP|}}_{\text{segmentation quality (SQ)}} \times \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{recognition quality (RQ)}}$$

Experimental Results

Detection Results

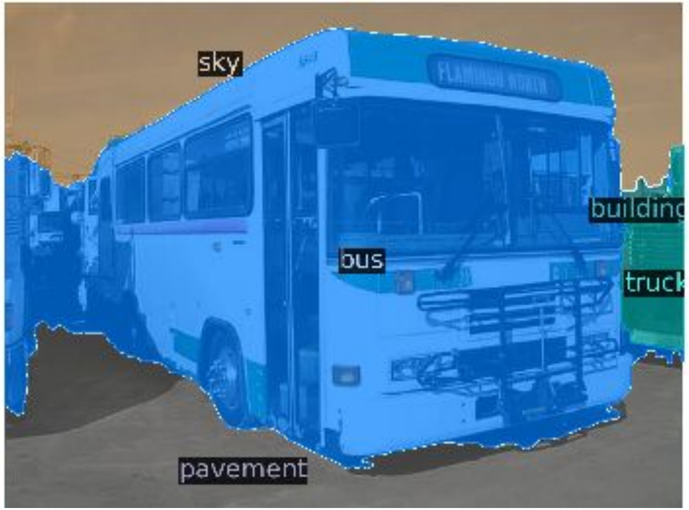
Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-DC5+	320/16	166M	41.1	61.4	44.3	22.9	45.9	55.0
Faster 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**

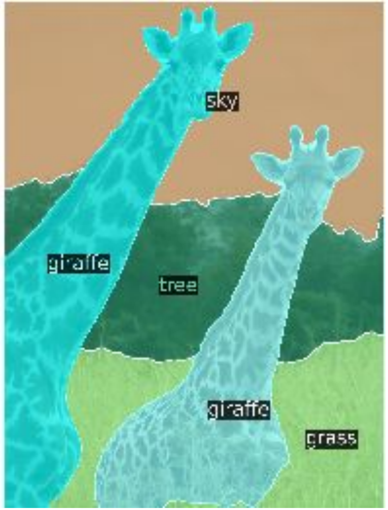
Panoptic Segmentation Results

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

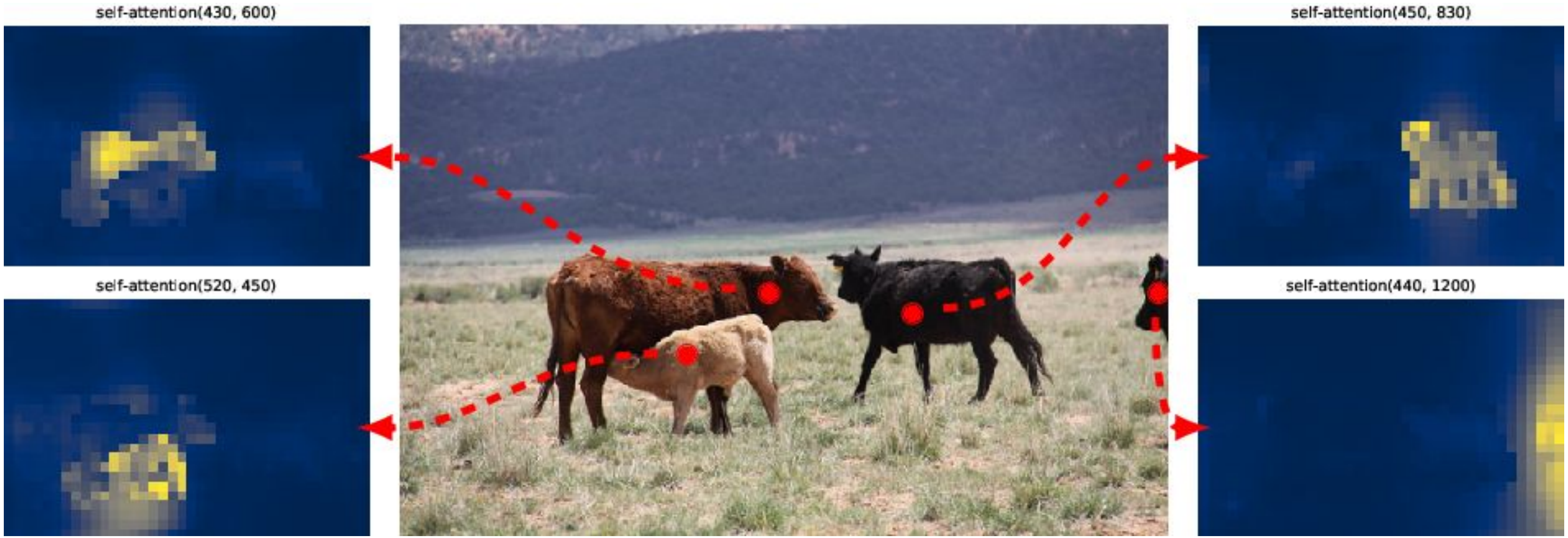
Panoptic Segmentation Example





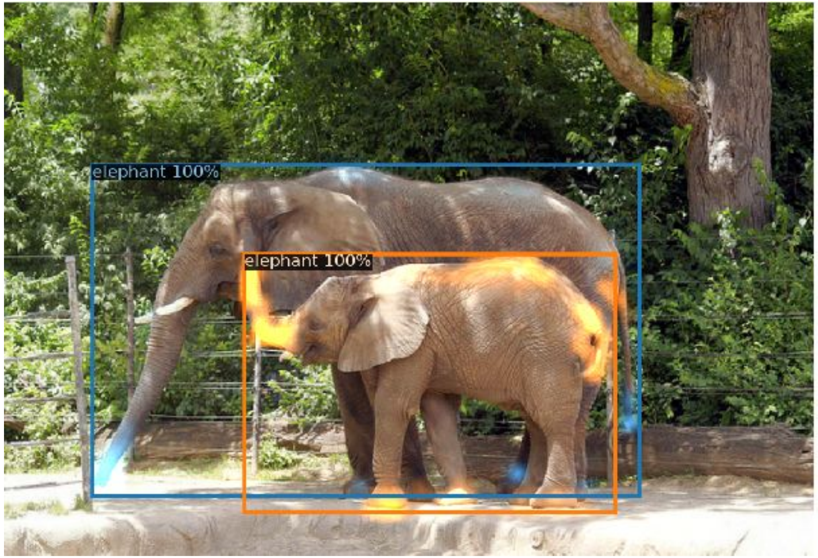
Ablations

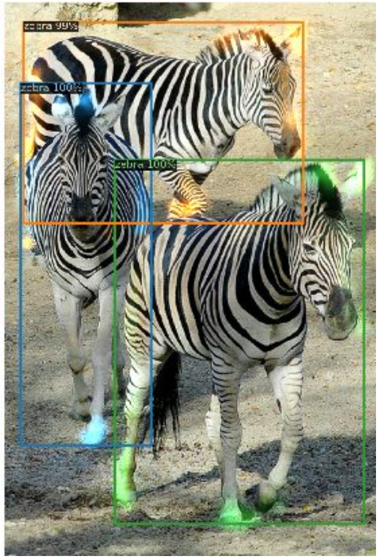
- More encoder layers improve AP overall
    - Without encoder layers, AP drops by 3.9 with a significant drop of 6.0 AP in large objects



Ablations

- AP and AP_{50} improve after every decoder layer trained with auxiliary loss, totalling +8.2/9.5 in AP
- Latter layers of decoder inhibits duplicate predictions





Ablations

- Removing FFN decreased AP by 2.3
- Removing positional encodings decreased AP by 7.8
- Using just L1 without the generalized IoU loss decreased AP by 4.8

Critique and Summary

Critique

- DETR requires a long training time
  - Self-attention is has a quadratic complexity of $O(n^2d)$
  - Baseline model took 3 days to train on 16 GPUs (4 image per GPU) for 300 epochs
- Faster R-CNN outperforms DETR in AP_S for object detection
  - AP increase by 1.4 comes at the expense of more GFLOPS and half (10 FPS) the FPS of the best Faster R-CNN model (20 FPS)

Extended Readings

- Han et al., 2021: A Survey on Vision Transformers
    - Addresses that DETR has a slow convergence and other limitations of DETR.
    - Proposed several papers that improved DETR’s training time and AP.
- Zhu et al., 2021: Deformable DETR: Deformable Transformers for End-to-End Object Detection
    - Use a deformed attention module instead of self-attention, which attends to a small sample of feature maps instead of all, and this improves both time complexity and AP.
- Chen et al., 2021: Points as Queries: Weakly Semi-supervised Object Detection by Points
    - Encode object centers (points) as object queries to DETR instead of learnt positional encodings. This is done by using a point encoder on predicted points on an image.
- Wang et al., 2021: Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions
    - A backbone that uses a transformer to generate feature pyramids, and the features are compatible with DETR. (Pure Transformers!)

Summary

- DETR: End-to-end object detection using transformers by modeling object detection as a set prediction problem
- Need to remove prediction duplicates without using hand-designed components
  - These components encoded prior knowledge about the task and impacted performance
- Training objective need to be invariant to permutations of predictions
  - Prior works used autoregressive decoding, which takes up inference time
- Bipartite matching allowed training objective to be permutation invariant
- Transformers attended to more information and can predict objects in parallel
- DETR models beat comparable Faster R-CNN models in AP and AP_L, but lose in AP_S

Thank You