End-to-End Object Detection with Transformers

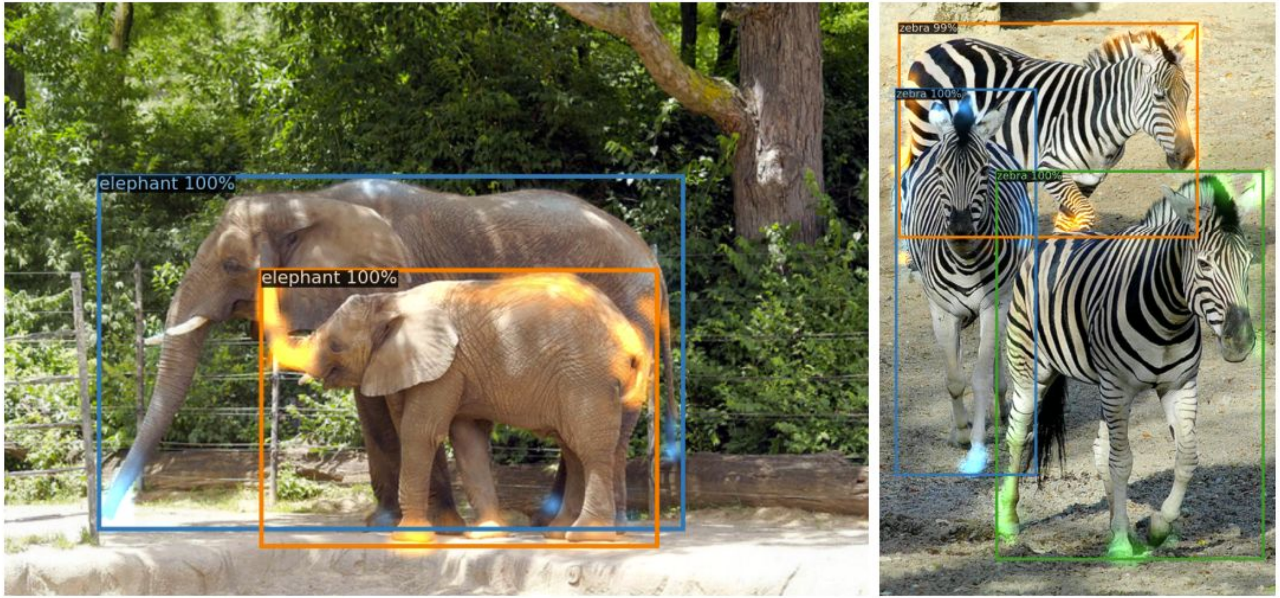
Nicolas Carion*, Francisco Massa*, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko

ECCV 2020

Presented by Xinyu Liu, Liujie Zheng, and Tarik Reza Toha

Object Detection

The goal is to predict a set of bounding boxes and category labels for each object of interest.



Prior Detectors

Modern detectors address this set prediction task in an **indirect** way, by defining **surrogate regression and classification problems** on a large set of proposals, anchors, or window centers.

Two-stage detectors predict boxes w.r.t. proposals, whereas single-stage methods make predictions w.r.t. anchors or a grid of possible object centers.

Prior Detectors

Limitation: Their performances are significantly influenced by postprocessing steps (non-maximum suppression) to collapse near-duplicate predictions, by the design of the anchor sets and by the heuristics that assign target boxes to anchors.

Is it possible to bypass the surrogate tasks and simplify the pipelines?

Prior Detectors

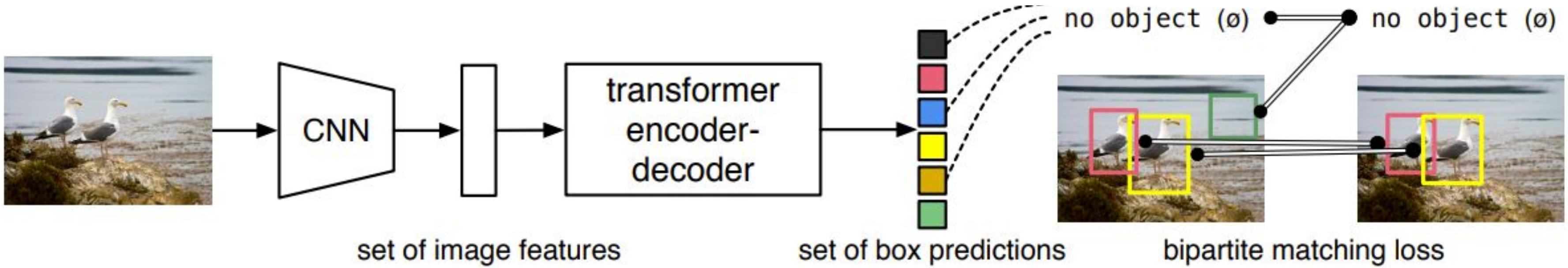
Limitation: Their performances are significantly influenced by postprocessing steps (non-maximum suppression) to collapse near-duplicate predictions, by the design of the anchor sets and by the heuristics that assign target boxes to anchors.

Is it possible to bypass the surrogate tasks and simplify the pipelines?

The authors streamline the training pipeline by viewing object detection as **a direct set prediction problem**, predicting the set of detections with absolute box prediction w.r.t. the input image rather than an anchor.

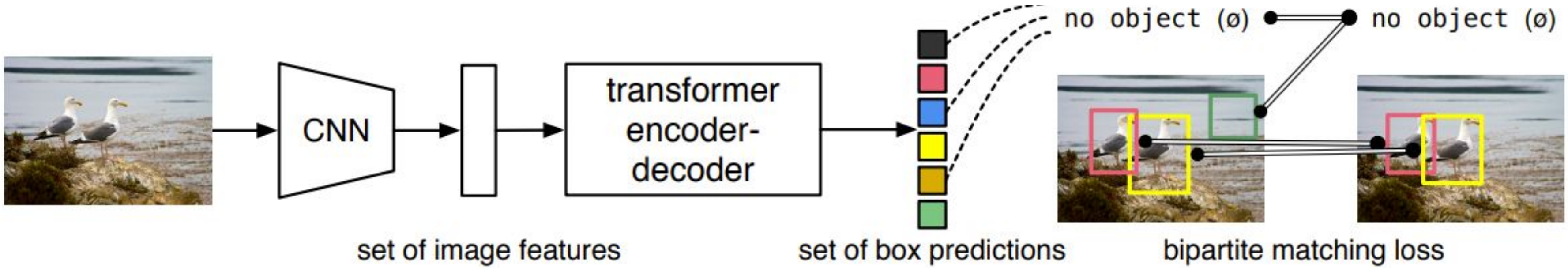
DETR

The DEtection TRansformer **predicts all objects at once**, and is trained **end-to-end** with a set loss function which performs **bipartite matching** between predicted and ground-truth objects.



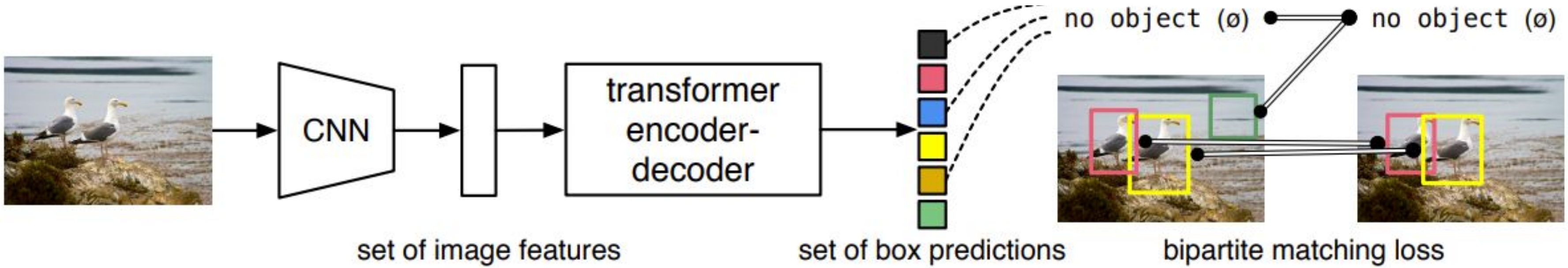
DETR

DETR streamlines the detection pipeline, effectively **removing the need for many hand-designed components**, like a non-maximum suppression procedure or anchor generation that explicitly encode prior knowledge about the task.

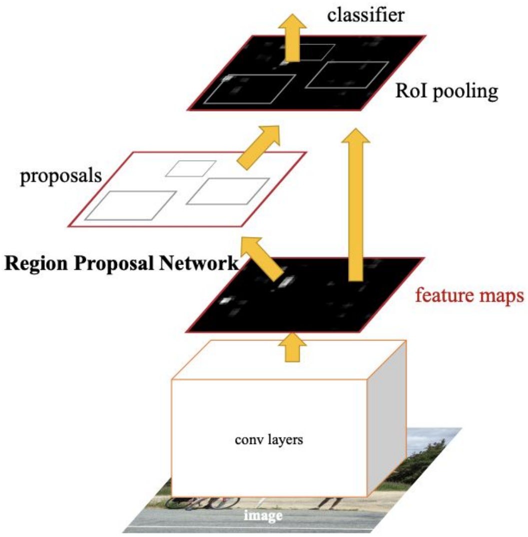


DETR

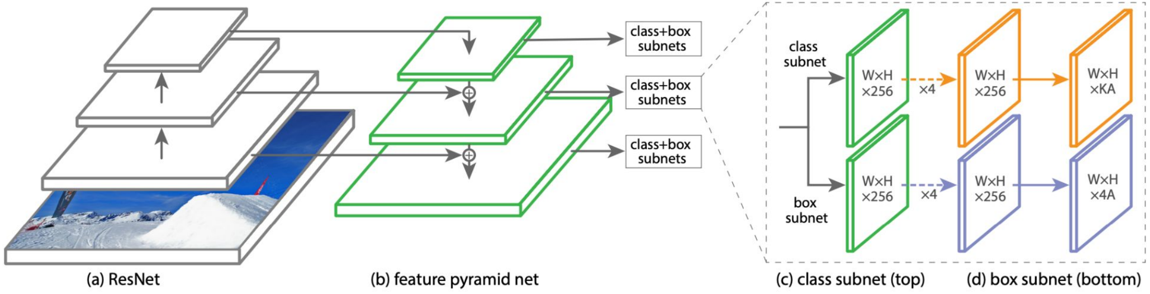
Unlike most existing detection methods, DETR doesn't require any customized layers, and thus **can be reproduced easily** in any framework that contains standard CNN and transformer classes.



Previous works



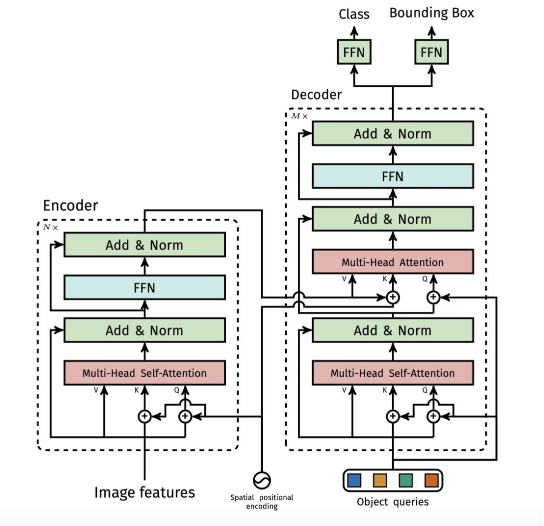
Faster R-CNN
(proposal-based)



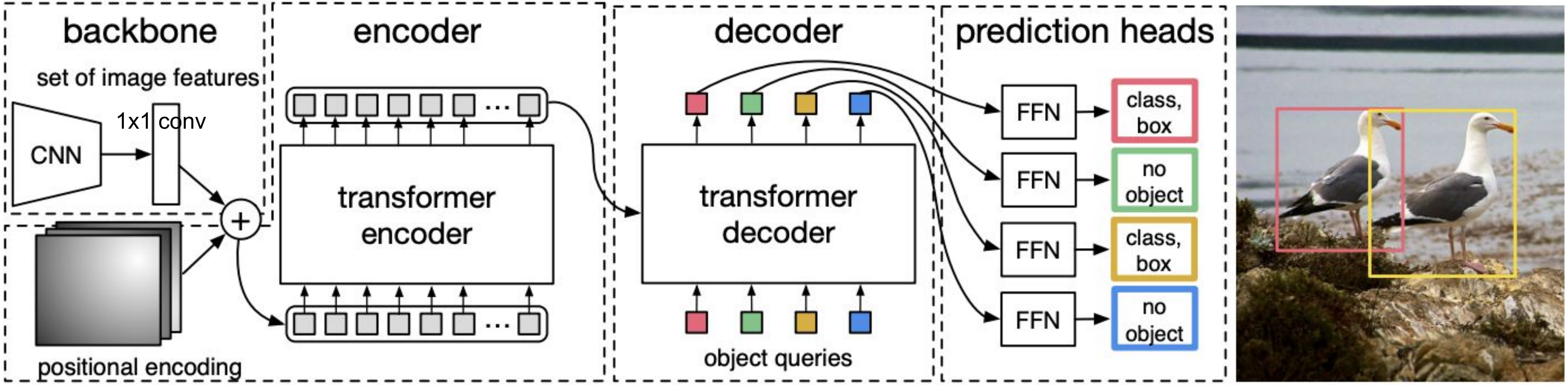
RetinaNet (anchor-based)

Drawbacks:
* require customized layers
* require post-processing

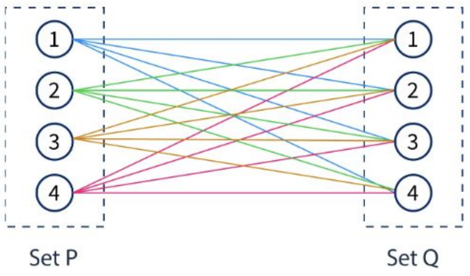
The DETR model



bipartite matching loss

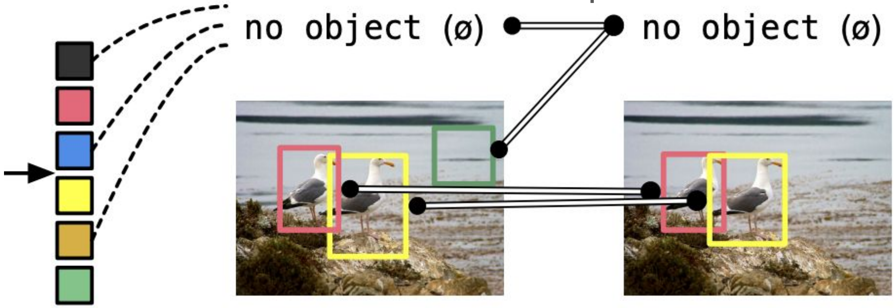


The set prediction loss



N predictions

M < N ground truths padded to N



bipartite matching loss

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

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

probability of class c_i

loss of bbox

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

IoU loss

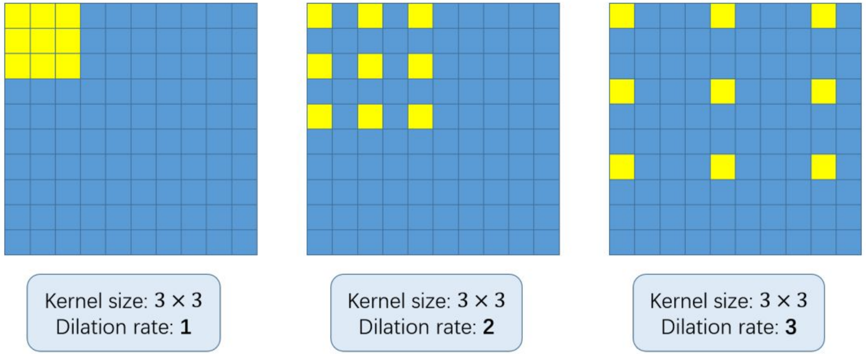
L1 loss

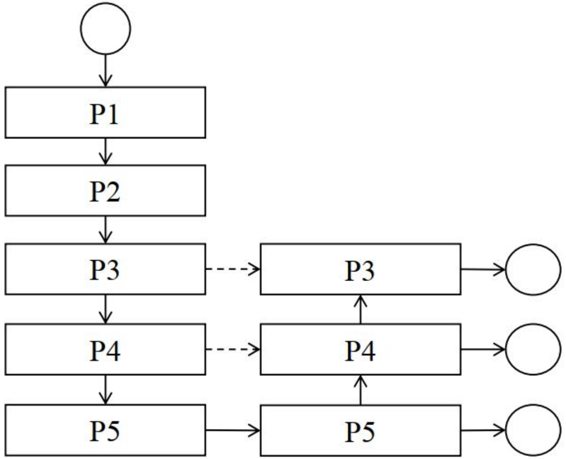
This optimal assignment is computed efficiently with the Hungarian algorithm

Experimental Results

Baseline Architectures

- Backbone network
  - ResNet-50 (pre-trained from ImageNet)
  - ResNet-101 (pre-trained from ImageNet)
- Object Detectors
  - Faster R-CNN (Proposal-based)
  - RetinaNet (Anchor-based)
- Additional Modules
  - Dilated convolution to expand the receptive field without losing resolution
  - Feature pyramid network for small object detection





Evaluation Results on COCO Dataset

Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
RetinaNet	205/18	38M	38.7	58.0	41.5	23.3	42.3	50.3
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
RetinaNet+	205/18	38M	41.1	60.4	43.7	25.6	44.8	53.6
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**

‘+’ means longer training, GIoU loss, and crop augmentation

Evaluation Results on COCO Dataset (contd.)

Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
RetinaNet	205/18	38M	38.7	58.0	41.5	23.3	42.3	50.3
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
RetinaNet+	205/18	38M	41.1	60.4	43.7	25.6	44.8	53.6
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**

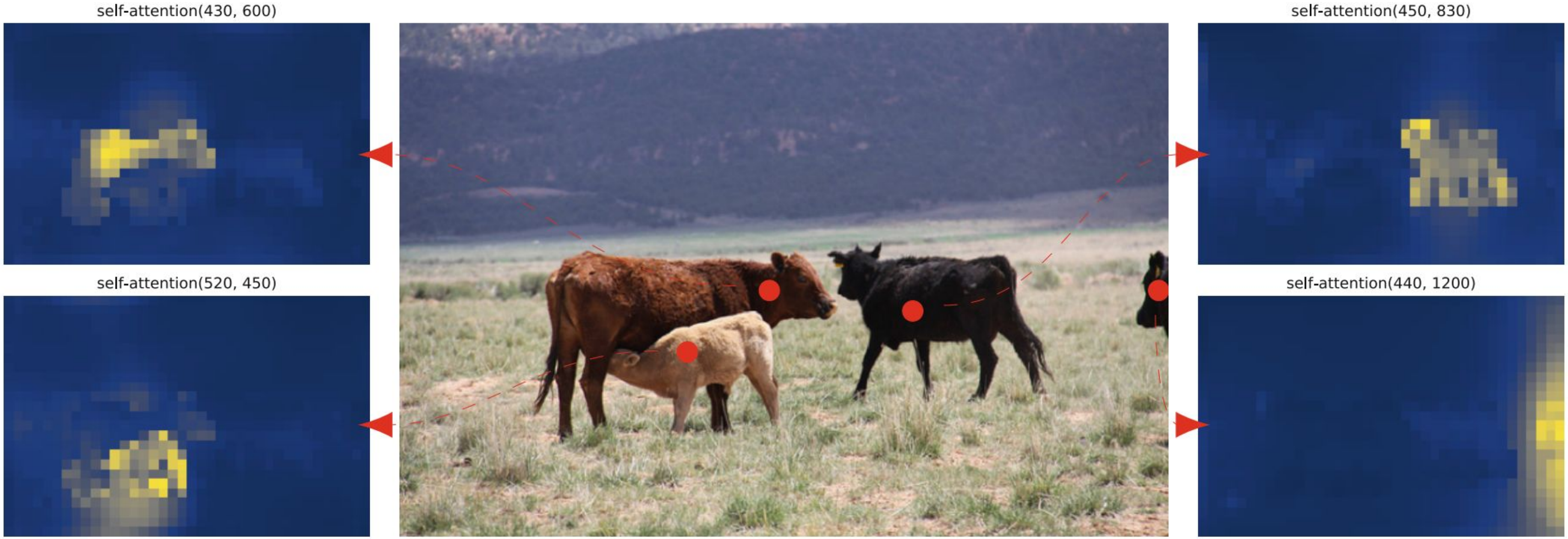
‘+’ means longer training, GIoU loss, and crop augmentation

Evaluation Results on COCO Dataset (contd.)

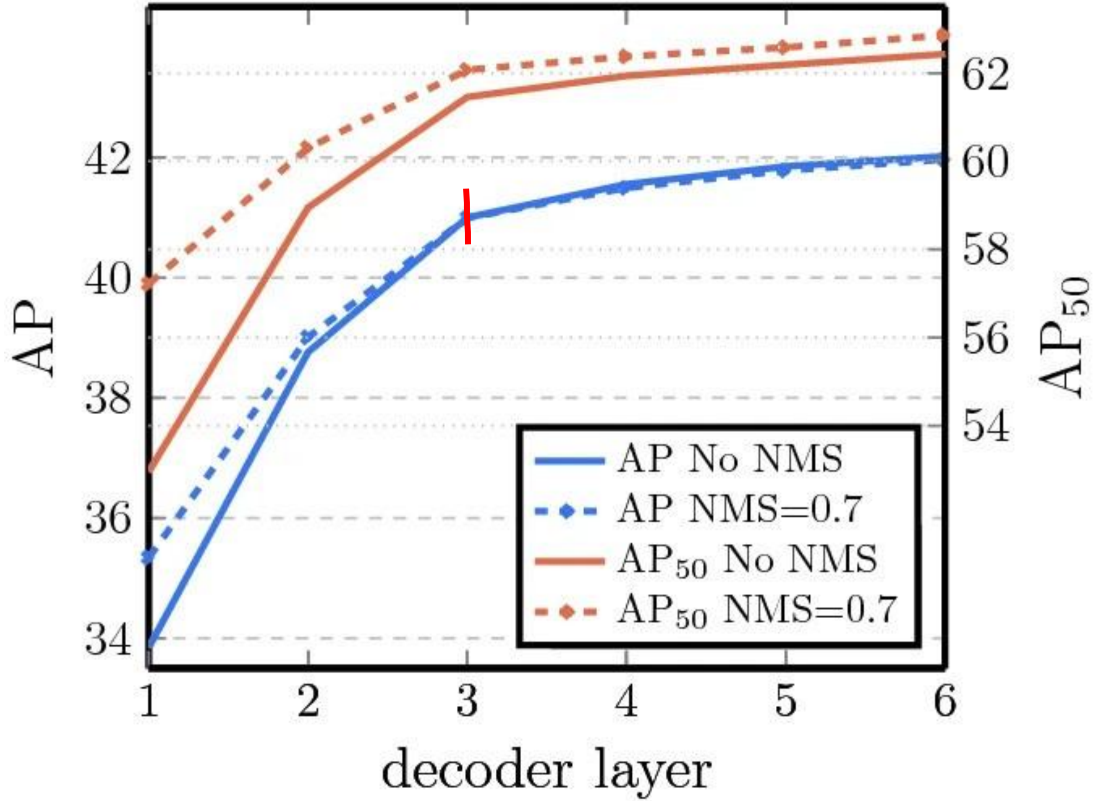
Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
RetinaNet	205/18	38M	38.7	58.0	41.5	23.3	42.3	50.3
Faster RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
RetinaNet+	205/18	38M	41.1	60.4	43.7	25.6	44.8	53.6
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**

‘+’ means longer training, GIoU loss, and crop augmentation

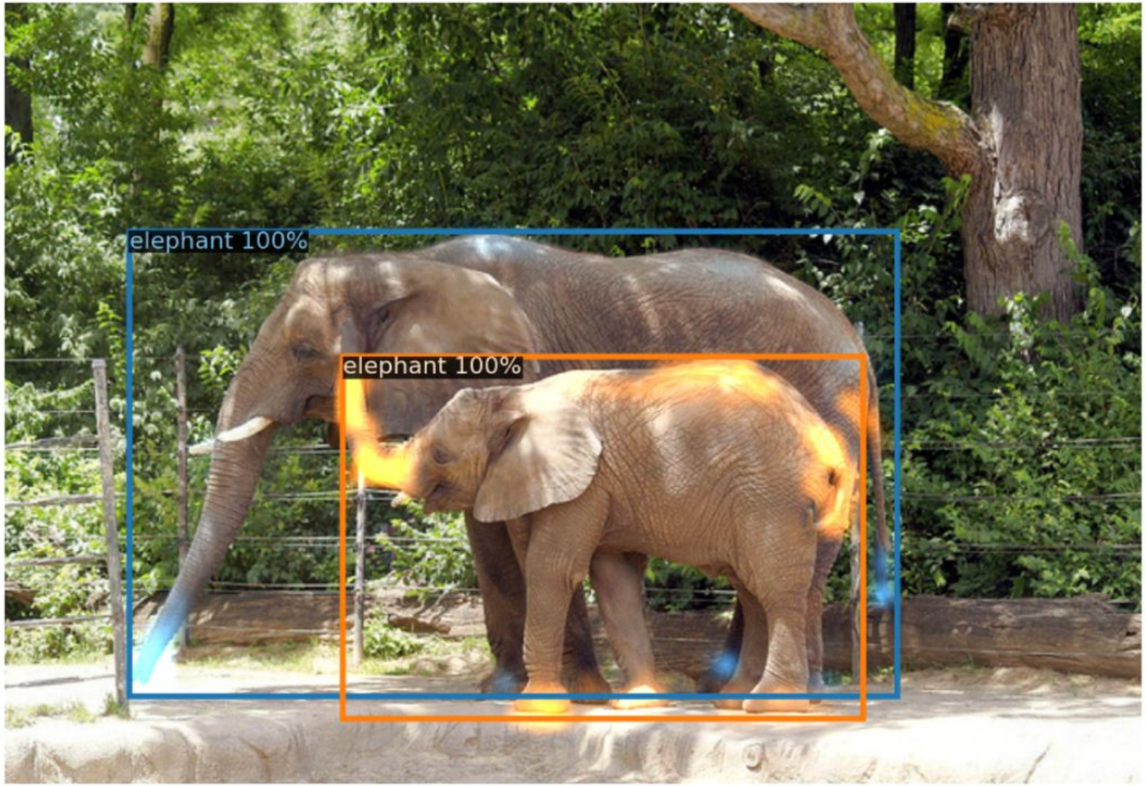
Efficacy of Encoder Module

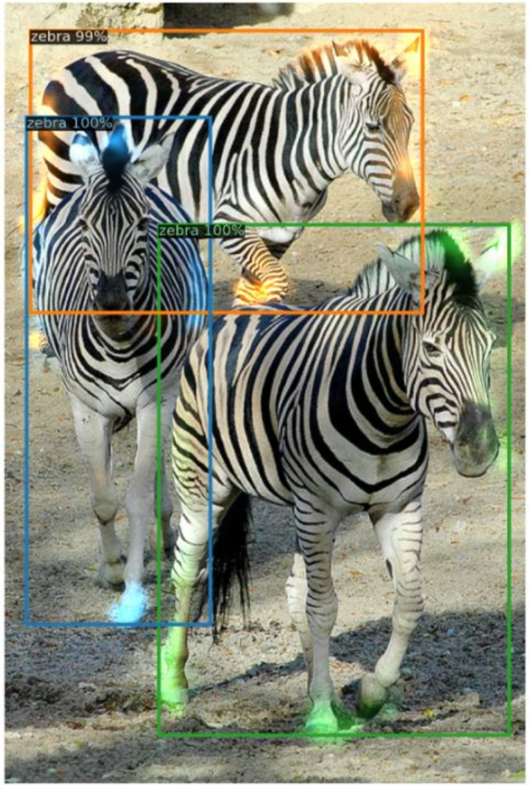


Efficacy of Decoder Module

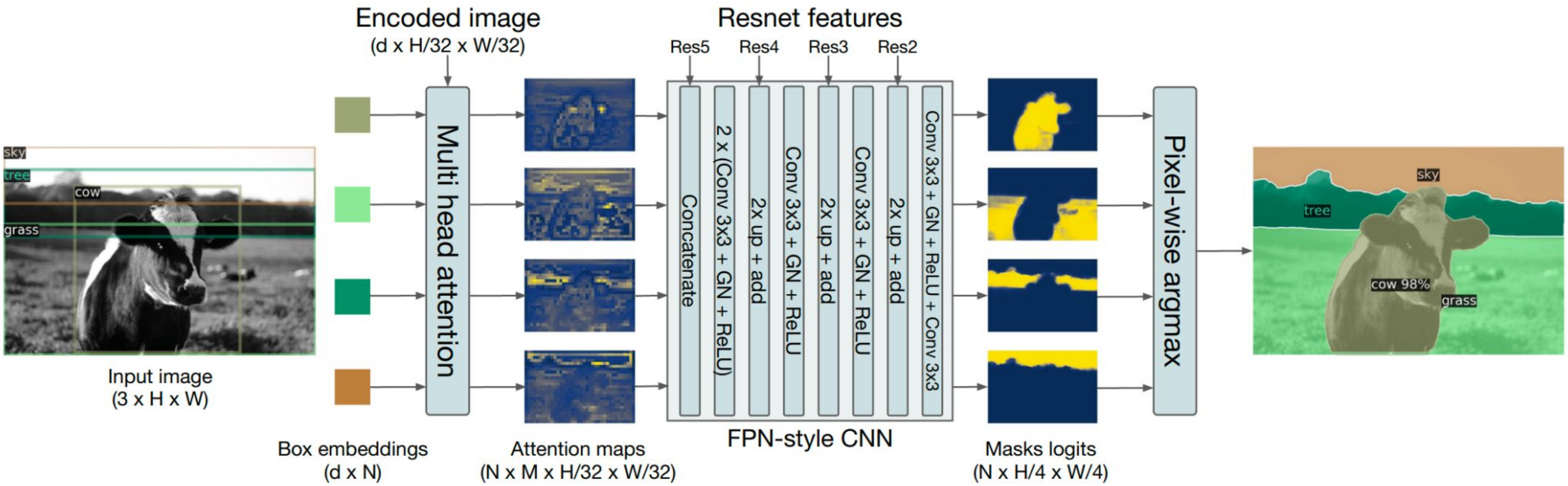


Visualizing the Decoder Attention





Extension: Panoptic Segmentation



Panoptic Results on COCO Dataset

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	45.1	79.9	55.5	50.5	80.9	61.7	37.0	78.5	46.0	33.0
DETR-DC5	R101	**45.6**	**80.0**	**56.1**	50.9	80.9	**62.2**	**37.5**	78.6	**46.8**	33.1

Panoptic Visualization

Conclusions

- DETR incorporates the transformer and bipartite matching loss for object detection task
  - It achieves comparable results to an optimized Faster R-CNN baseline on the challenging COCO dataset
  - It is easily extensible to panoptic segmentation with competitive results
- Although DETR performs significantly better on large objects, it cannot deliver similar improvement on small objects
  - It is left as a future work