

Object Detection II

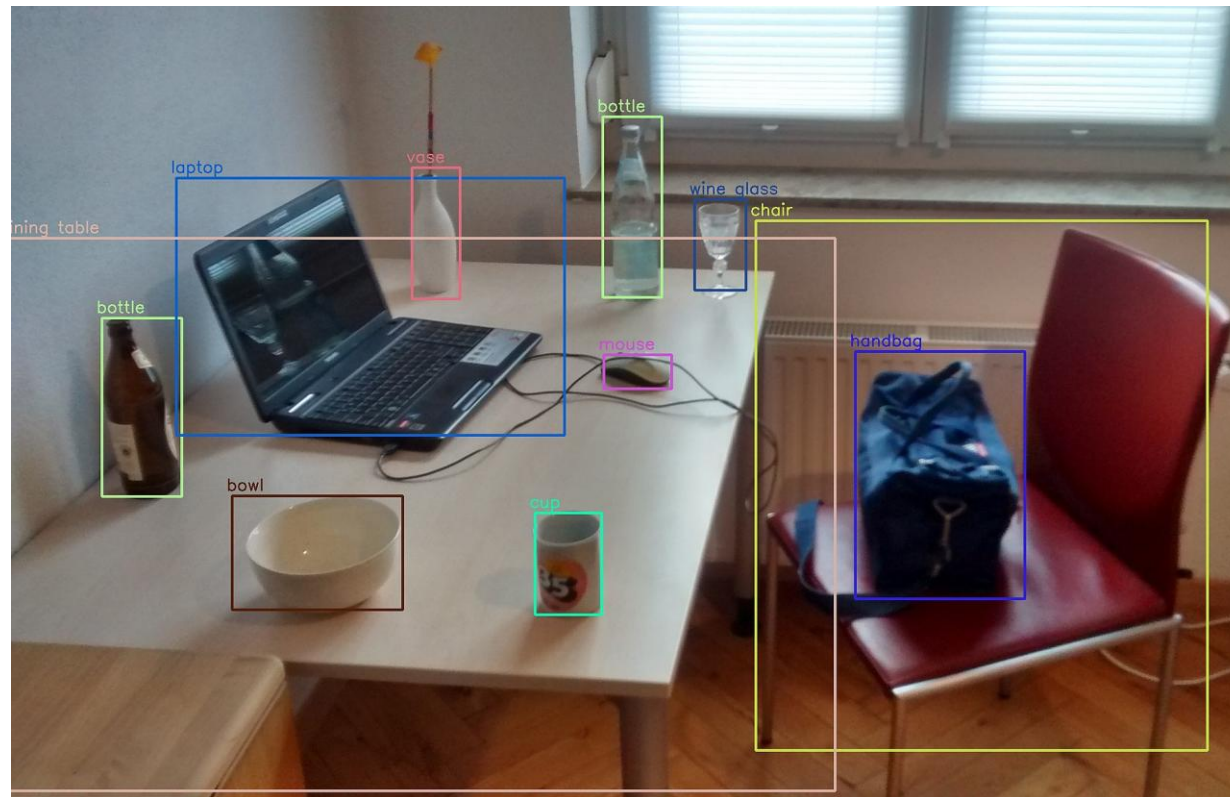
CS 4391 Introduction Computer Vision

Professor Yu Xiang

The University of Texas at Dallas

Object Detection

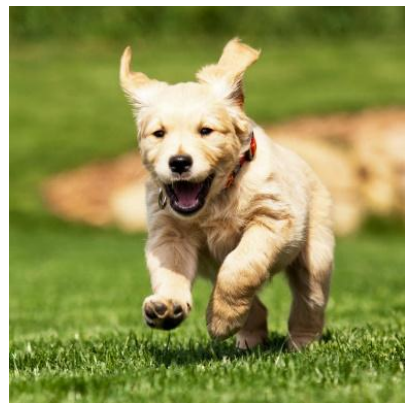
- Localize objects in images and classify them



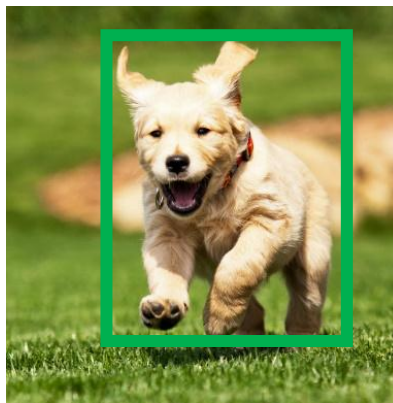
Wikipedia

Object Detection

- Localization + Classification

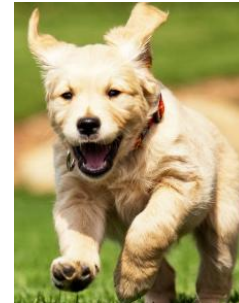


Input Image



Localization

Crop



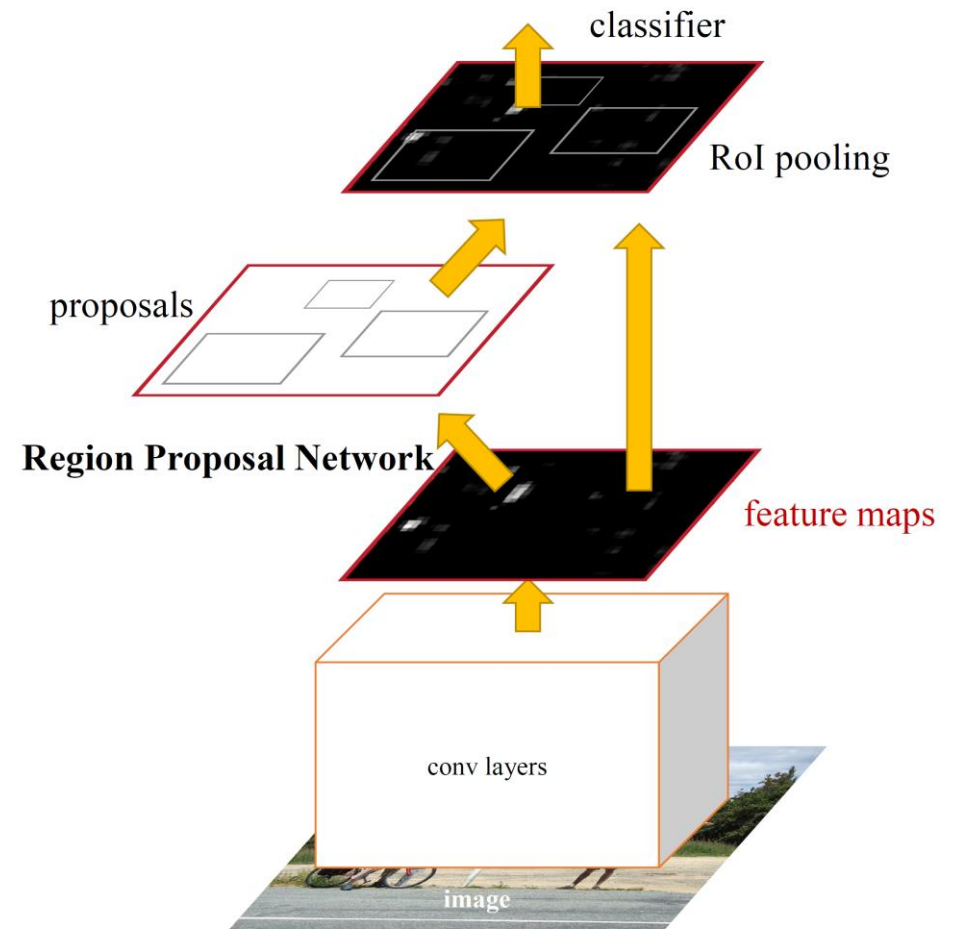
Classifier



Dog

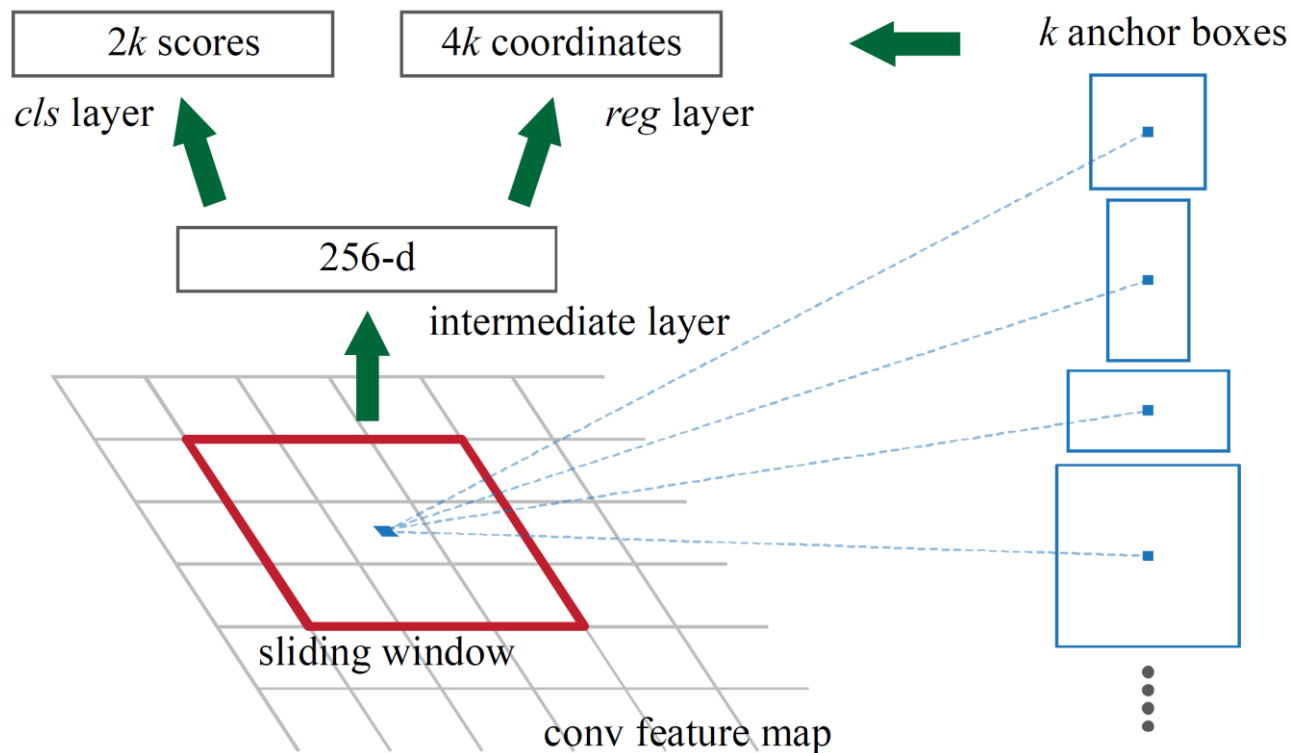
Faster R-CNN

- A single network for object detection
 - Region proposal network
 - Classification network



Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Ren et al., NeurIPS, 2015

Region Proposal Network



An anchor is centered at the sliding window in question, and is associated with a scale and aspect ratio (3 scales, 3 aspect ratios, $k=9$)

3x3 conv layer to 256-d

```
layer {
  name: "rpn_conv/3x3"
  type: "Convolution"
  bottom: "conv5"
  top: "rpn/output"
  param { lr_mult: 1.0 }
  param { lr_mult: 2.0 }
  convolution_param {
    num_output: 256
    kernel_size: 3 pad: 1 stride: 1
    weight_filler { type: "gaussian" std: 0.01 }
    bias_filler { type: "constant" value: 0 }
  }
}
```

classification

```
layer {
  name: "rpn_cls_score"
  type: "Convolution"
  bottom: "rpn/output"
  top: "rpn_cls_score"
  param { lr_mult: 1.0 }
  param { lr_mult: 2.0 }
  convolution_param {
    num_output: 18 # 2(bg/fg) * 9(anchors)
    kernel_size: 1 pad: 0 stride: 1
    weight_filler { type: "gaussian" std: 0.01 }
    bias_filler { type: "constant" value: 0 }
  }
}
```

Bounding box regression

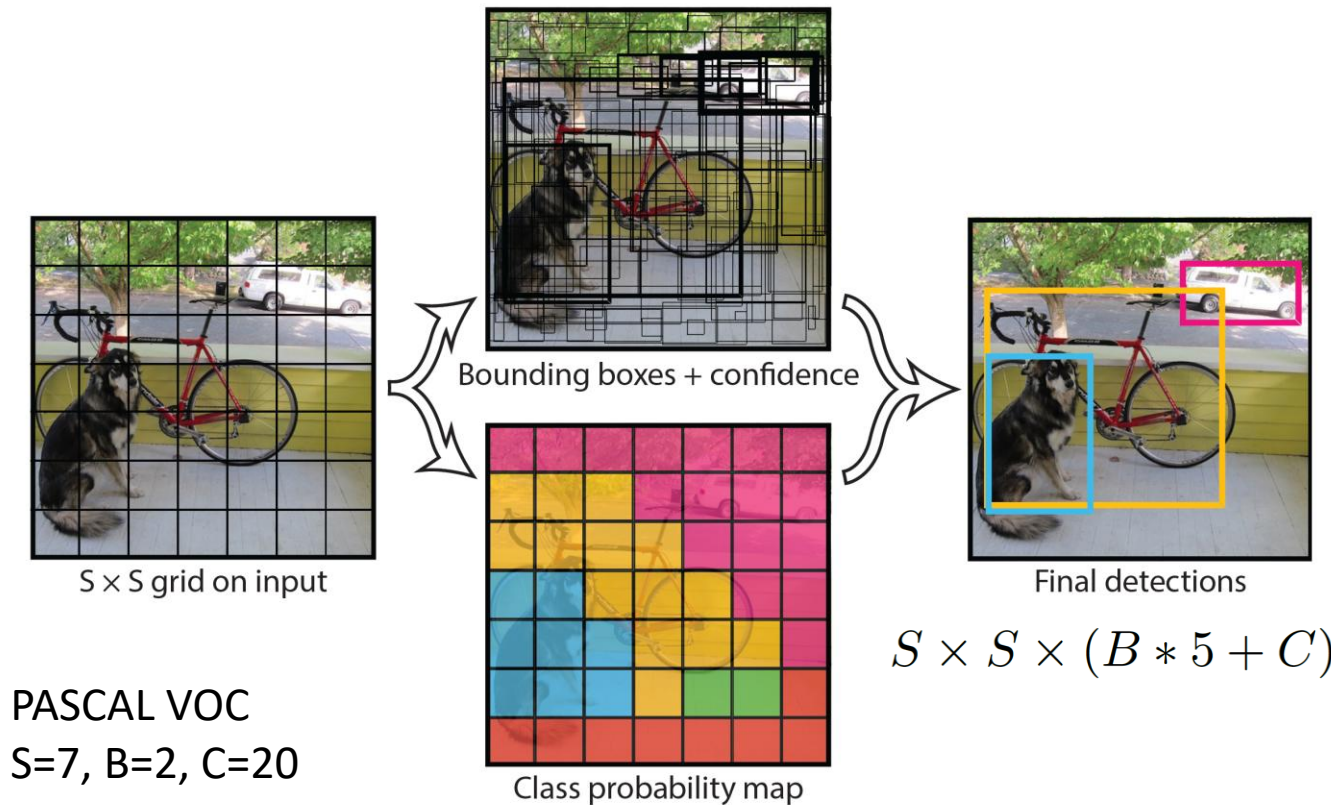
```
layer {
  name: "rpn_bbox_pred"
  type: "Convolution"
  bottom: "rpn/output"
  top: "rpn_bbox_pred"
  param { lr_mult: 1.0 }
  param { lr_mult: 2.0 }
  convolution_param {
    num_output: 36 # 4 * 9(anchors)
    kernel_size: 1 pad: 0 stride: 1
    weight_filler { type: "gaussian" std: 0.01 }
    bias_filler { type: "constant" value: 0 }
  }
}
```

Two stage vs One stage

- Two stage detection methods
 - Stage 1: generate region proposals
 - Stage 2: classify region proposals and refine their locations
 - E.g., R-CNN, Fast R-CNN, Faster R-CNN
- One stage detection methods
 - An end-to-end network for object detection
 - E.g., YOLO

YOLO

- Regress to bounding box locations and class probabilities



- Each grid handles objects with centers (x, y) in it
- Each grid predicts B bounding boxes
- Each bounding box predicts (x, y, w, h) and confidence (IoU of box and ground truth box)

$$\Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

- Each grid also predicts C class probabilities

$$\Pr(\text{Class}_i | \text{Object})$$

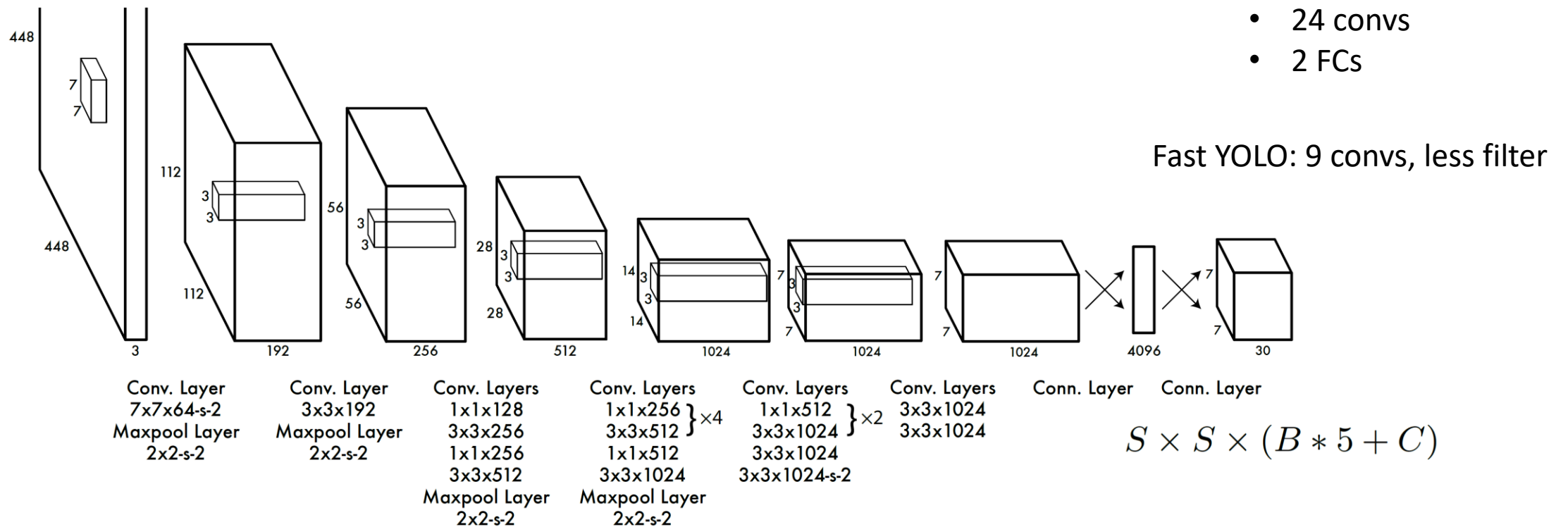
In testing, class-specific confidence scores for each box

$$\Pr(\text{Class}_i | \text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

You Only Look Once: Unified, Real-Time Object Detection. Redmon et al., CVPR, 2016

YOLO

- Regress to bounding box locations and class probabilities



You Only Look Once: Unified, Real-Time Object Detection. Redmon et al., CVPR, 2016

YOLO

- Training loss function

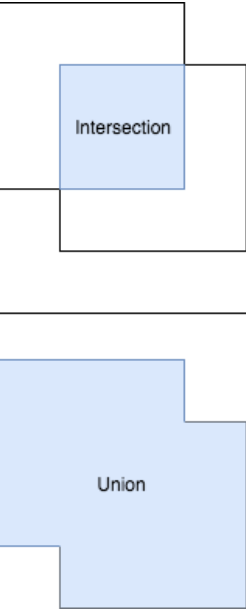
$$\begin{aligned}
 & \mathbb{1}_{ij}^{\text{obj}} \quad \text{jth bounding box from cell i} \\
 & \quad \text{“responsible” for the prediction} \\
 & \quad \text{highest current IOU with the ground truth} \\
 & \mathbb{1}_i^{\text{obj}} \quad \text{If object appears in cell i} \\
 & \lambda_{\text{coord}} = 5 \quad \lambda_{\text{noobj}} = .5
 \end{aligned}$$

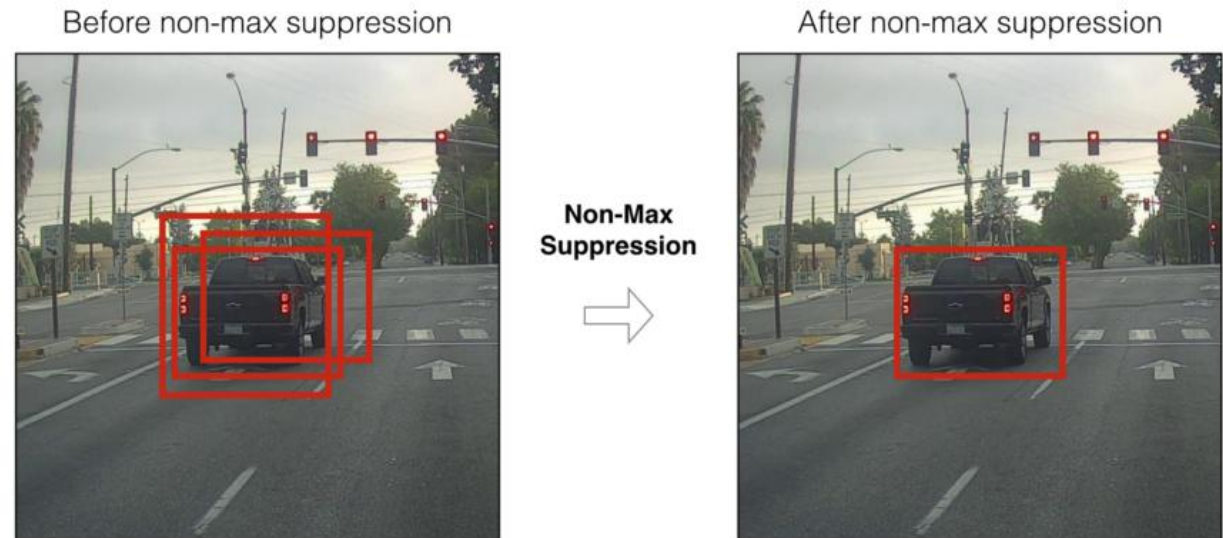
$$\begin{aligned}
 & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
 & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
 & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\
 & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\
 & + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
 \end{aligned}$$

You Only Look Once: Unified, Real-Time Object Detection. Redmon et al., CVPR, 2016

Non-maximum Suppression

- Keep the box with the highest confidence/score
- Compute IoU between this box and other boxes
- Suppress boxes with $\text{IoU} > \text{threshold}$

$$\text{IoU} = \frac{\text{Intersection}}{\text{Union}}$$




<https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c>

YOLO

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	155
YOLO	2007+2012	63.4	45
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

You Only Look Once: Unified, Real-Time Object Detection. Redmon et al., CVPR, 2016

YOLOv2 and YOLOv3

- YOLOv2

- Batch normalization (normalization of the layers' inputs by re-centering and re-scaling)
- High resolution classifier 416x416
- Convolutional with anchor boxes (remove FC layers)
- Dimension clustering to decide the anchor boxes
- Bounding box regression
- Multi-scale training (change input image size)

- YOLOv3

- Binary cross-entropy loss for the class predictions
- Prediction across scales

YOLO9000: Better, Faster, Stronger. Redmon & Farhadi, CVPR, 2017

YOLOv3: An Incremental Improvement

	Type	Filters	Size	Output
	Convolutional	32	3×3	256×256
	Convolutional	64	$3 \times 3 / 2$	128×128
1x	Convolutional	32	1×1	128×128
	Convolutional	64	3×3	
	Residual			
	Convolutional	128	$3 \times 3 / 2$	64×64
2x	Convolutional	64	1×1	64×64
	Convolutional	128	3×3	
	Residual			
	Convolutional	256	$3 \times 3 / 2$	32×32
8x	Convolutional	128	1×1	32×32
	Convolutional	256	3×3	
	Residual			
	Convolutional	512	$3 \times 3 / 2$	16×16
8x	Convolutional	256	1×1	16×16
	Convolutional	512	3×3	
	Residual			
	Convolutional	1024	$3 \times 3 / 2$	8×8
4x	Convolutional	512	1×1	8×8
	Convolutional	1024	3×3	
	Residual			
	Avgpool		Global	
	Connected		1000	
	Softmax			

Table 1. **Darknet-53.**

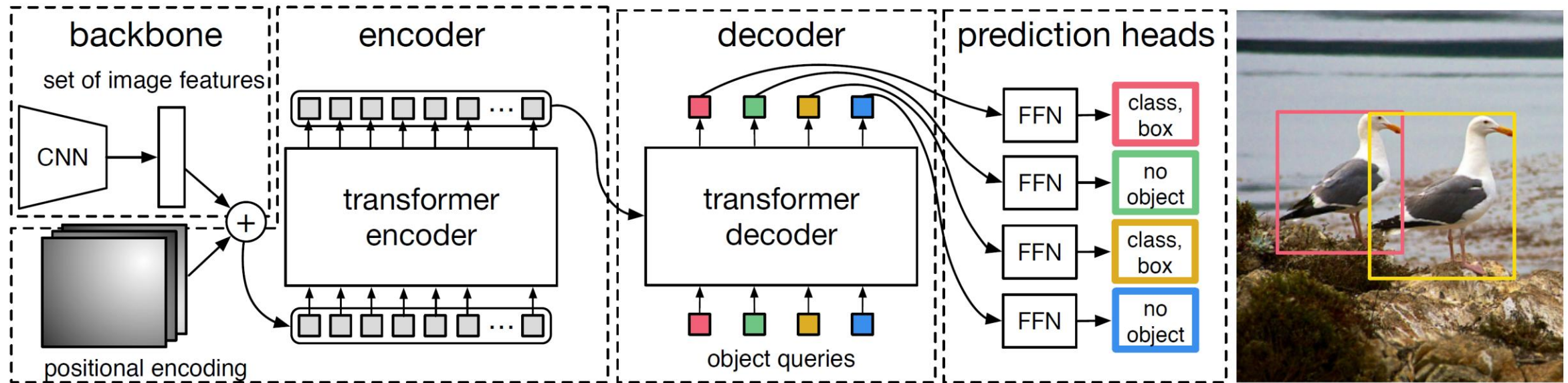
YOLOv8



<https://www.youtube.com/watch?v=QgF5PHDCwHw>

DETR

- Vision transformer-based object detection



End-to-End Object Detection with Transformers. Carion et al., ECCV, 2020

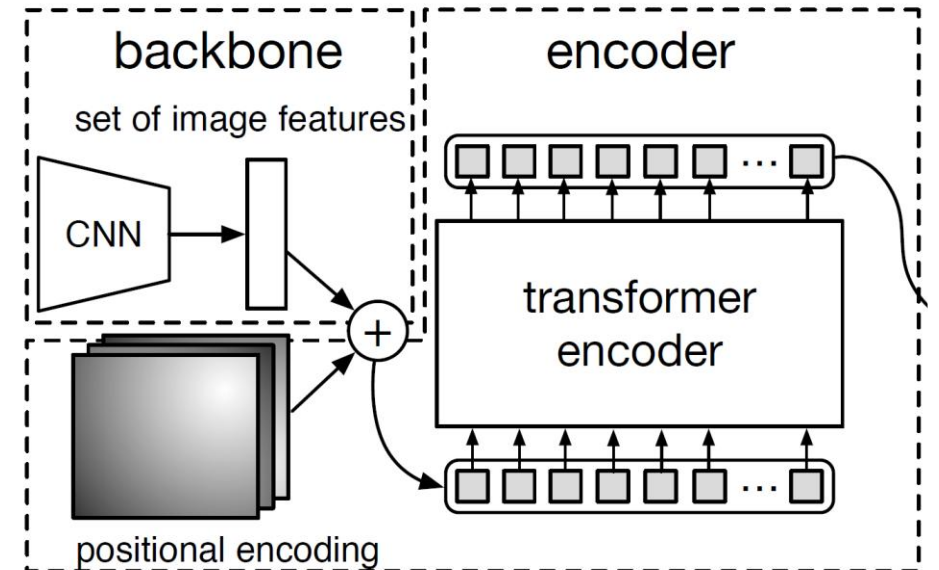
DETR

- Backbone

$$x_{\text{img}} \in \mathbb{R}^{3 \times H_0 \times W_0} \xrightarrow{\text{red arrow}} f \in \mathbb{R}^{C \times H \times W}$$
$$C = 2048 \quad H, W = \frac{H_0}{32}, \frac{W_0}{32}$$

- Encoder

- 1x1 conv on f $z_0 \in \mathbb{R}^{d \times H \times W}$
- $H \times W$ tokens with d -dimension each



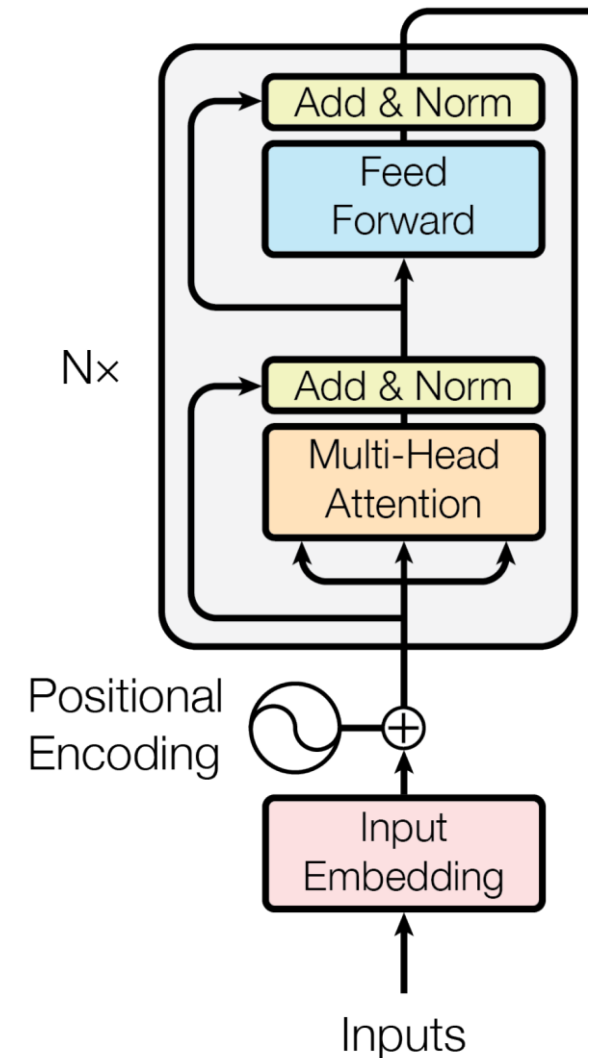
End-to-End Object Detection with Transformers. Carion et al., ECCV, 2020

Transformer: Encoder

- Positional encoding
 - Make use the order of the sequence
 - With dimension d_{model} for each input

$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$

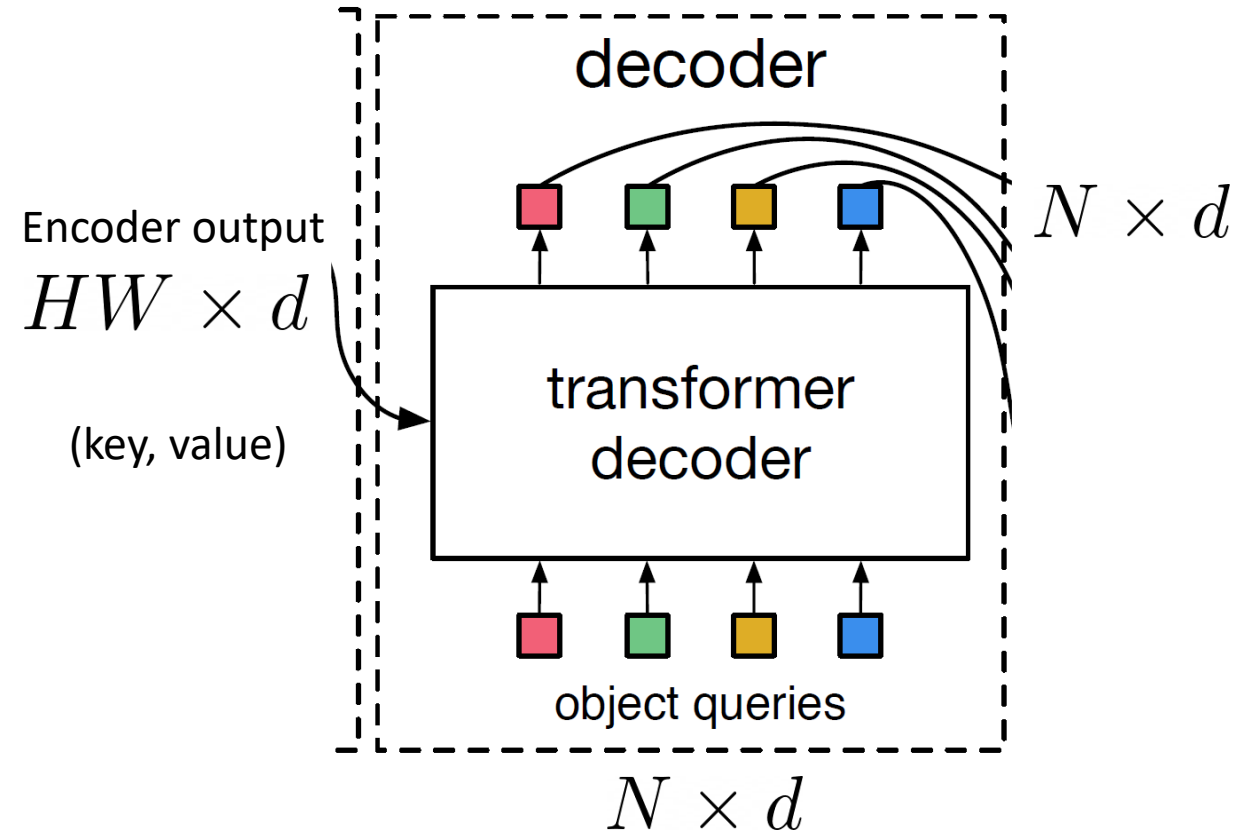
$$PE_{(pos, 2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$



Attention is all you need. Vaswani et al., NeurIPS'17

DETR

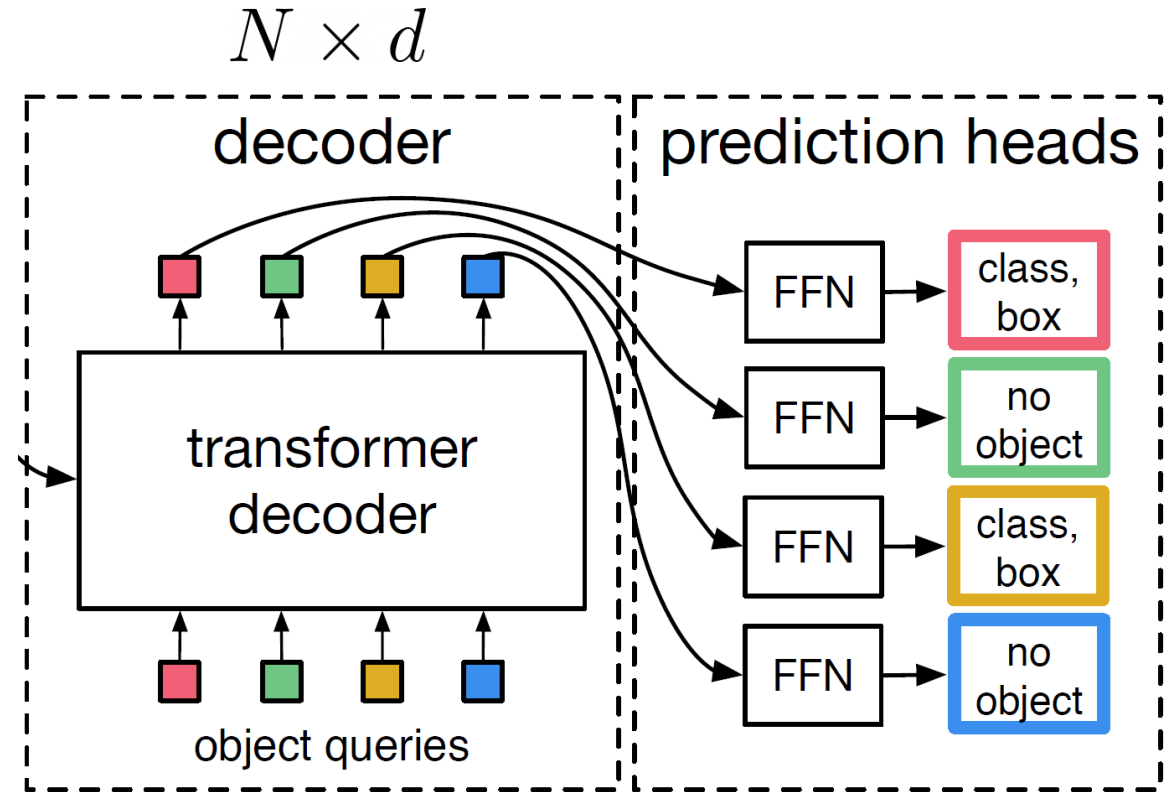
- Decoder
 - Decodes N object queries in parallel
 - Object queries: learned positional encodings (treat as weights in the network)



End-to-End Object Detection with Transformers. Carion et al., ECCV, 2020

DETR

- Prediction heads
 - 3 FC layers
- Box: normalized (x, y, h, w) w.r.t. the input image
- Class: softmax prediction with the “no object” class (that no object is detected within a slot)



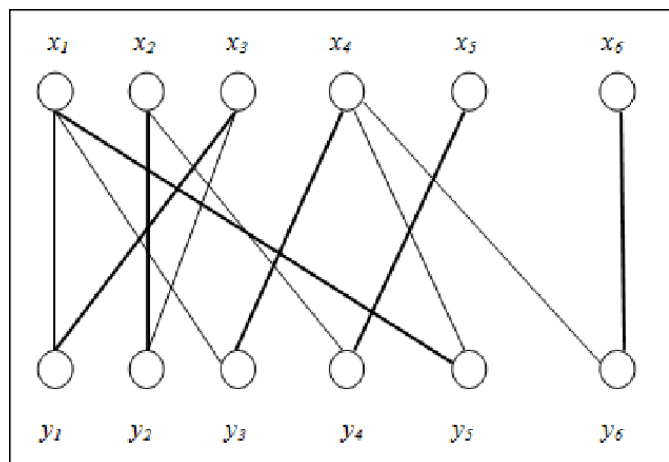
End-to-End Object Detection with Transformers. Carion et al., ECCV, 2020

DETR

- Training
 - bipartite matching between predicted and ground truth objects

Predicted boxes $\hat{y} = \{\hat{y}_i\}_{i=1}^N$

Ground truth boxes $y = \{y_i\}_{i=1}^N$
padded with non-object



Hungarian algorithm

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

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

End-to-End Object Detection with Transformers. Carion et al., ECCV, 2020

DETR

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

DC5: dilated C5 stage

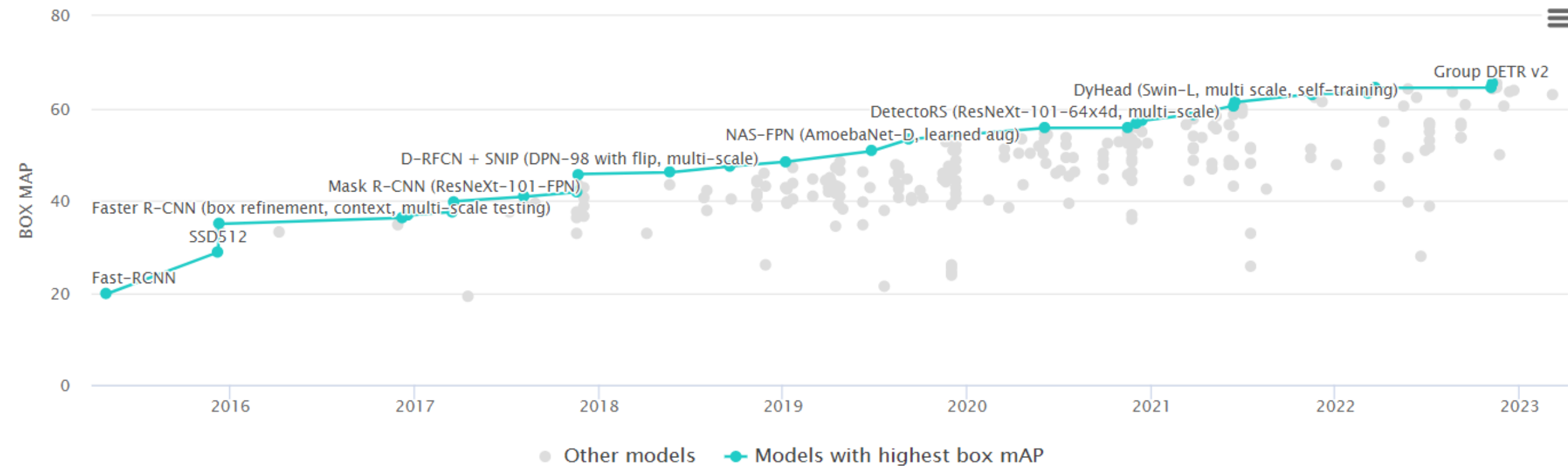
FPN: Feature pyramid networks

End-to-End Object Detection with Transformers. Carion et al., ECCV, 2020

Summary

- Two-stage detectors
 - R-CNN, Fast R-CNN, Faster R-CNN
 - Region proposal + classification
 - Good performance, slow
- One-stage detectors
 - YOLO, SSD
 - End-to-end network to regress to bounding boxes
 - Fast, comparable performance to two-stage detectors
- Transformer-based detectors
 - DETR
 - Attention-based set prediction, using object queries

Object Detection on COCO test-dev



<https://paperswithcode.com/sota/object-detection-on-coco>

Further Reading

- Viola–Jones object detection, 2001
<https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf>
- Deformable part model, 2010,
<https://ieeexplore.ieee.org/document/5255236>
- R-CNN, 2014 <https://arxiv.org/abs/1311.2524>
- Fast R-CNN, 2015 <https://arxiv.org/abs/1504.08083>
- Faster R-CNN, 2015 <https://arxiv.org/abs/1506.01497>
- YOLO, 2015 <https://arxiv.org/abs/1506.02640>
- YOLOv2, 2016 <https://arxiv.org/abs/1612.08242>
- Feature Pyramid Networks, 2017 <https://arxiv.org/pdf/1612.03144.pdf>
- DETR, 2020 <https://arxiv.org/abs/2005.12872>