Lecture 1.5
State-of-the-art Object Detection

Dimitrios Papadopoulos

Assistant Professor, DTU Compute

02514 Deep Learning in Computer Vision, 09/06/22

Last time – Object detection

- Various Problem Formulations
- General Strategies for Object Detection
- R-CNN, Fast R-CNN, Faster R-CNN
- Comparing Boxes
- Evaluating Object Detectors
- Project 1.2

Today

- Single-stage Object Detection

- State-of-the-art Object Detection

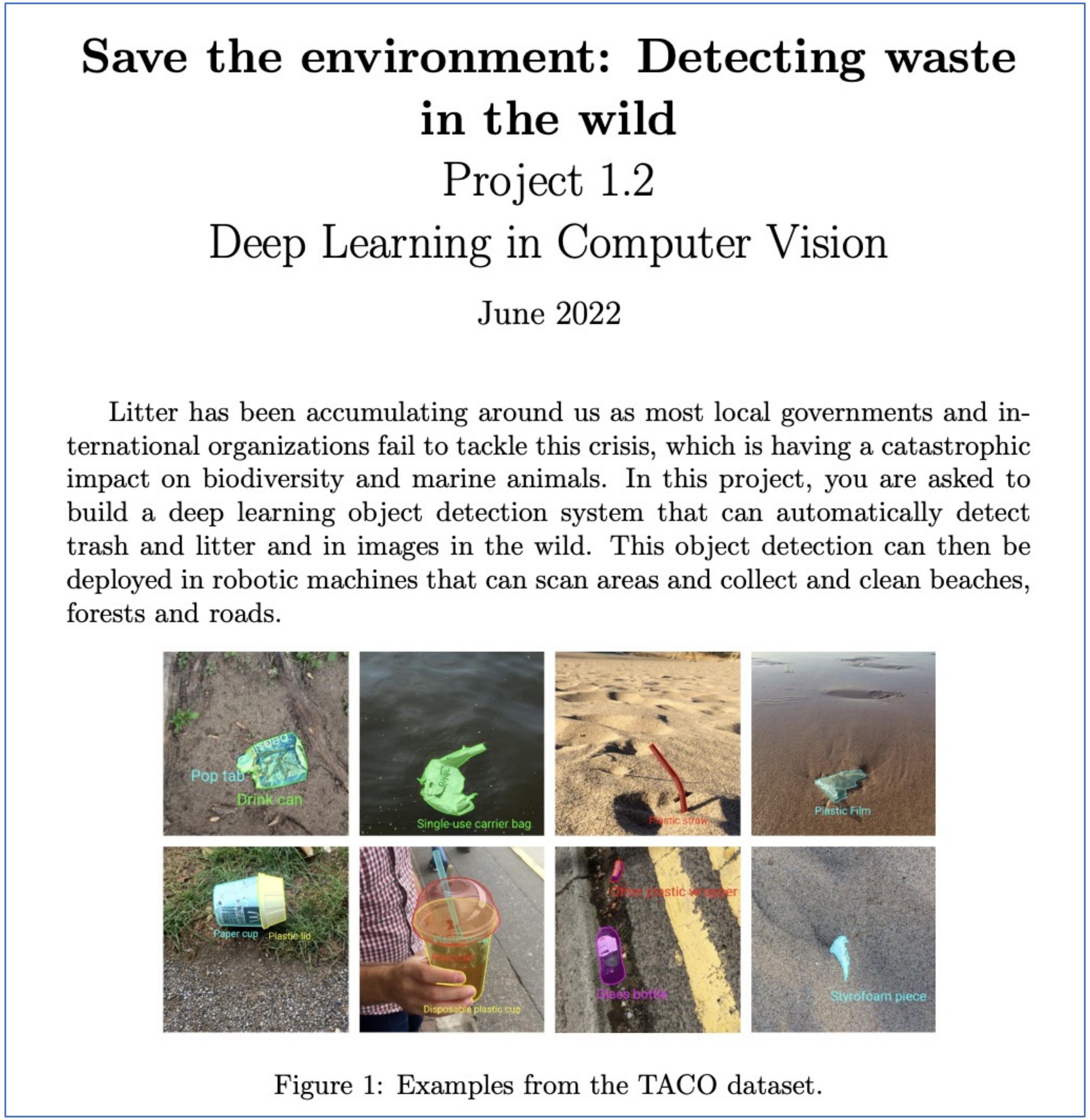
- Speed vs Accuracy

Project 1.2

Detecting waste in the wild

Tasks for simple object detector:

- Extract object proposals
- Finetune a CNN for object detector on object proposals (replace last layer)
- Apply the model onE test images
- Implement NMS
- Evaluate the object detection performance



Save the environment: Detecting waste in the wild

Project 1.2

Deep Learning in Computer Vision

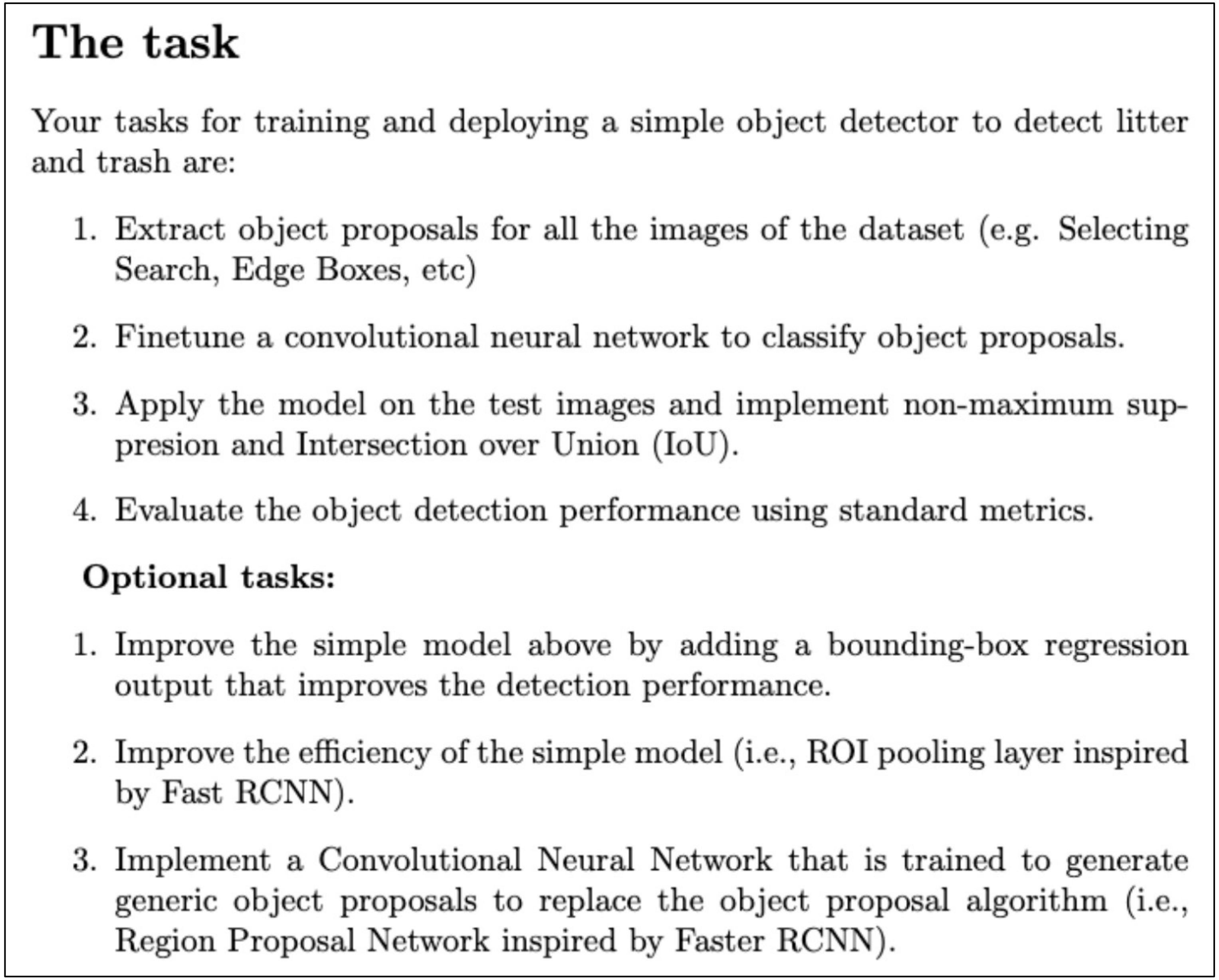
June 2022

Litter has been accumulating around us as most local governments and international organizations fail to tackle this crisis, which is having a catastrophic impact on biodiversity and marine animals. In this project, you are asked to build a deep learning object detection system that can automatically detect trash and litter and in images in the wild. This object detection can then be deployed in robotic machines that can scan areas and collect and clean beaches, forests and roads.

Figure 1: Examples from the TACO dataset.

Project 1.2

Detecting waste in the wild



The task

Your tasks for training and deploying a simple object detector to detect litter and trash are:

1. Extract object proposals for all the images of the dataset (e.g. Selecting Search, Edge Boxes, etc)
2. Finetune a convolutional neural network to classify object proposals.
3. Apply the model on the test images and implement non-maximum suppresion and Intersection over Union (IoU).
4. Evaluate the object detection performance using standard metrics.

Optional tasks:

1. Improve the simple model above by adding a bounding-box regression output that improves the detection performance.
2. Improve the efficiency of the simple model (i.e., ROI pooling layer inspired by Fast RCNN).
3. Implement a Convolutional Neural Network that is trained to generate generic object proposals to replace the object proposal algorithm (i.e., Region Proposal Network inspired by Faster RCNN).

Fundamental Tasks in CV

- Human-level understanding and perception of the world, *e.g.* what, where, why, *etc*
- Representing objects in the image:
  - Class labels
  - Bounding box
  - Semantic pixel-wise labels
  - Instance pixel-wise labels

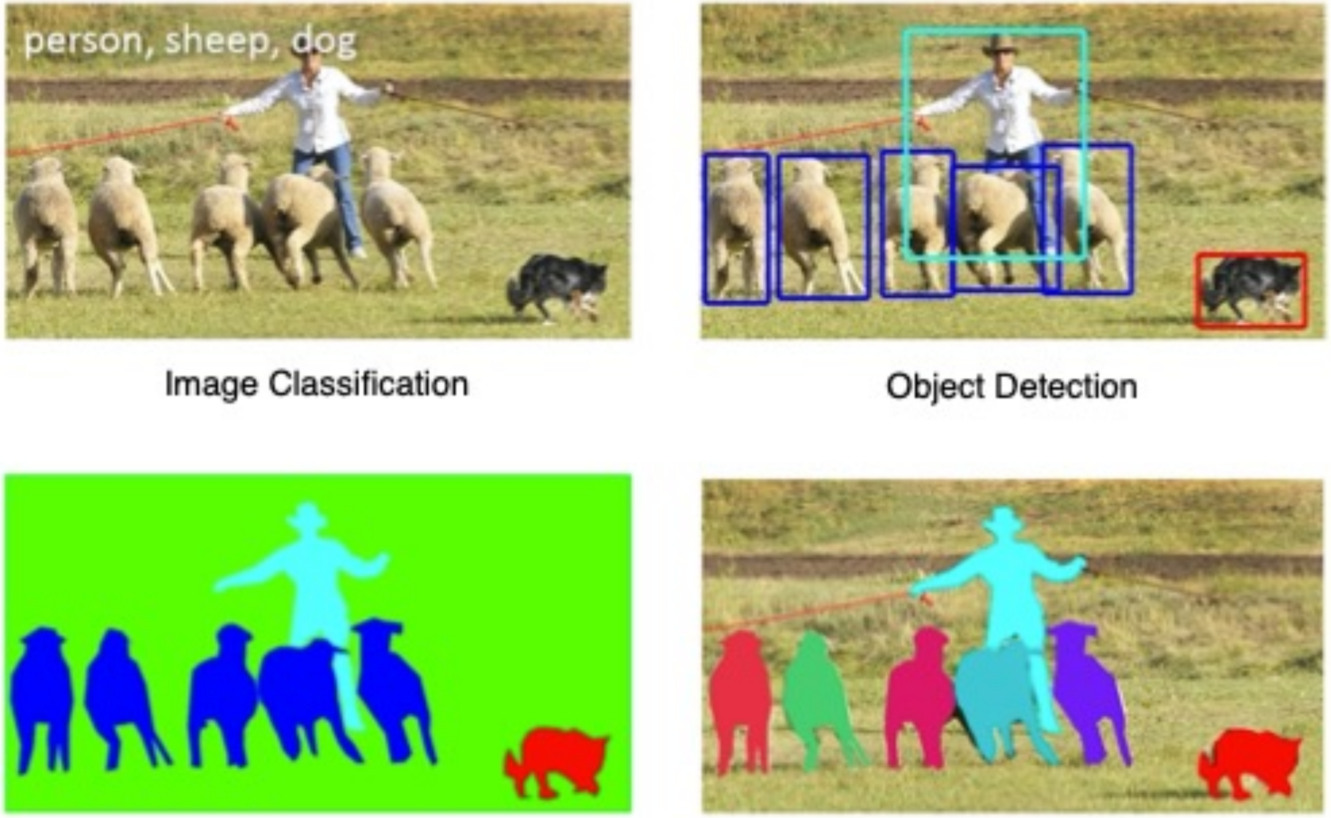


Image Classification

Object Detection

Semantic Segmentation

Instance Segmentation

[Lin et al, MSCOCO]

Architectures

- **Two-stage architectures**
  - Propose large number of regions with **high recall**, meaning all potential objects have been included
  - Classify the regions as object category or background
  - Possibly slow because of the two steps
  - Examples: RCNN, Fast-RCNN, Faster-RCNN, Mask-RCNN, …

Pop Quiz Answer

DOG, (x, y, w, h)

CAT, (x, y, w, h)

CAT, (x, y, w, h)

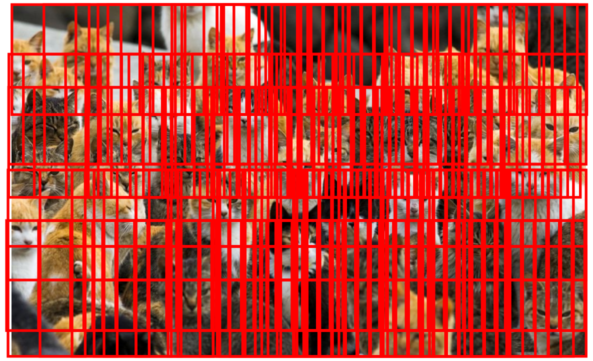
DUCK (x, y, w, h)

= 16 numbers

DOG, (x, y, w, h)

CAT, (x, y, w, h)

= 8 numbers



CAT, (x, y, w, h)

CAT, (x, y, w, h)

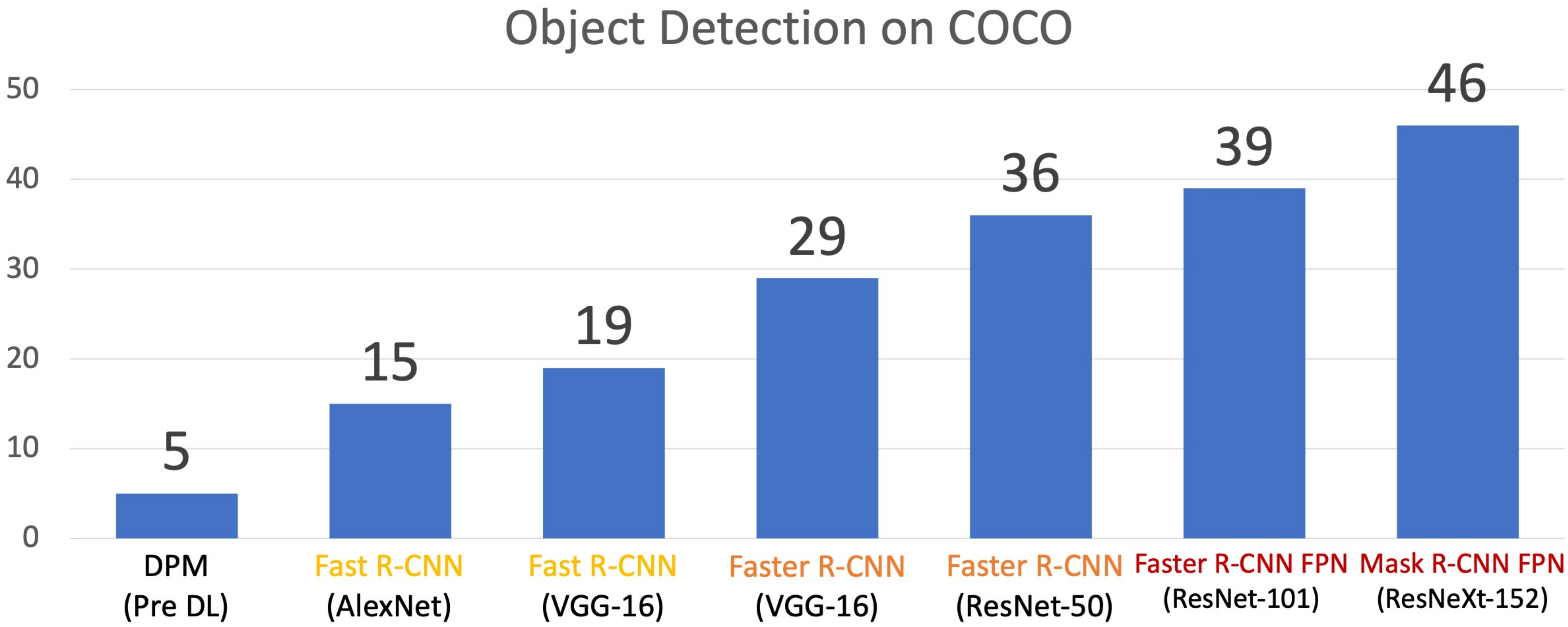
....

CAT (x, y, w, h)

= many numbers

Need variable sized outputs!
Sliding window → computationally prohibited

Object Detection



Ross Girshick, “The Generalized R-CNN Framework for Object Detection”, ICCV 2017 Tutorial on Instance-Level Visual Recognition

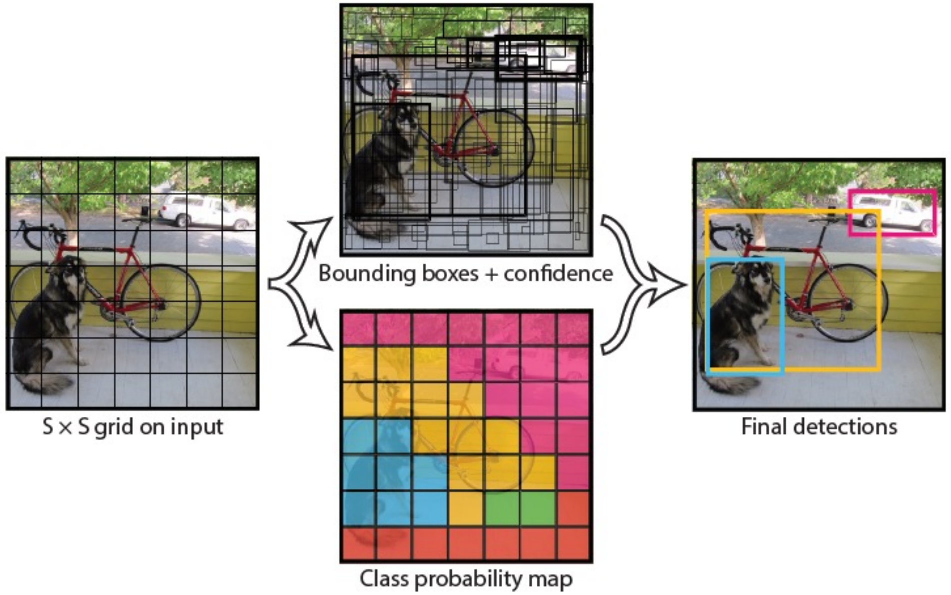
Architectures

- **Two-stage architectures**
  - Propose large number of regions with **high recall**, meaning all potential objects have been included
  - Classify the regions as object category or background
  - Possibly slow because of the two steps
  - Examples: RCNN, Fast-RCNN, Faster-RCNN, Mask-RCNN, …

- **One-stage**
  - Regions are built into the architecture (fully convolutional layers)
  - Can be fast
  - Examples:
    - anchor based, *e.g.* YOLO, SSD, RetinaNet, EfficientDet (CVPR2020)
    - point based, *e.g.* CornerNet, CenterNet, FCOS

YOLO – You Only Look Once

Idea:
No sliding windows →
Predict a class and a box for every location in a grid.



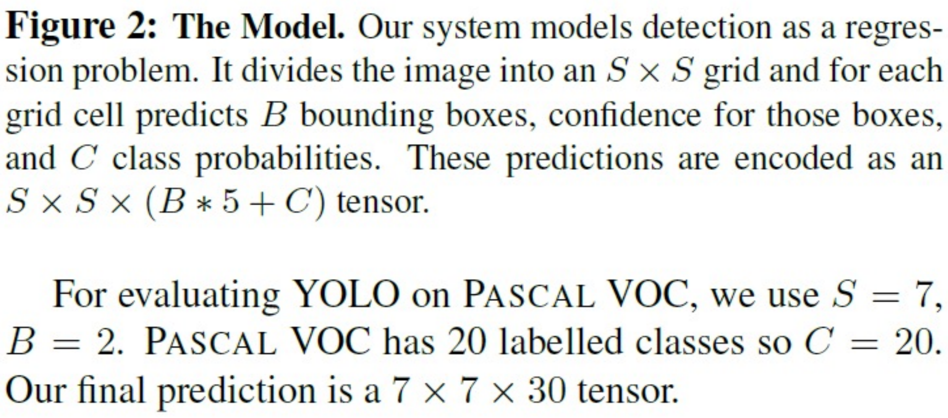


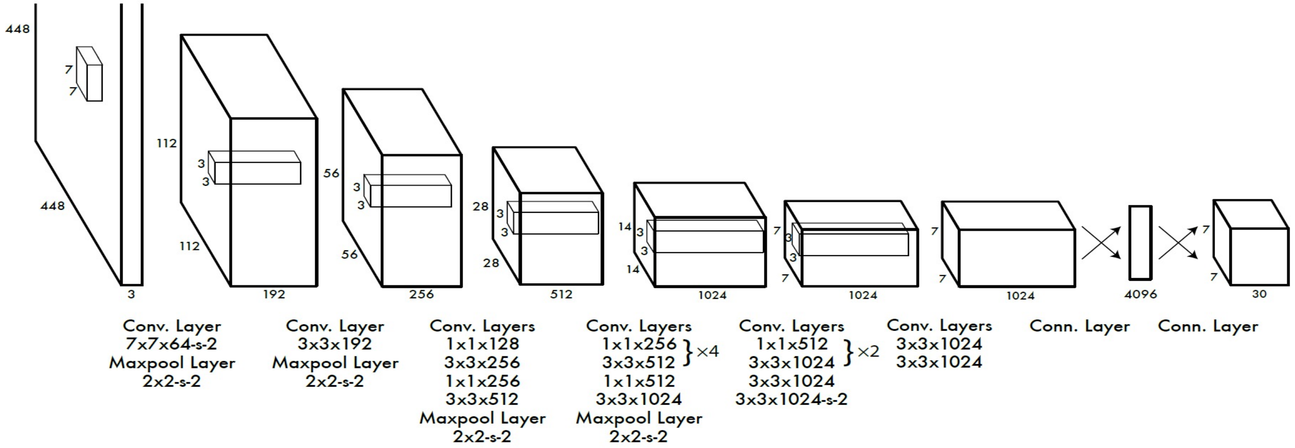
Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

For evaluating YOLO on PASCAL VOC, we use $S = 7$, $B = 2$. PASCAL VOC has 20 labelled classes so $C = 20$. Our final prediction is a $7 \times 7 \times 30$ tensor.

[You Only Look Once: Unified, Real-Time Object Detection Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi CVPR 2016]

YOLO – You Only Look Once

- Network inspired by GoogLeNet without Inception modules
- Pre-trained on ImageNet
- Network objective: predict 98 bounding boxes, confidence scores and 49 object classes

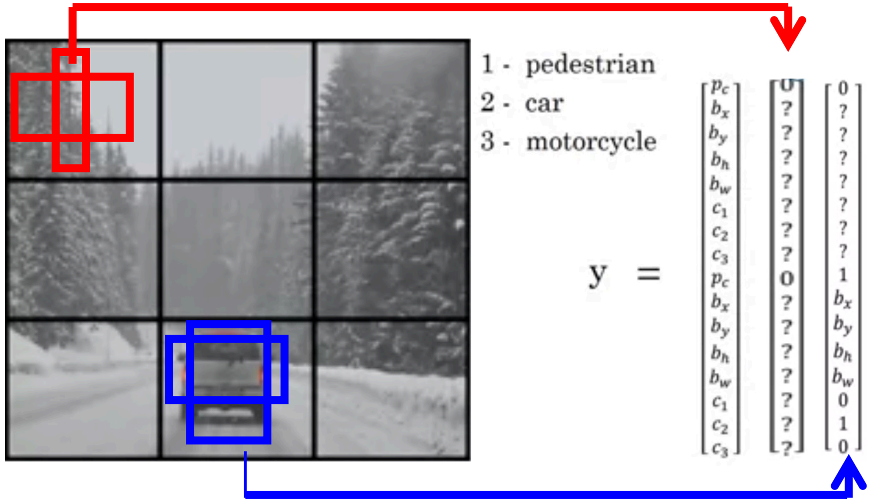

448
7
7
448
3
112
3
3
112
192
56
3
3
56
256
28
3
3
28
512
14
3
3
14
1024
7
3
3
7
1024
7
7
1024
4096
7
7
30
Conv. Layer 7x7x64-s-2 Maxpool Layer 2x2-s-2
Conv. Layer 3x3x192 Maxpool Layer 2x2-s-2
Conv. Layers 1x1x128 3x3x256 1x1x256 3x3x512 Maxpool Layer 2x2-s-2
Conv. Layers 1x1x256 3x3x512 }×4 1x1x512 3x3x1024 Maxpool Layer 2x2-s-2
Conv. Layers 1x1x512 3x3x1024 }×2 3x3x1024 3x3x1024-s-2
Conv. Layers 3x3x1024 3x3x1024
Conn. Layer
Conn. Layer

[You Only Look Once: Unified, Real-Time Object Detection Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi CVPR 2016]

YOLO – You Only Look Once

- **During training:**
  - “Virtually” split the image into a 7x7 grid (3x3 for illustration)
  - Create 2 anchor boxes per cell
  - At each iteration, assign the anchor box with highest IOU to each ground truth box

- **Penalties:**
  - For assigned anchor boxes: coordinates, correct class, confidence (difference from 1)
  - For unassigned boxes: only confidence (difference from 0)

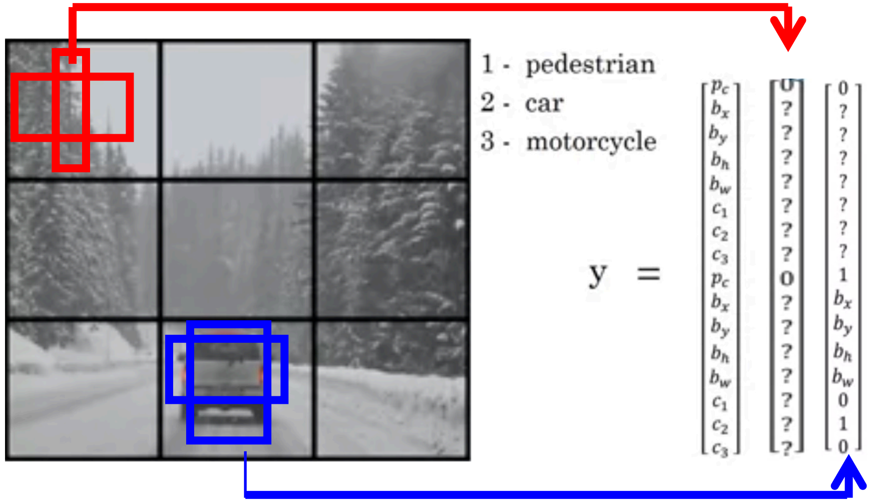


1 - pedestrian
2 - car
3 - motorcycle
y =
p_c b_x b_y b_h b_w c_1 c_2 c_3 p_c b_x b_y b_h b_w c_1 c_2 c_3
0 ? ? ? ? ? ? ? 0 ? ? ? ? ? ? ?
0 ? ? ? ? ? ? ? 1 b_x b_y b_h b_w 0 1 0

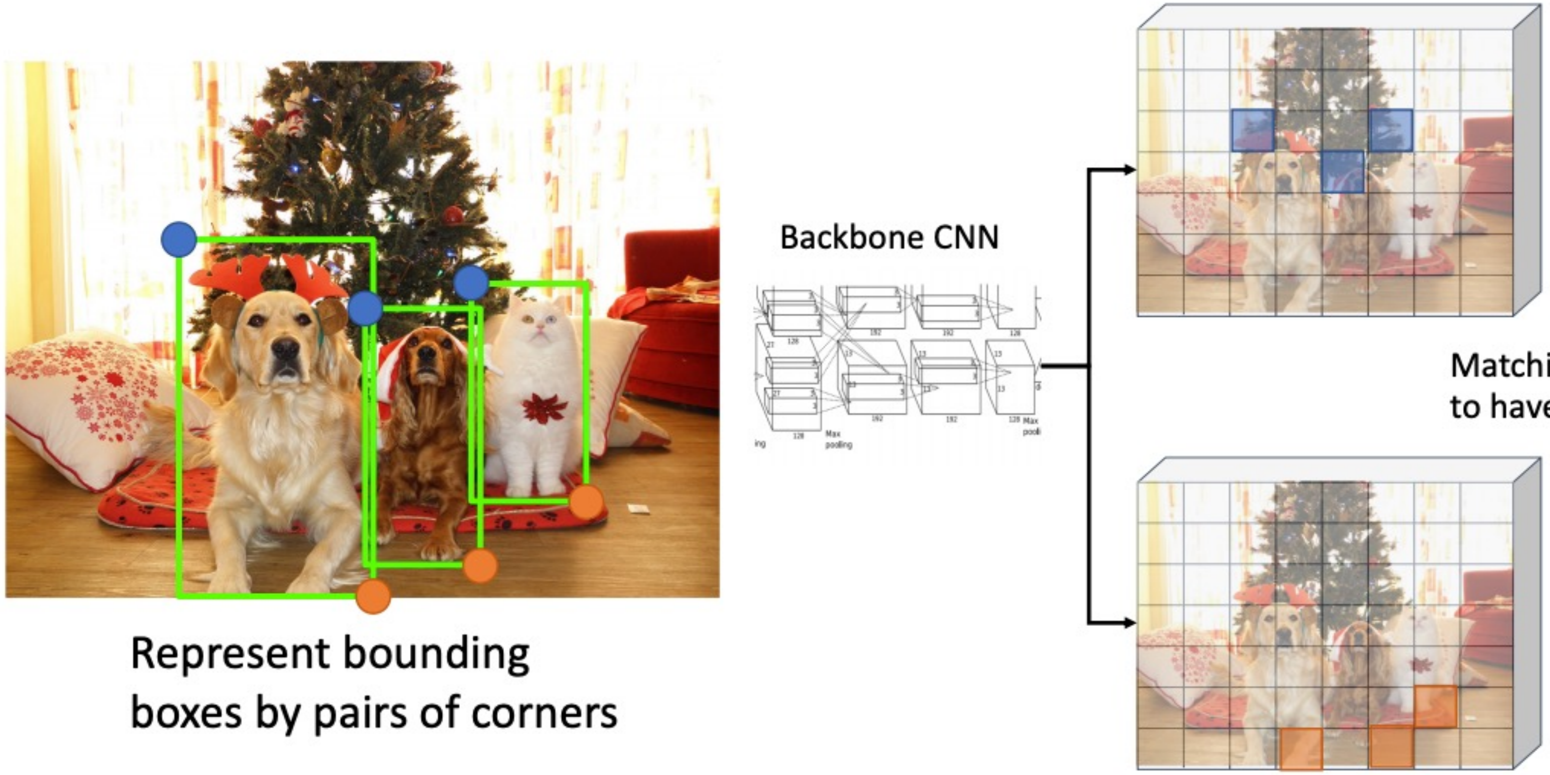
- **Prediction size:**
  - 3 (#grid rows) x 3 (#grid columns) x 2 (#bounding boxes) x 8 (confidence 1, coord 4, class 3) → Constant

YOLO – You Only Look Once

- **End-to-end trainable!**
  - Learn to predict the ground truth vectors → Fixed size output!


1 - pedestrian
2 - car
3 - motorcycle
y =
p_c b_x b_y b_h b_w c_1 c_2 c_3 p_c b_x b_y b_h b_w c_1 c_2 c_3
0 ? ? ? ? ? ? ? 0 ? ? ? ? ? ? ?
0 ? ? ? ? ? ? ? 1 b_x b_y b_h b_w 0 1 0

Detection without anchors: CornerNet



Represent bounding boxes by pairs of corners

Backbone CNN

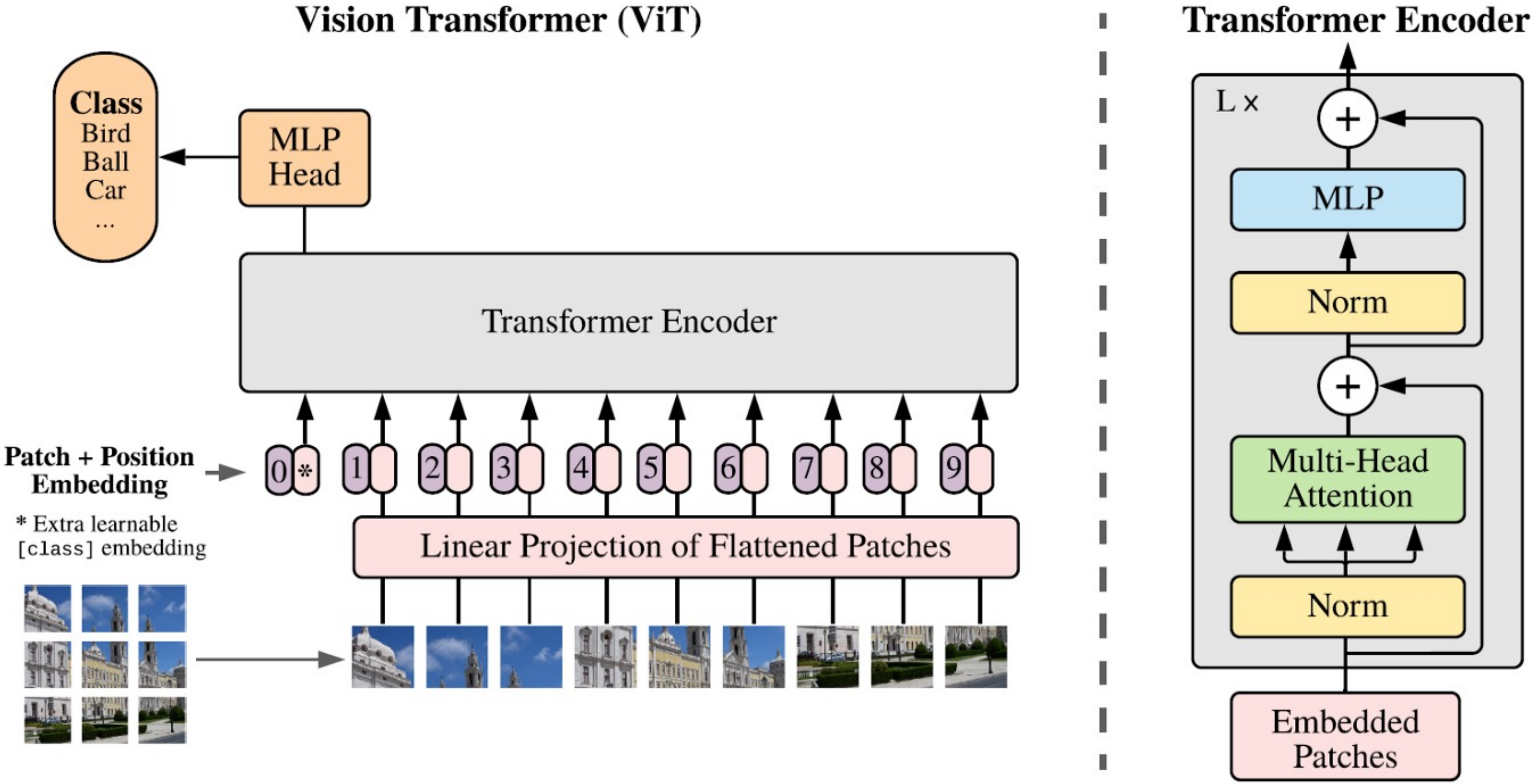
Upper left corners
Heatmap: C x H x W
Embeddings: D x H x W

Matching corners are trained to have similar embeddings

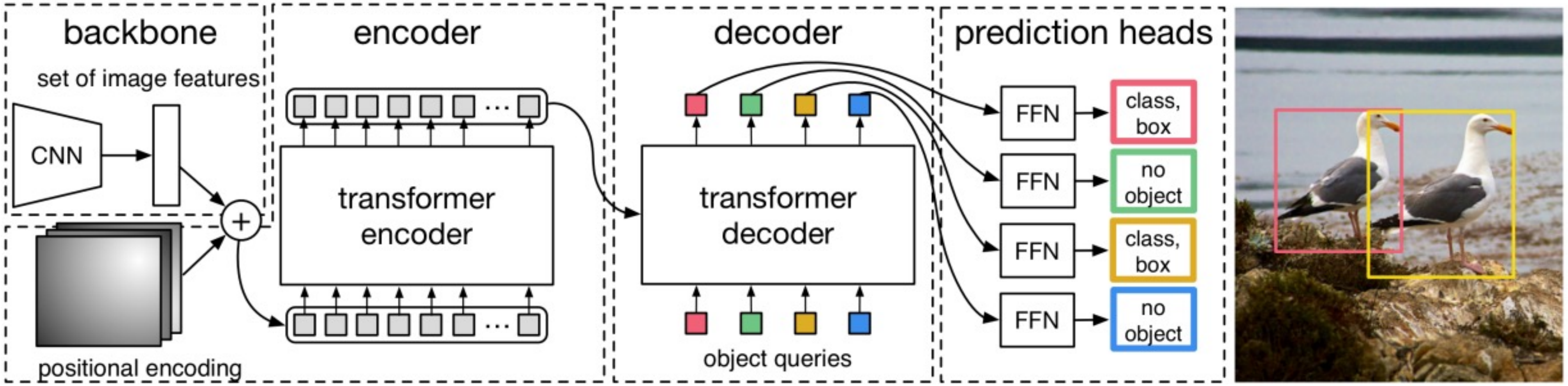
Lower right corners
Heatmap: C x H x W
Embeddings: D x H x W

[Law and Deng “CornerNet: Detecting Objects as Paired Keypoints”, ECCV 2018]

Vision Transformer

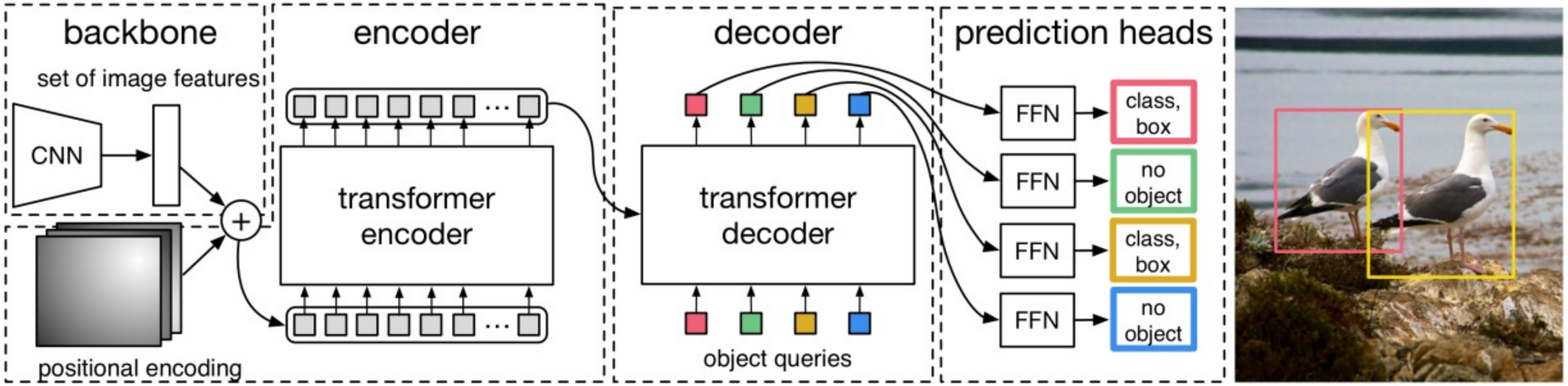


Detection without anchors: Transformers



[Carion et al, "End-to-End Object Detection with Transformers", ECCV 2020]

Detection without anchors: Transformers



(+) No RPN!

(+) No anchors!

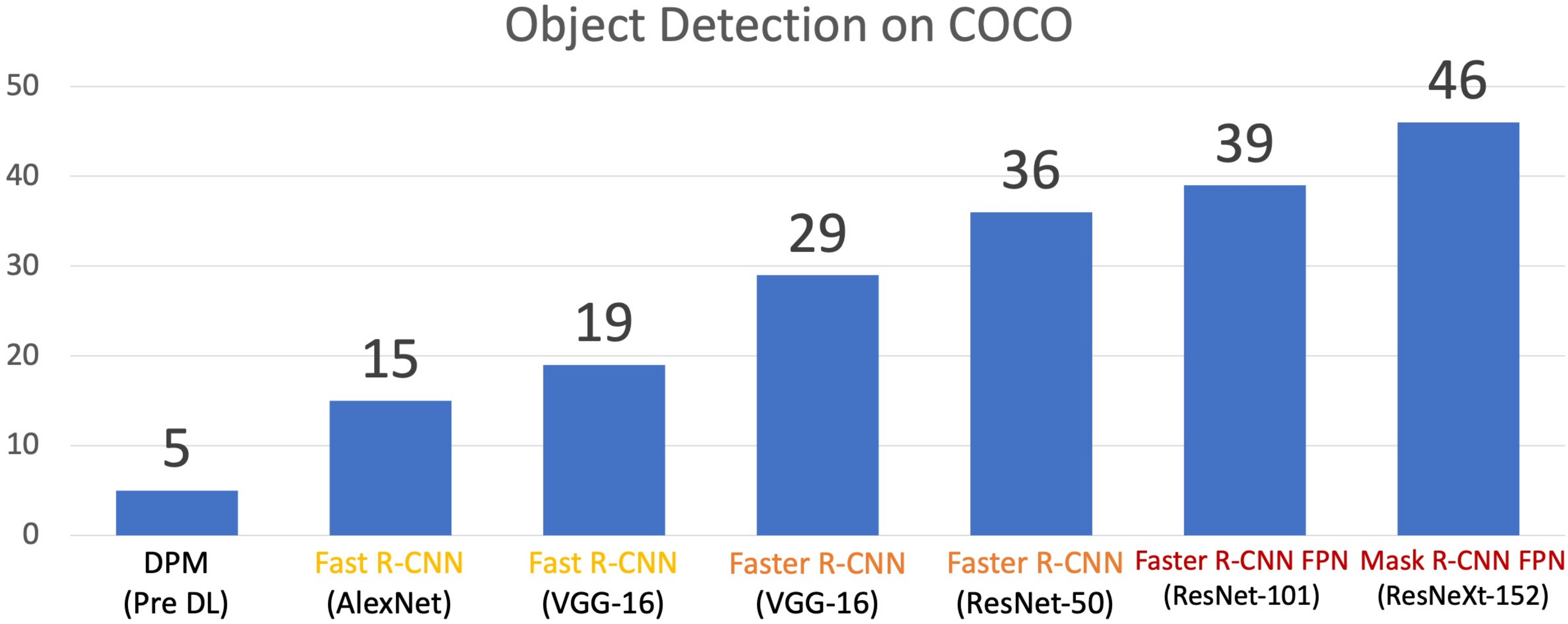
(+) No NMS!

(-) Slow and harsh to train

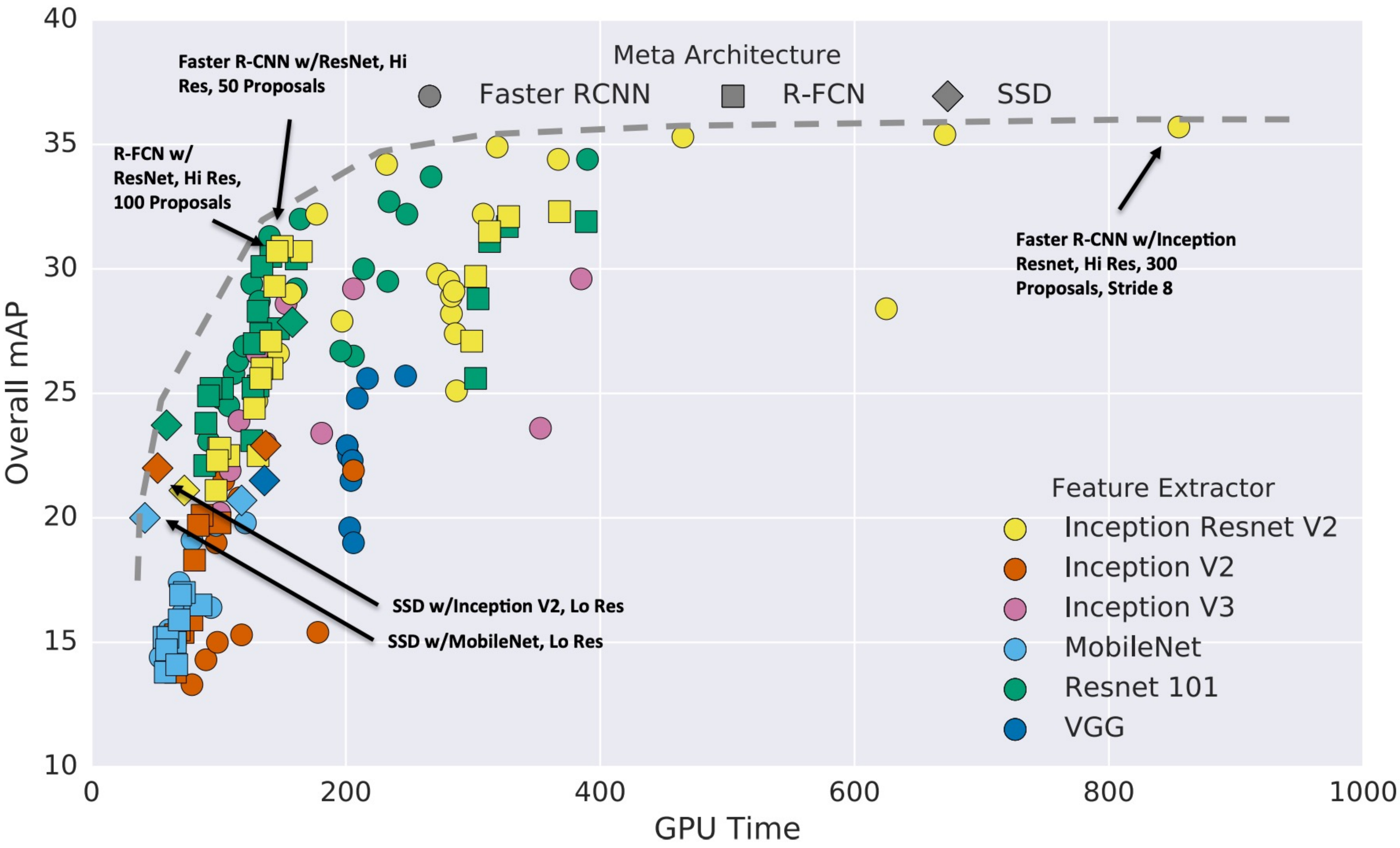
(-) bad performance for small objects

[Carion et al, "End-to-End Object Detection with Transformers", ECCV 2020]

Object Detection

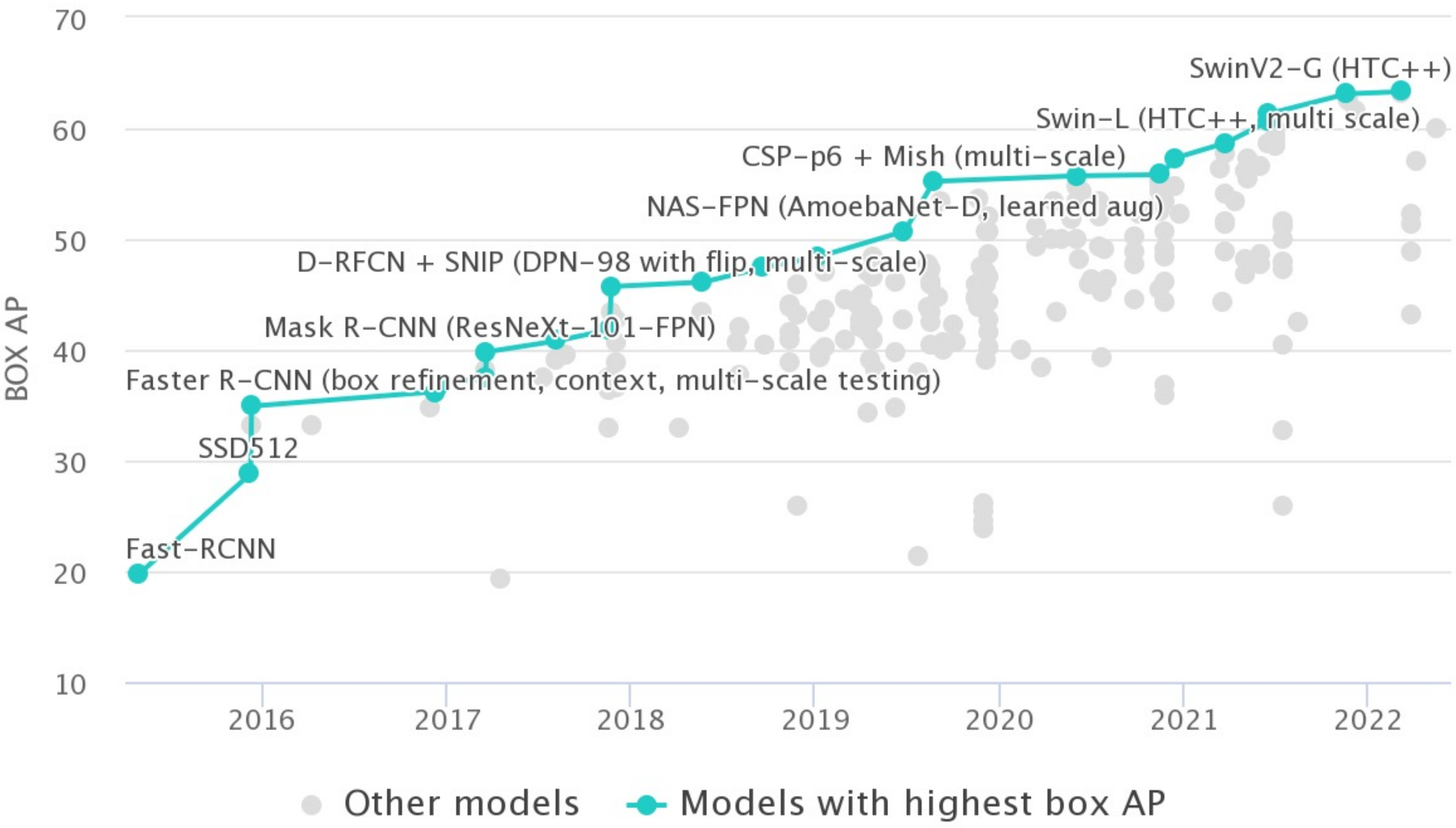


Ross Girshick, “The Generalized R-CNN Framework for Object Detection”, ICCV 2017 Tutorial on Instance-Level Visual Recognition



Huang et al, Speed/accuracy trade-offs for modern convolutional object detectors, CVPR 2017

State-of-the-art Object Detection on COCO



[paperswithcode]

Questions?

Thank you!!!