

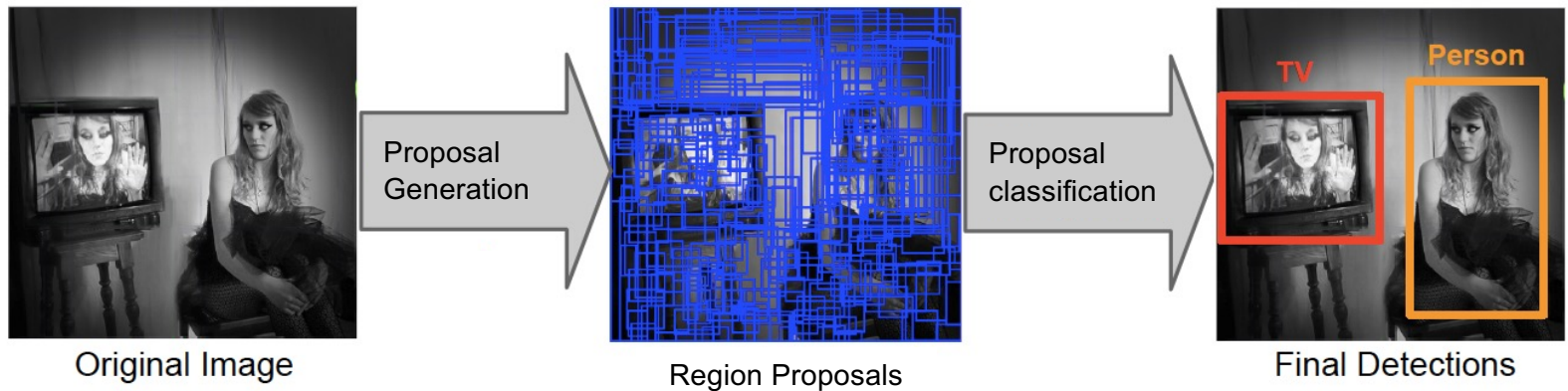
Review: Single-stage vs. two-stage detectors

Review: Single-stage vs. two-stage detectors

Single-stage



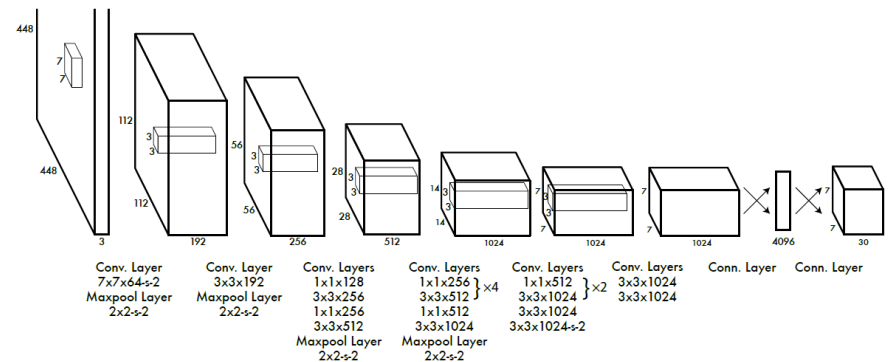
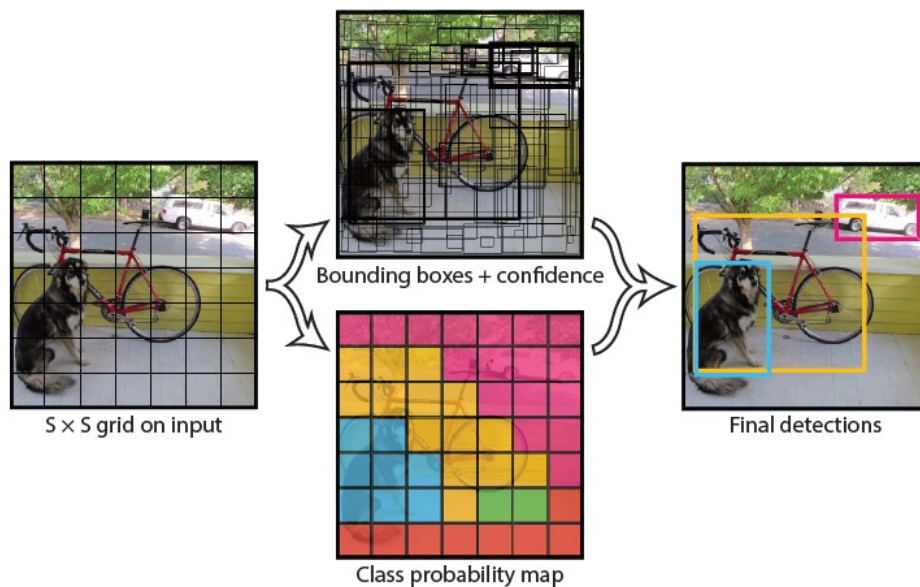
Two-stage



[Image source](#)

Review: YOLO

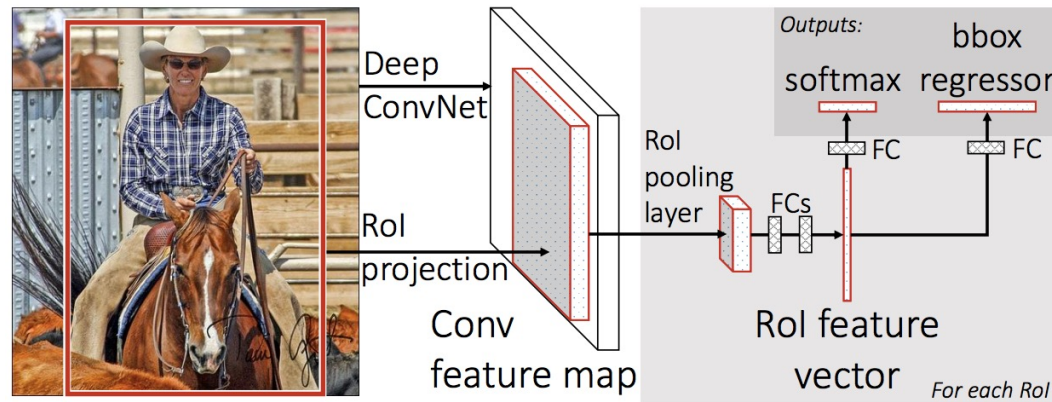
- Divide the image into a coarse grid and directly predict class label and a few candidate boxes for each grid cell



- At each location, predict a score for each class and two bboxes w/ confidences
- For PASCAL, output is $7 \times 7 \times 30$ ($30 = 20 + 2 * (4 + 1)$)

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, [You Only Look Once: Unified, Real-Time Object Detection](#), CVPR 2016

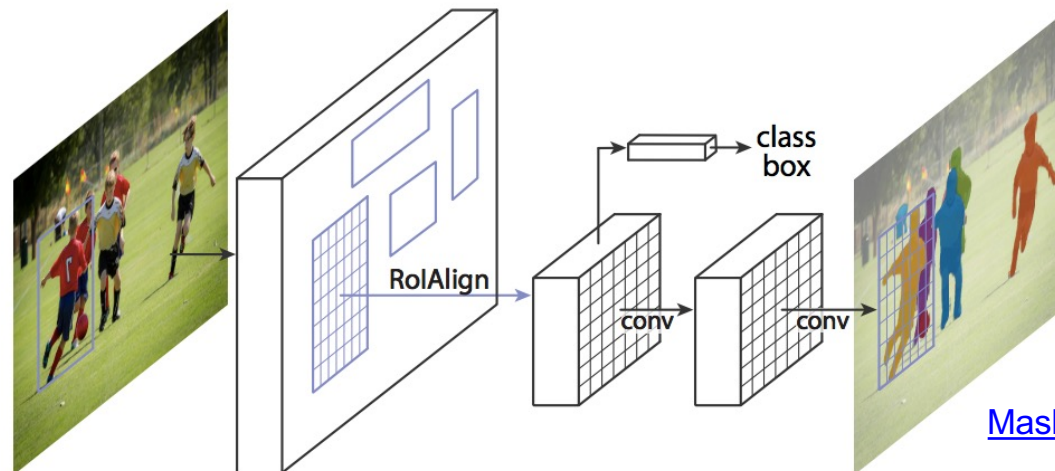
Two-stage detectors: R-CNN family



[R-CNN](#) (Girshick et al., CVPR 2014)

[Fast R-CNN](#) (Girshick, ICCV 2015)

[Faster R-CNN](#) (Ren et al., NeurIPS 2015)

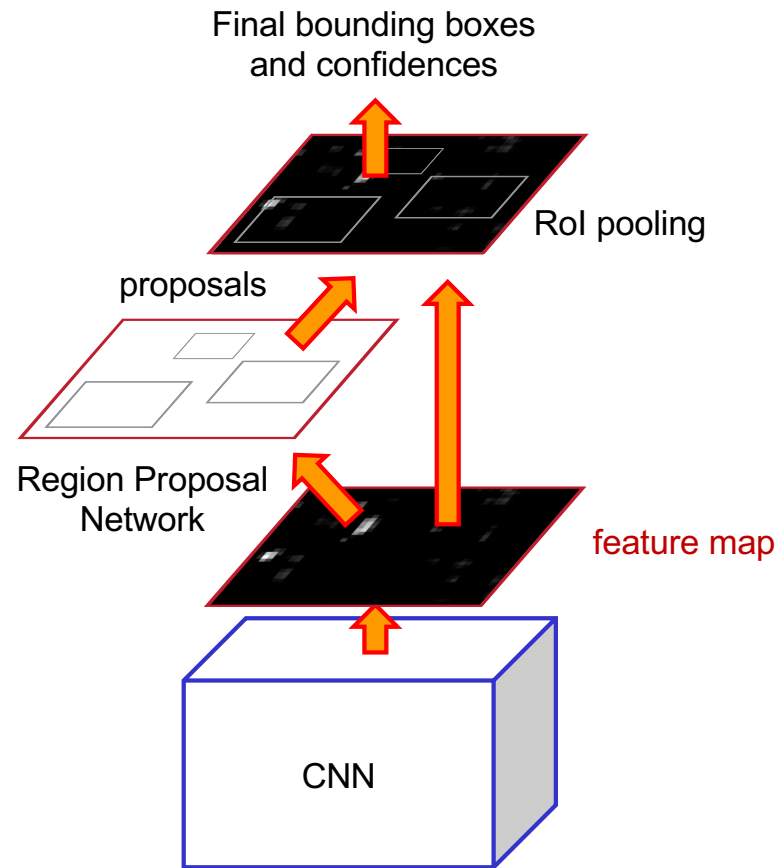


[Mask R-CNN](#) (He et al., ICCV 2017)

Outline

- Faster R-CNN
 - Region proposal network (RPN)
 - RoI pooling
- Mask R-CNN
- Other detectors

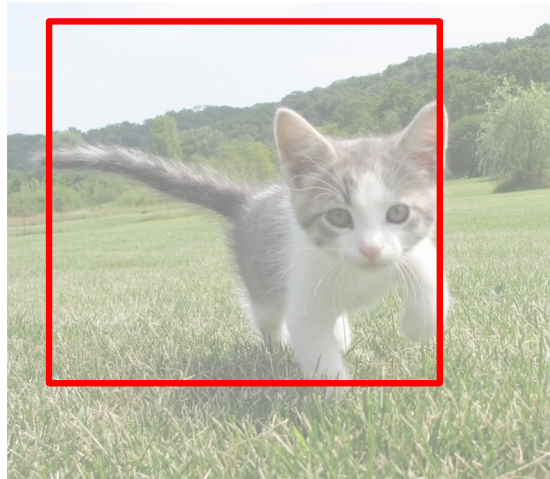
Faster R-CNN



S. Ren et al. [Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks](#). NeurIPS 2015

Region proposal network (RPN)

- Idea: Tile the image with category-independent “anchor boxes” of a set size and try to predict how likely each anchor is to contain an object

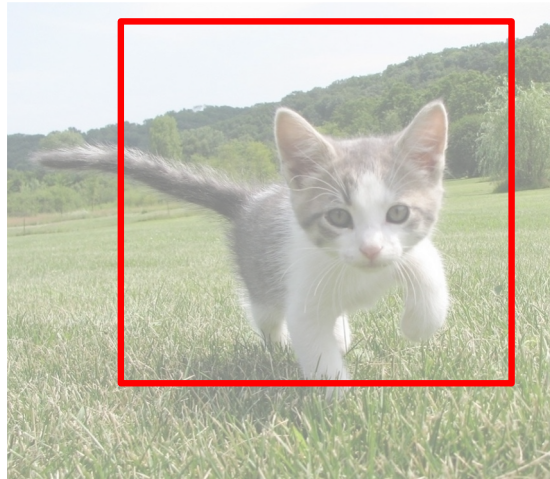


**Anchor is
an object?**

Figure source: [J. Johnson](#)

Region proposal network (RPN)

- Idea: Tile the image with category-independent “anchor boxes” of a set size and try to predict how likely each anchor is to contain an object

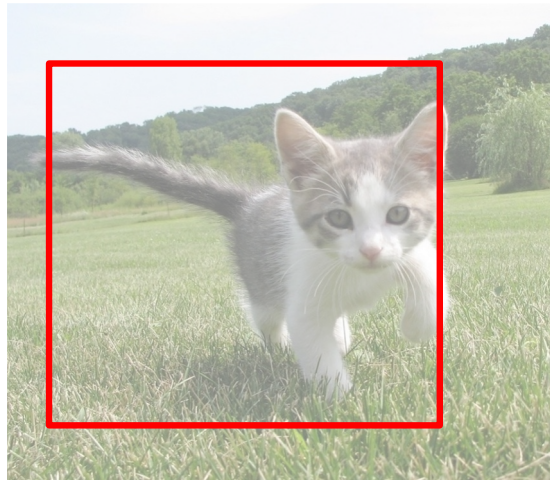


**Anchor is
an object?**

Figure source: [J. Johnson](#)

Region proposal network (RPN)

- Idea: Tile the image with category-independent “anchor boxes” of a set size and try to predict how likely each anchor is to contain an object

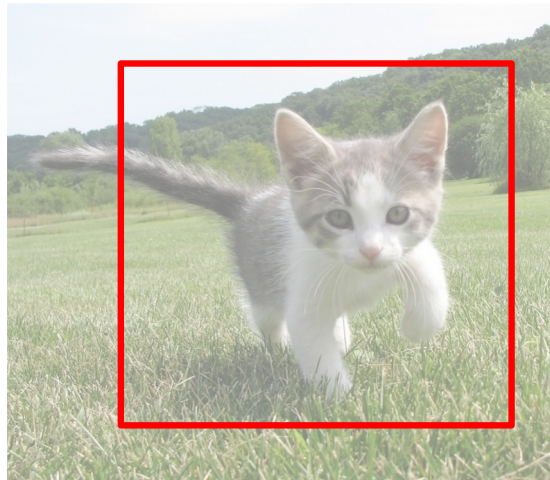


**Anchor is
an object?**

Figure source: [J. Johnson](#)

Region proposal network (RPN)

- Idea: Tile the image with category-independent “anchor boxes” of a set size and try to predict how likely each anchor is to contain an object

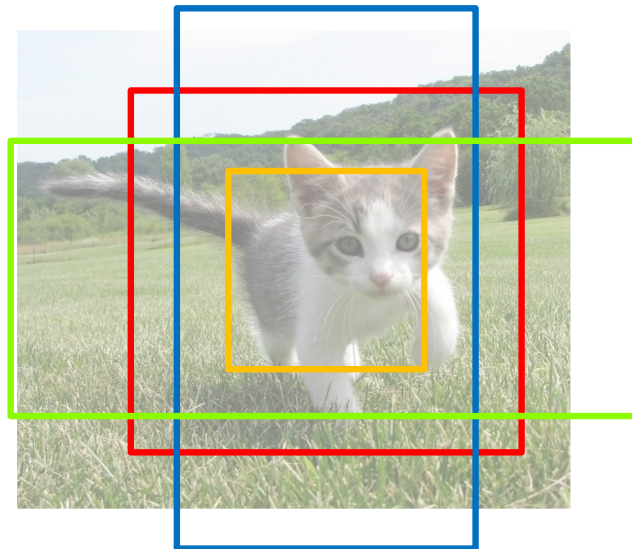


**Anchor is
an object?**

Figure source: [J. Johnson](#)

Region proposal network (RPN)

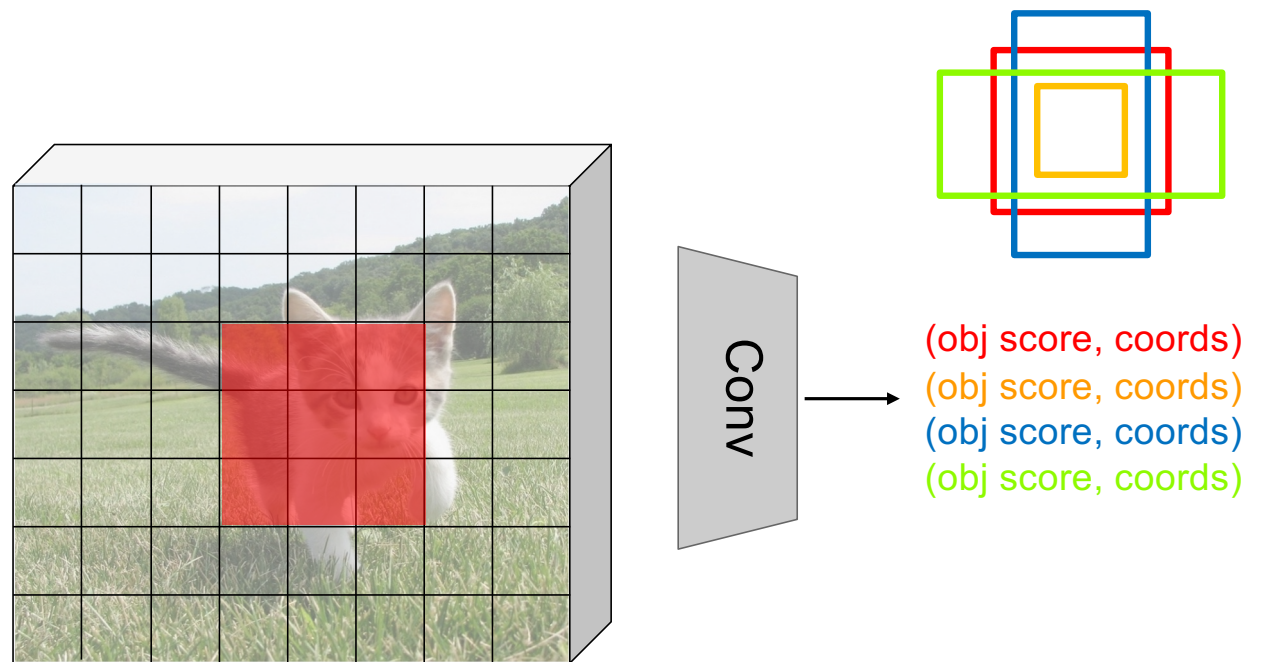
- Idea: Tile the image with category-independent “anchor boxes” of a set size and try to predict how likely each anchor is to contain an object
- Introduce anchor boxes at multiple scales and aspect ratios to handle a wider range of object sizes and shapes



Anchor is an object?
Anchor is an object?
Anchor is an object?
Anchor is an object?

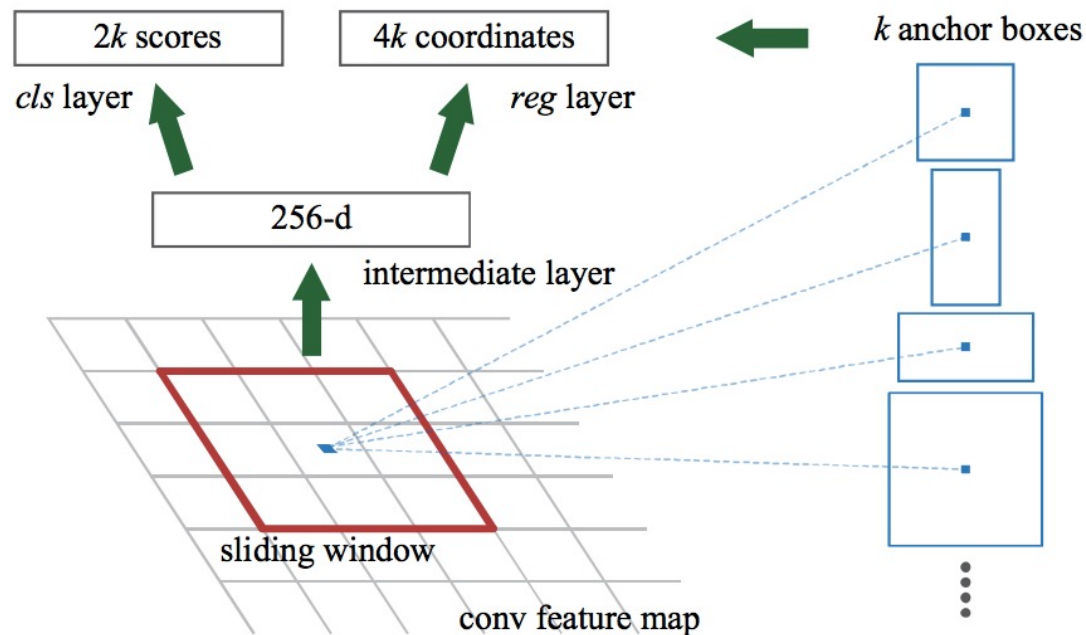
Region proposal network (RPN)

- Implementation: put conv layers over low-resolution feature grid, for each grid location predict “object/no object” scores and bounding box regression coordinates

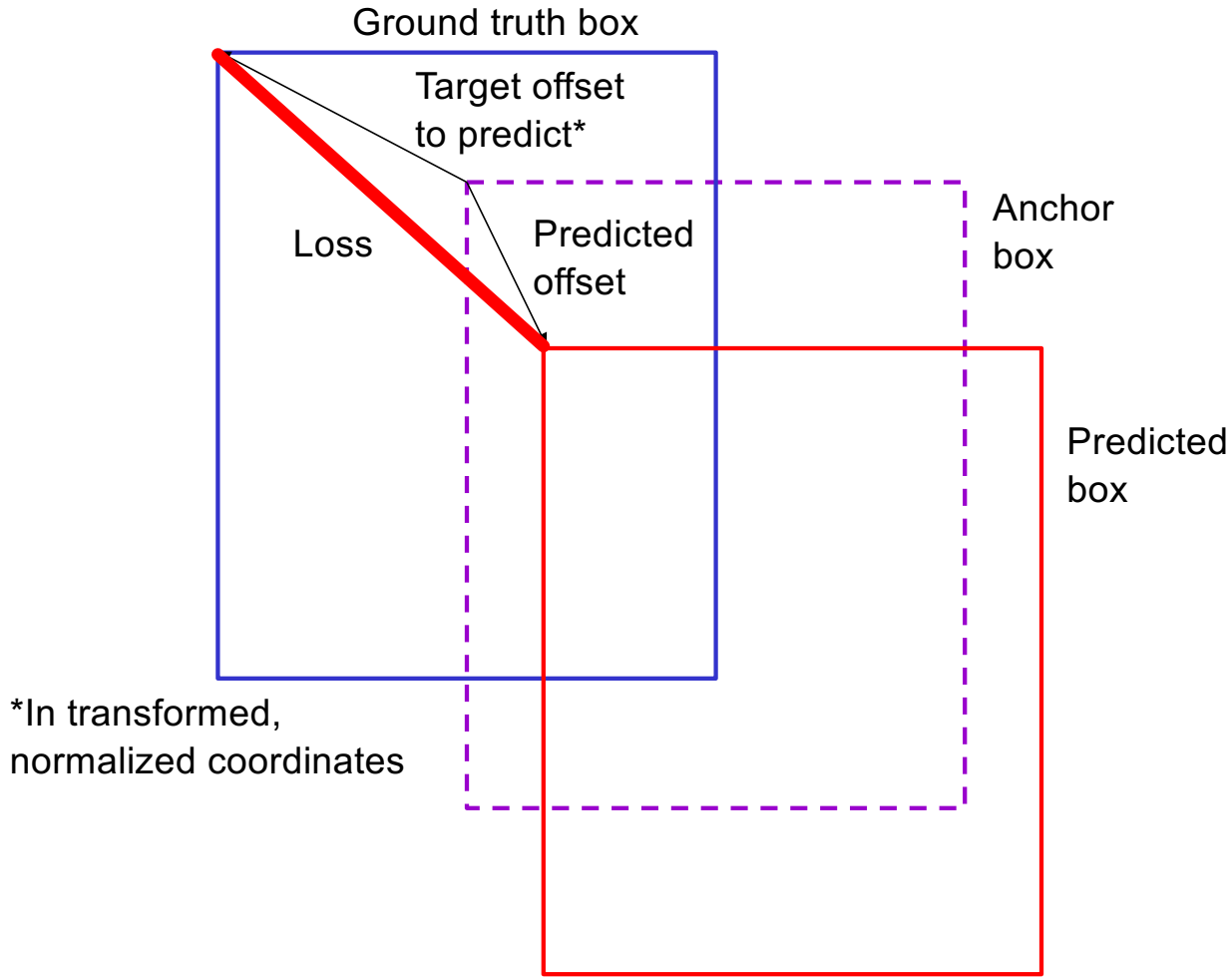


Region proposal network (RPN)

- Slide a small window (3x3) over the conv5 feature maps
 - Predict object/no object
 - Regress bounding box coordinates with reference to *anchors* (3 scales x 3 aspect ratios)



Regression relative to anchor box

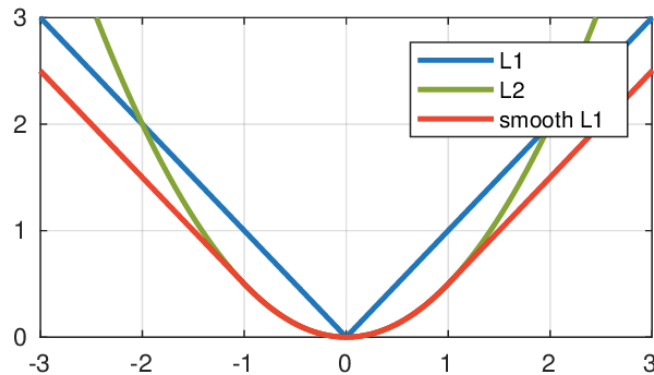


RPN loss

- For a set of anchors indexed by i , the RPN predicts BB coordinates $\{\hat{b}_i\}$ and object/no-object probabilities $\{\hat{p}_i\}$. The loss is given by

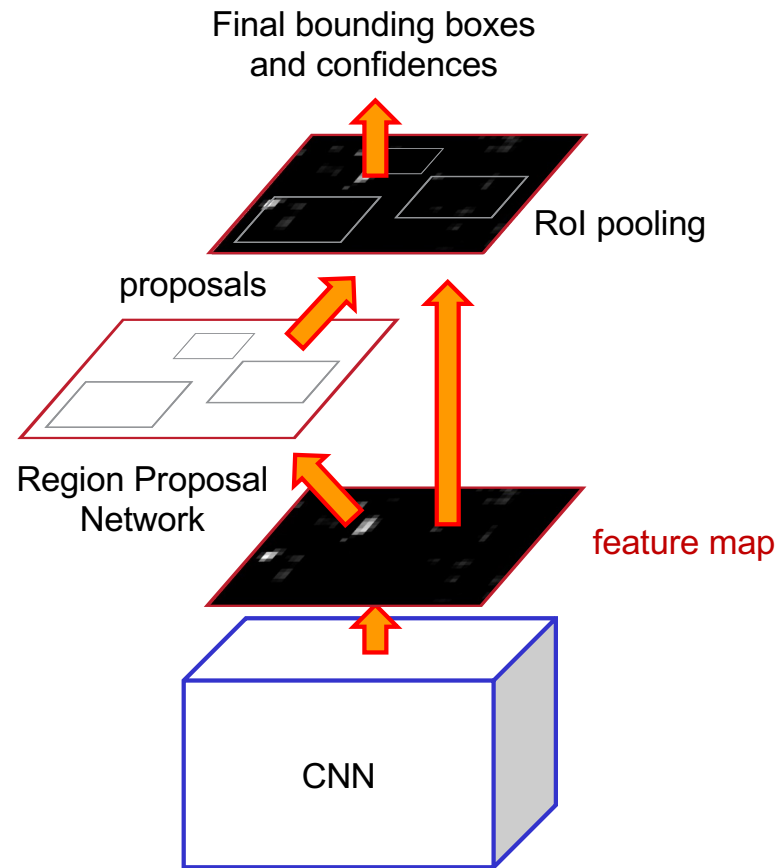
$$L(\{\hat{p}_i\}, \{\hat{b}_i\}) = \lambda_{\text{cls}} \sum_i \underbrace{L_{\text{cls}}(p_i, \hat{p}_i)}_{\text{logistic loss}} + \lambda_{\text{reg}} \sum_i \underbrace{L_{\text{reg}}(b_i, \hat{b}_i)}_{\text{regression loss}}$$

- Regression loss: *smooth* L_1 loss on top of offsets relative to the anchor (summed over BB coordinates)

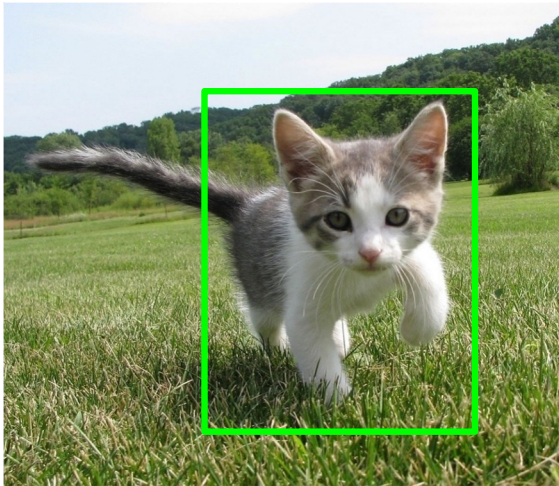


$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

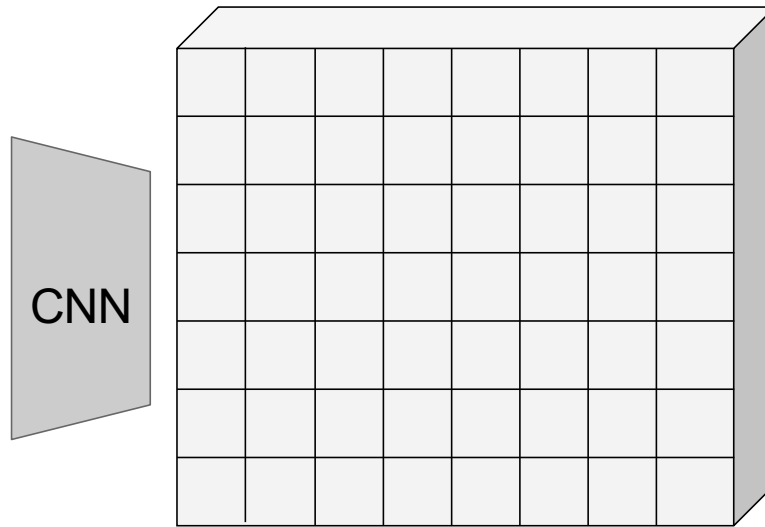
Rol pooling



Rol pooling



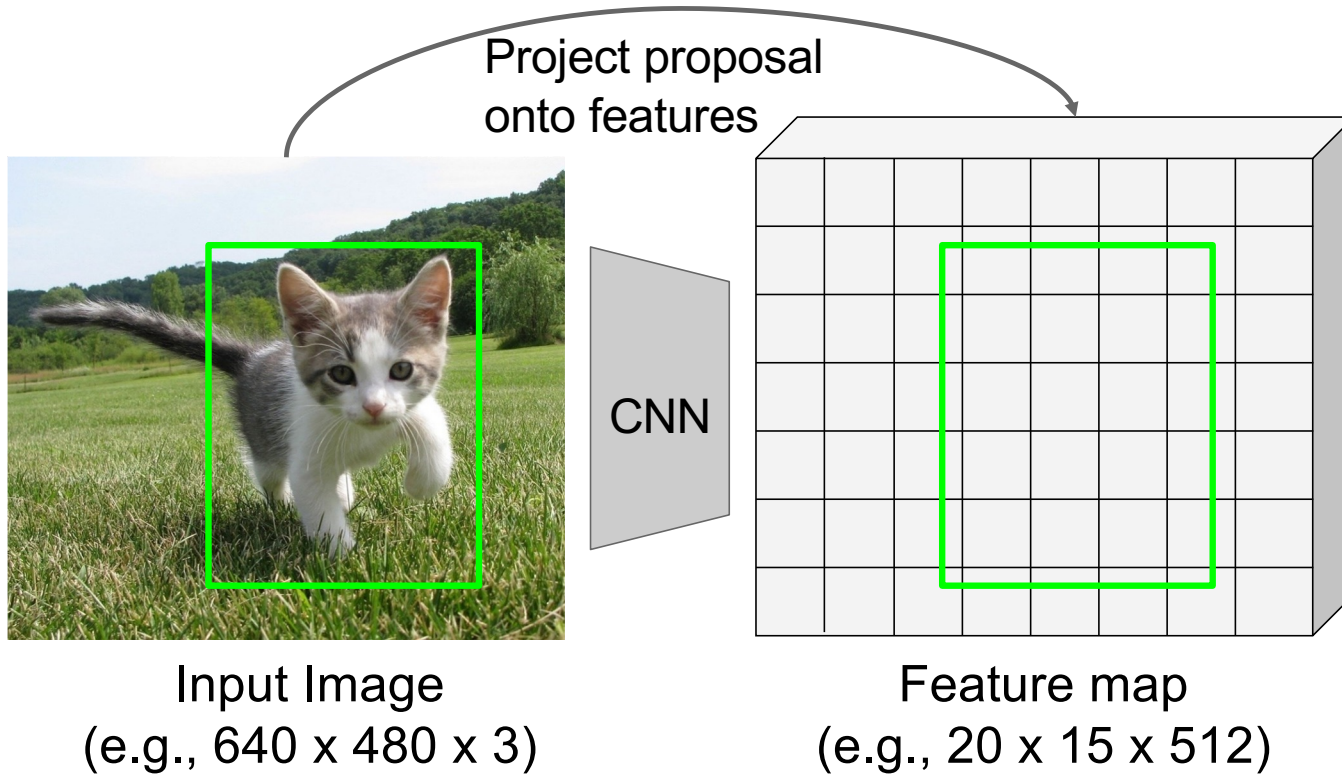
Input Image
(e.g., 640 x 480 x 3)



Feature map
(e.g., 20 x 15 x 512)

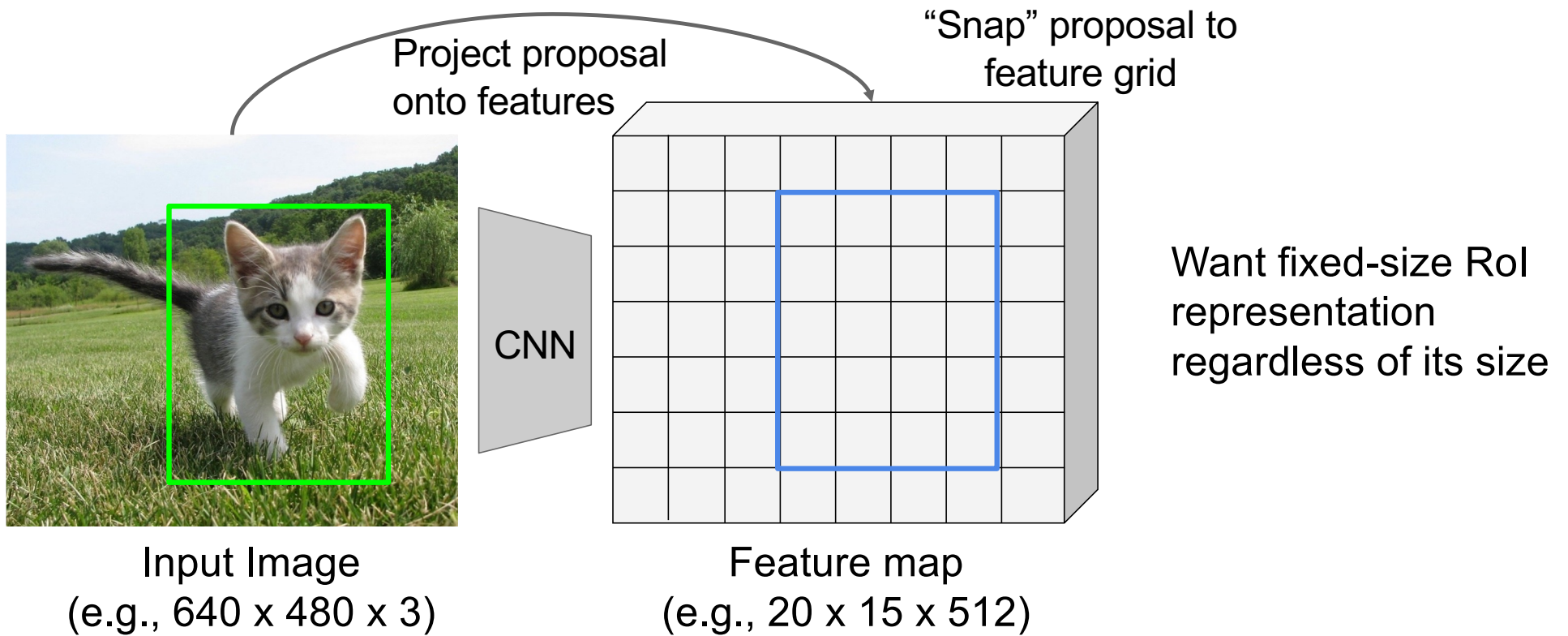
Source: [J. Johnson](#)

RoI pooling



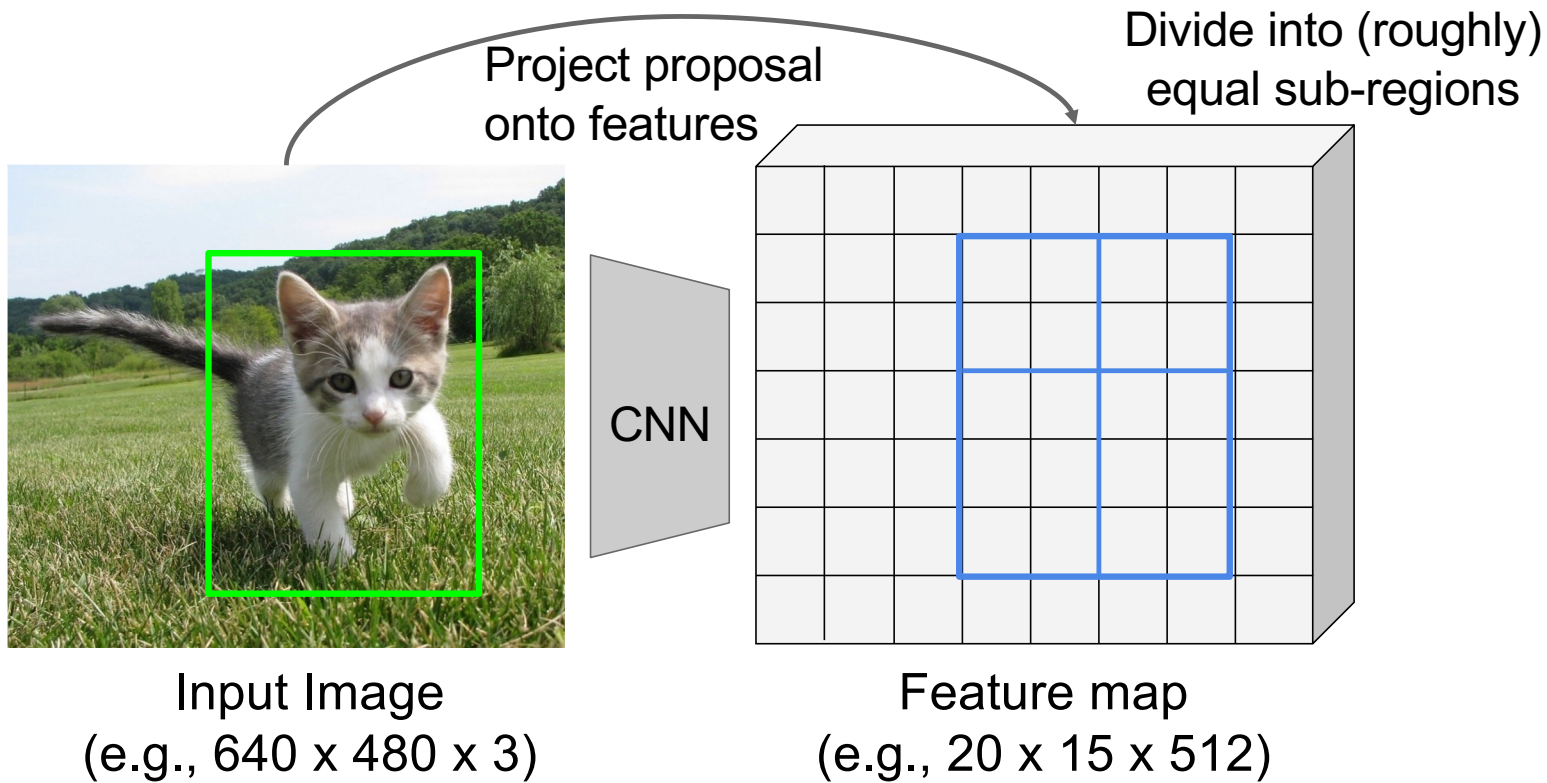
Source: [J. Johnson](#)

Rol pooling



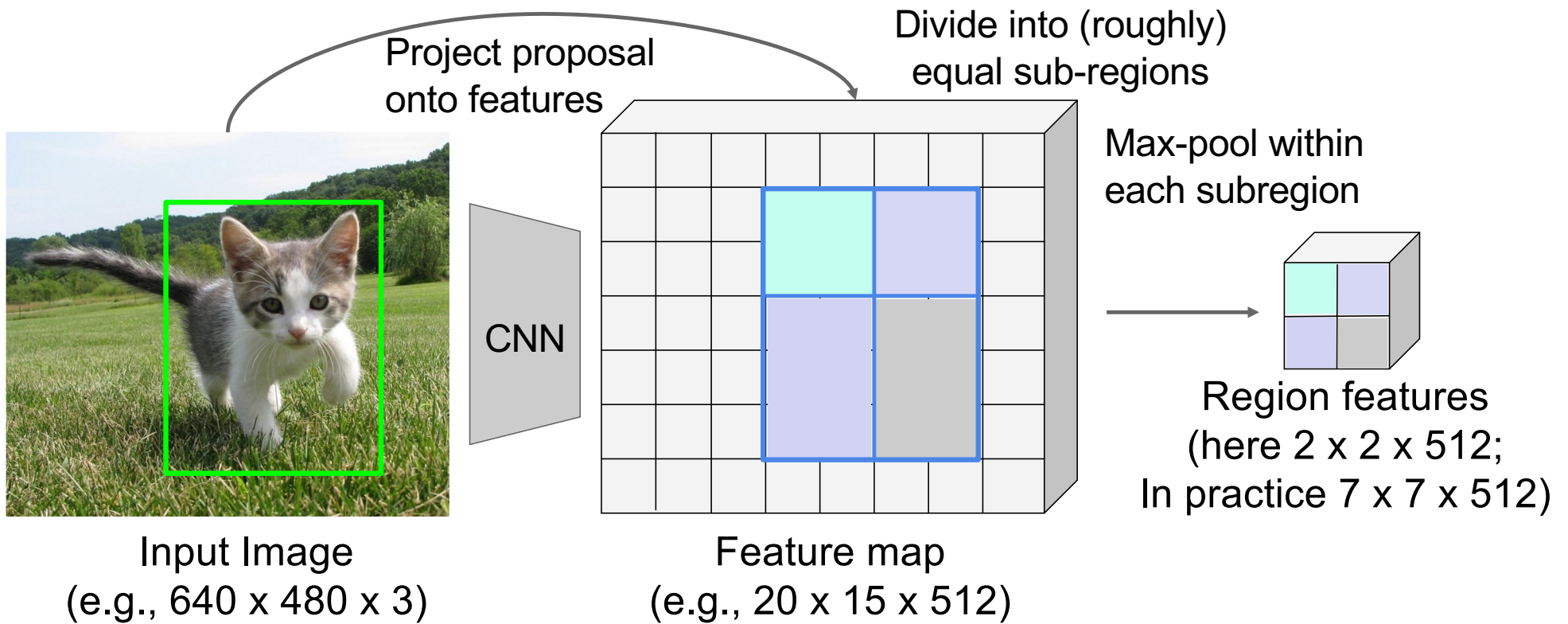
Source: [J. Johnson](#)

RoI pooling



Source: [J. Johnson](#)

RoI pooling



Source: [J. Johnson](#)

Rol pooling illustration

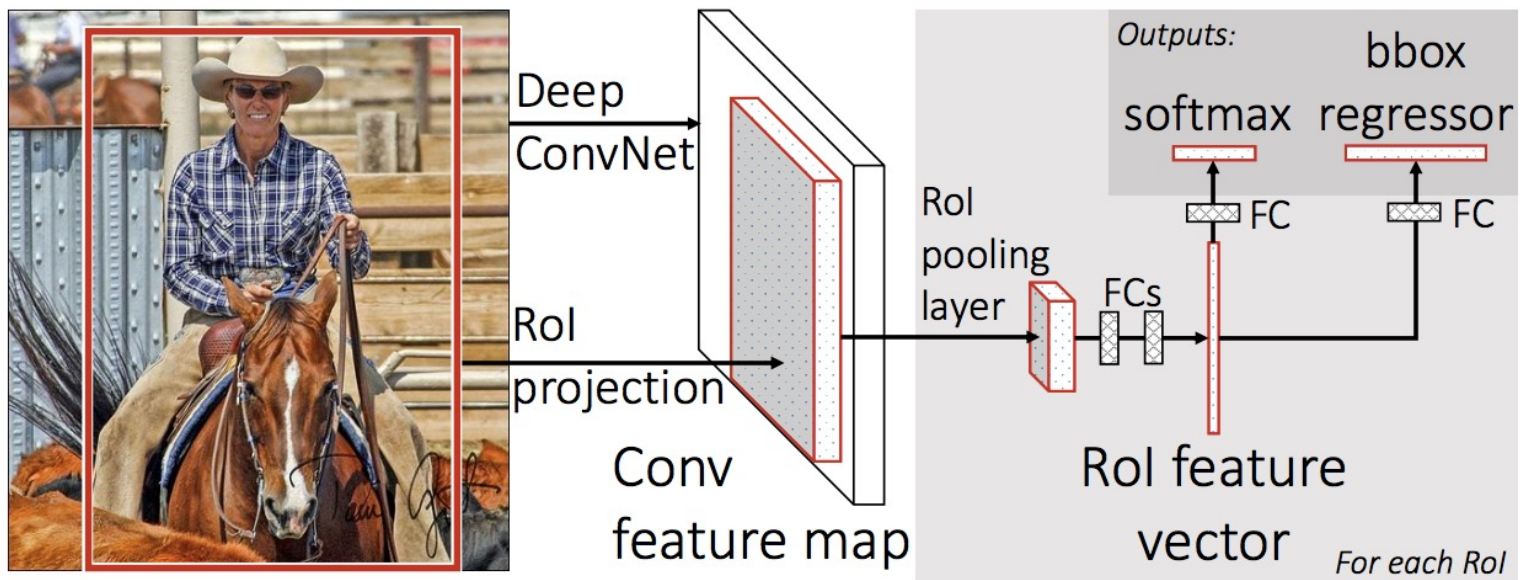
input

| | | | | | | | |
|------|------|------|------|------|------|------|------|
| 0.88 | 0.44 | 0.14 | 0.16 | 0.37 | 0.77 | 0.96 | 0.27 |
| 0.19 | 0.45 | 0.57 | 0.16 | 0.63 | 0.29 | 0.71 | 0.70 |
| 0.66 | 0.26 | 0.82 | 0.64 | 0.54 | 0.73 | 0.59 | 0.26 |
| 0.85 | 0.34 | 0.76 | 0.84 | 0.29 | 0.75 | 0.62 | 0.25 |
| 0.32 | 0.74 | 0.21 | 0.39 | 0.34 | 0.03 | 0.33 | 0.48 |
| 0.20 | 0.14 | 0.16 | 0.13 | 0.73 | 0.65 | 0.96 | 0.32 |
| 0.19 | 0.69 | 0.09 | 0.86 | 0.88 | 0.07 | 0.01 | 0.48 |
| 0.83 | 0.24 | 0.97 | 0.04 | 0.24 | 0.35 | 0.50 | 0.91 |

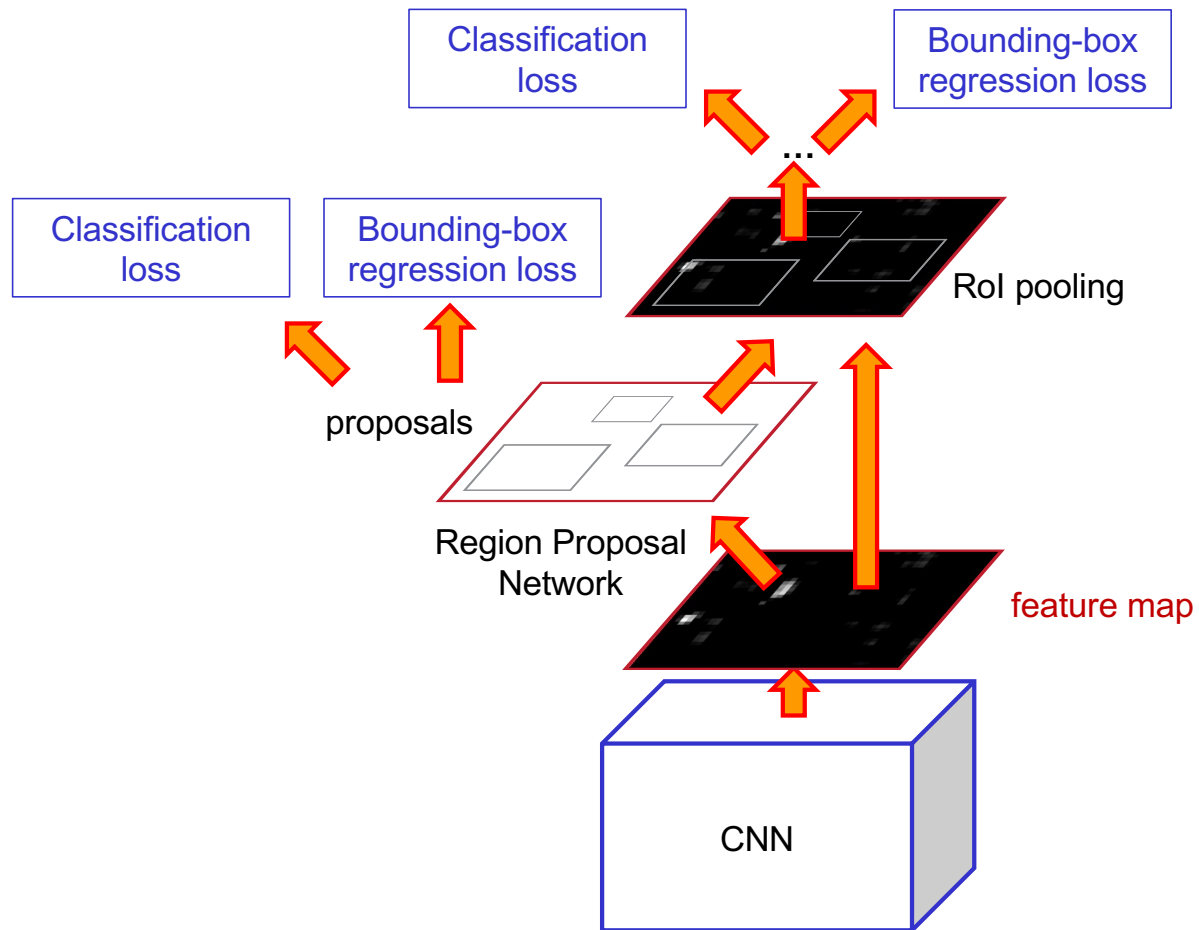
[Image source](#)

Final prediction stage

- For each RoI, predict probabilities for $C + 1$ classes (class 0 is background) and four bounding box offsets for C classes



Faster R-CNN losses and joint training



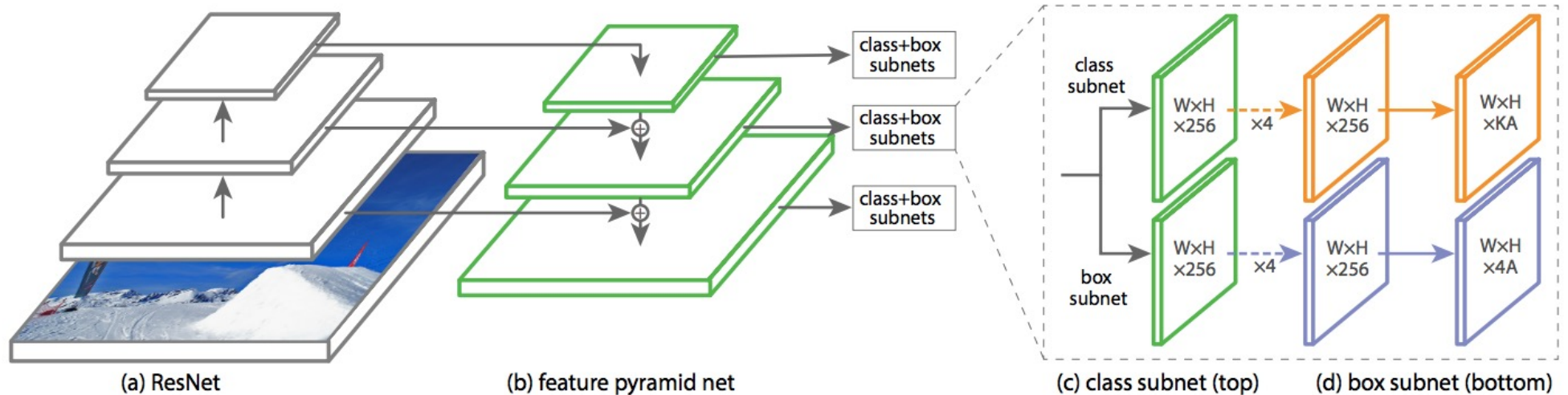
Results on PASCAL VOC

| system | time | 07 data | 07+12 data |
|--------------|--------------|-------------|-------------|
| R-CNN | ~50s | 66.0 | - |
| Fast R-CNN | ~2s | 66.9 | 70.0 |
| Faster R-CNN | 198ms | 69.9 | 73.2 |

detection mAP on PASCAL VOC 2007, with VGG-16 pre-trained on ImageNet

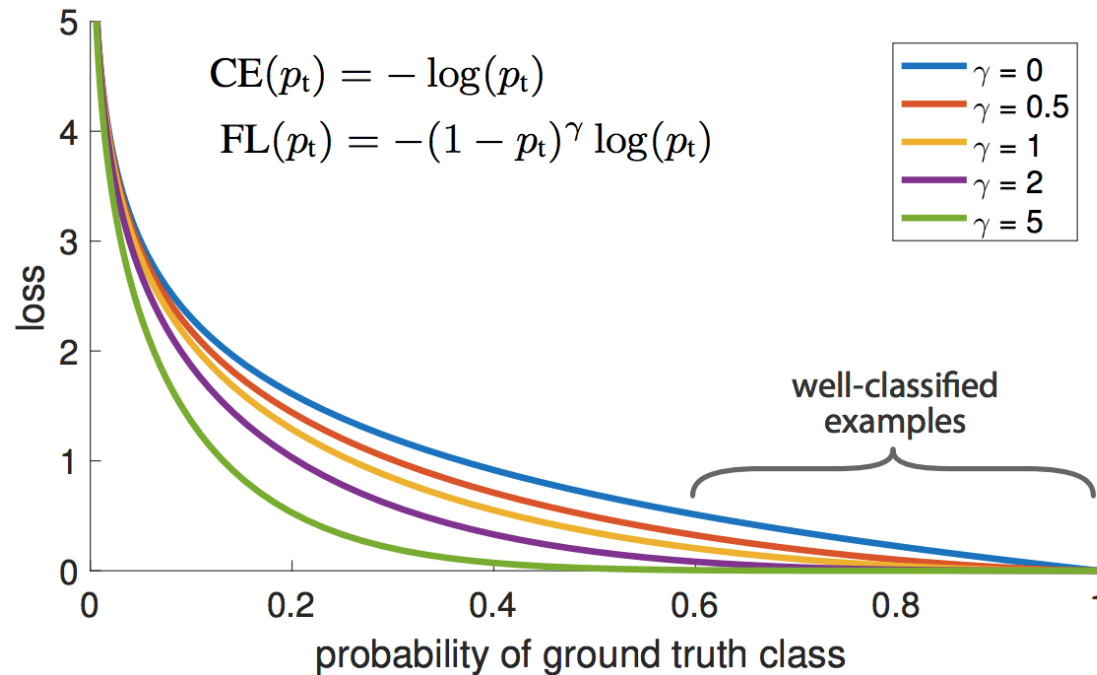
Multiscale detection: RetinaNet

- **Classification subnet:** predict the probability of object at each position for each of A anchors and K object classes
- **Box subnet:** for each position and each anchor, predict offset to ground truth box (if any)

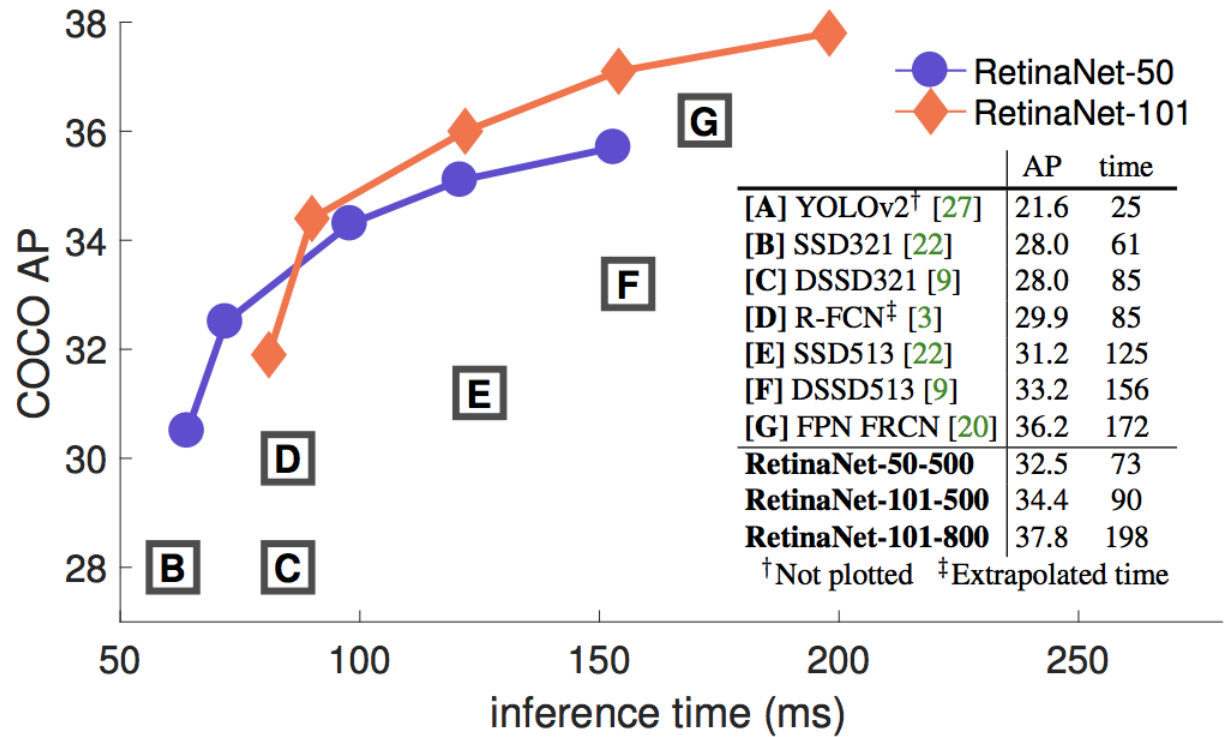


Multiscale detection: RetinaNet

- **Focal loss:** down-weight the standard cross-entropy loss for well-classified examples



RetinaNet: Results



T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, [Focal loss for dense object detection](#), ICCV 2017

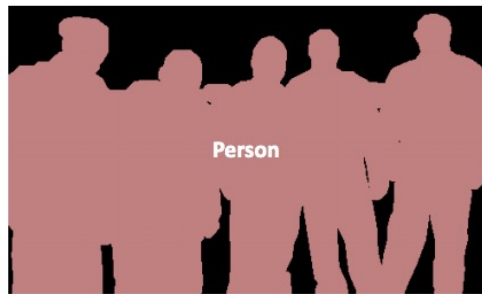
Outline

- Faster R-CNN
 - Region proposal network (RPN)
 - RoI pooling
- Mask R-CNN

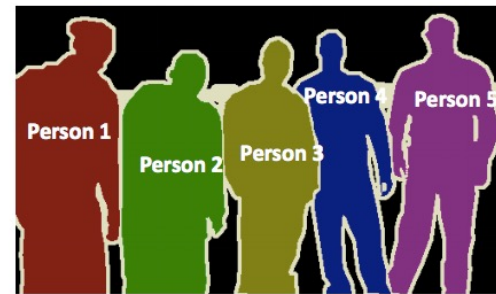
Instance segmentation



Object Detection



Semantic Segmentation



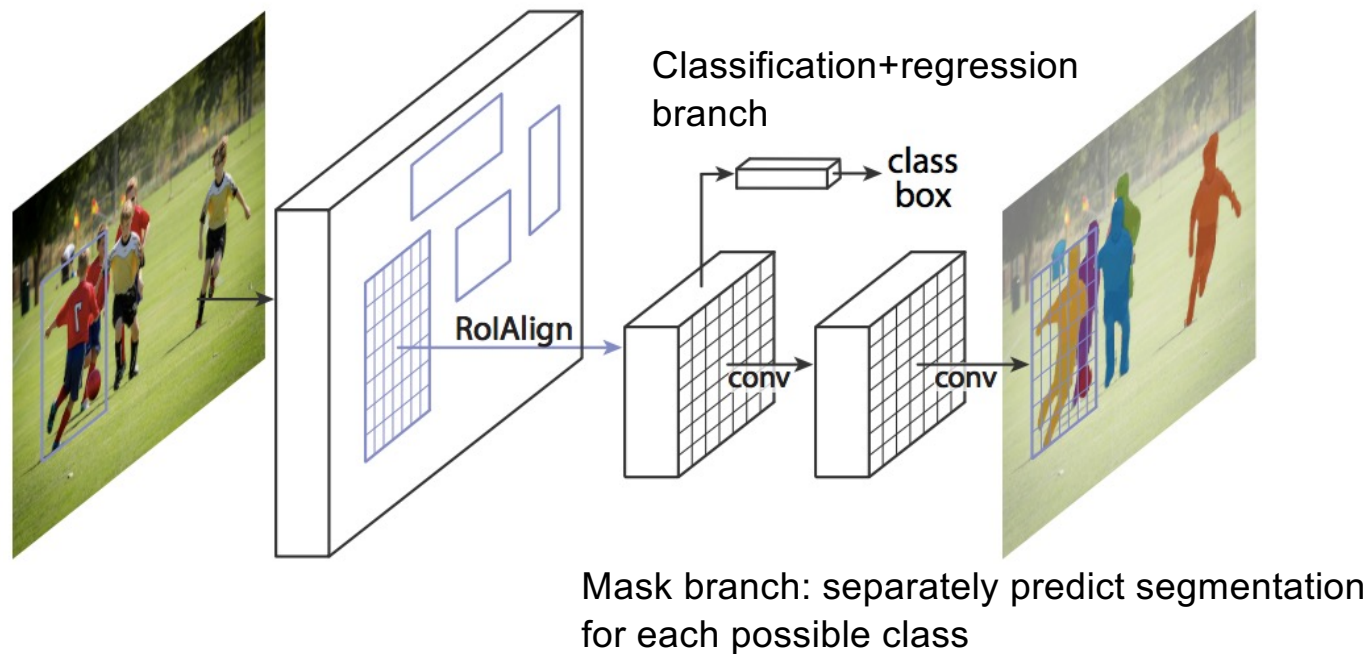
Instance Segmentation



K. He, G. Gkioxari, P. Dollar, and R. Girshick. [Mask R-CNN](#). ICCV 2017

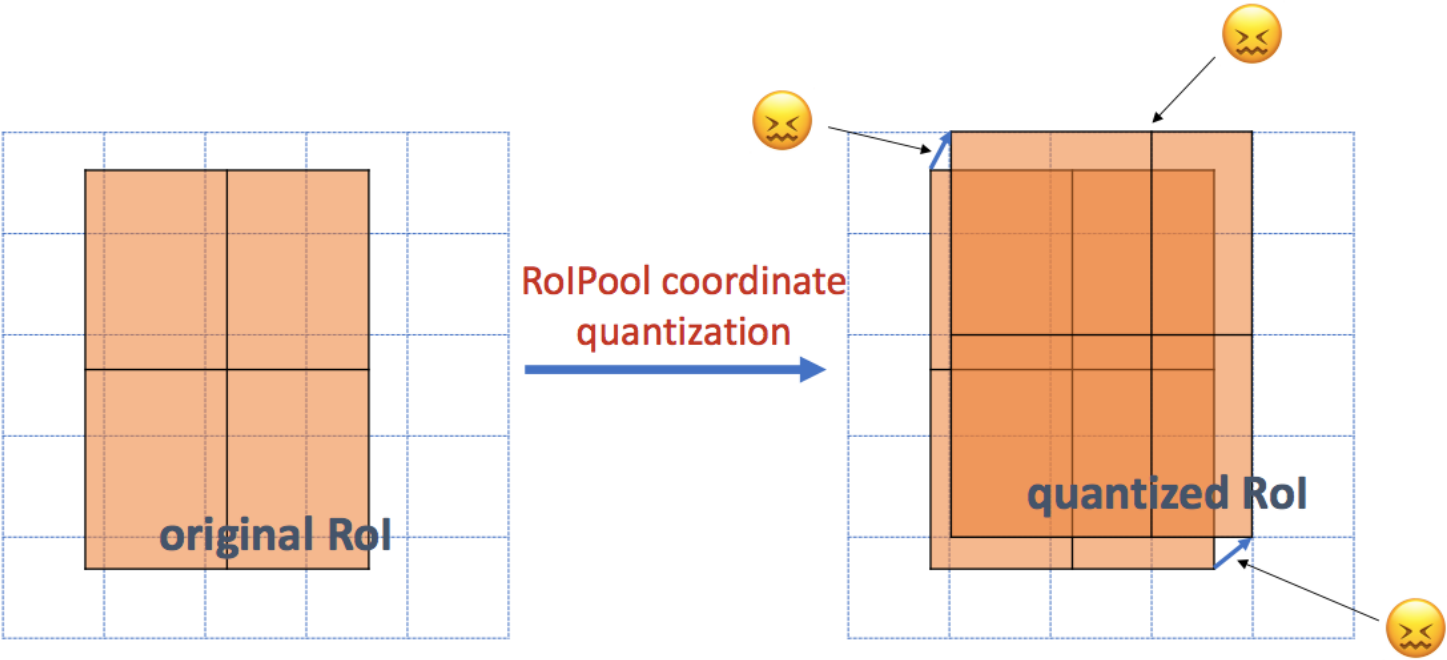
Mask R-CNN

- Mask R-CNN = Faster R-CNN + dense prediction on Rols



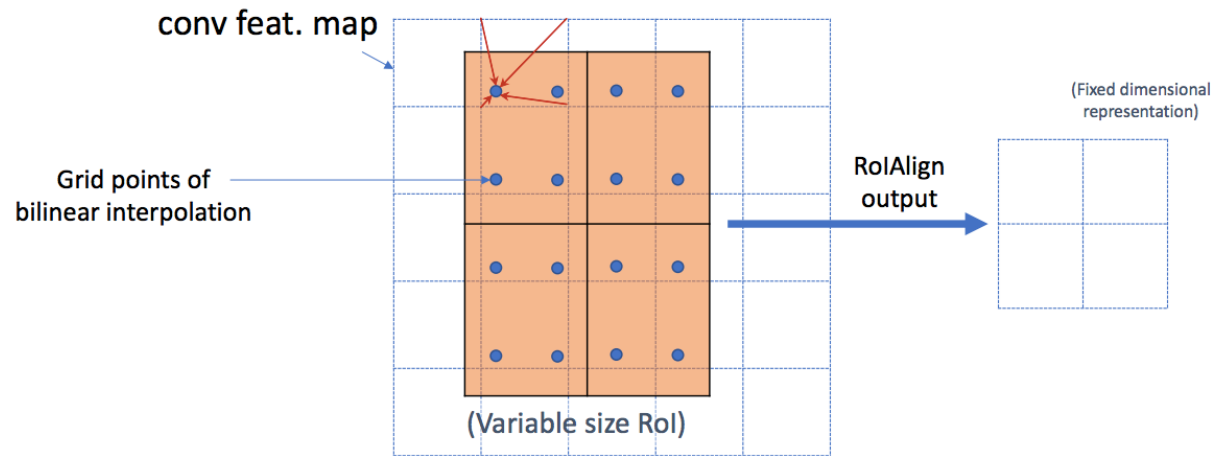
K. He, G. Gkioxari, P. Dollar, and R. Girshick. [Mask R-CNN](#). ICCV 2017

RoIPool: Nearest neighbor quantization

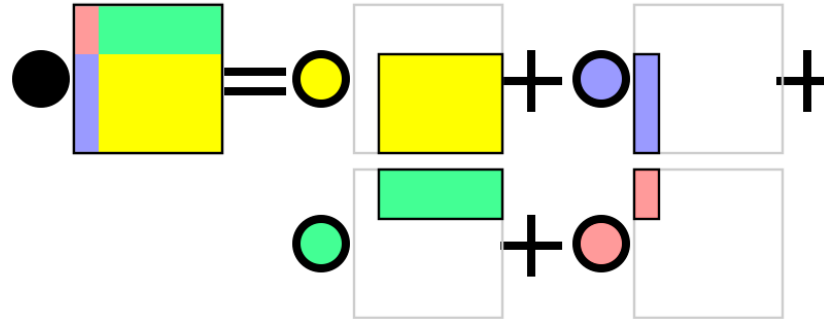
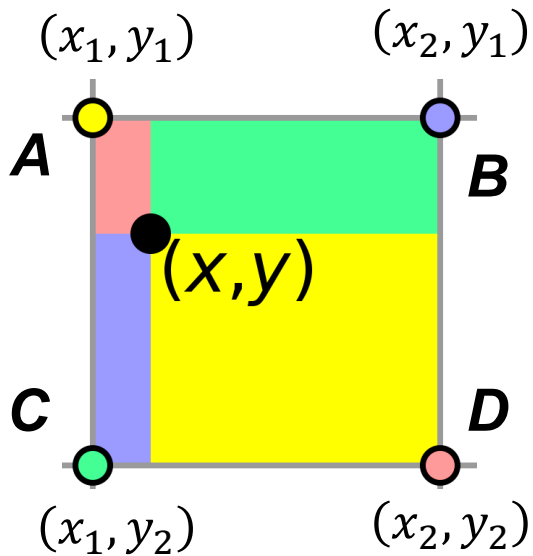


Source: K. He, R. Girshick

RoIAlign: Bilinear interpolation



Bilinear interpolation



$$f(x, y) = w_{11}A + w_{21}B + w_{12}C + w_{22}D$$

$$w_{11} = \frac{(x_2 - x)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)}$$

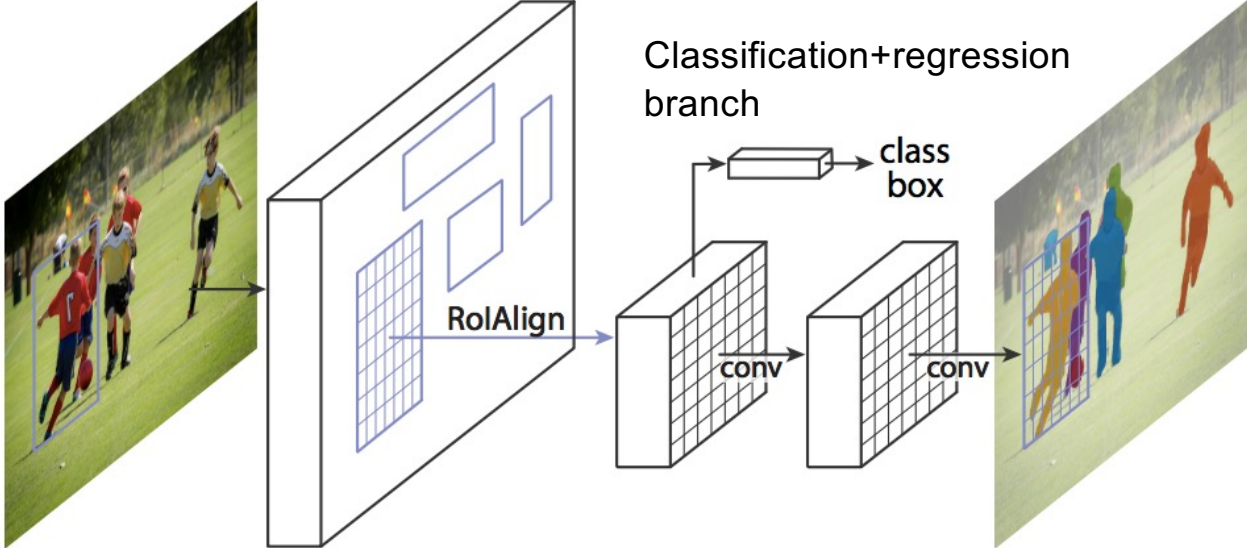
$$w_{12} = \frac{(x_2 - x)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)}$$

$$w_{21} = \frac{(x - x_1)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)}$$

$$w_{22} = \frac{(x - x_1)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)}$$

http://en.wikipedia.org/wiki/Bilinear_interpolation

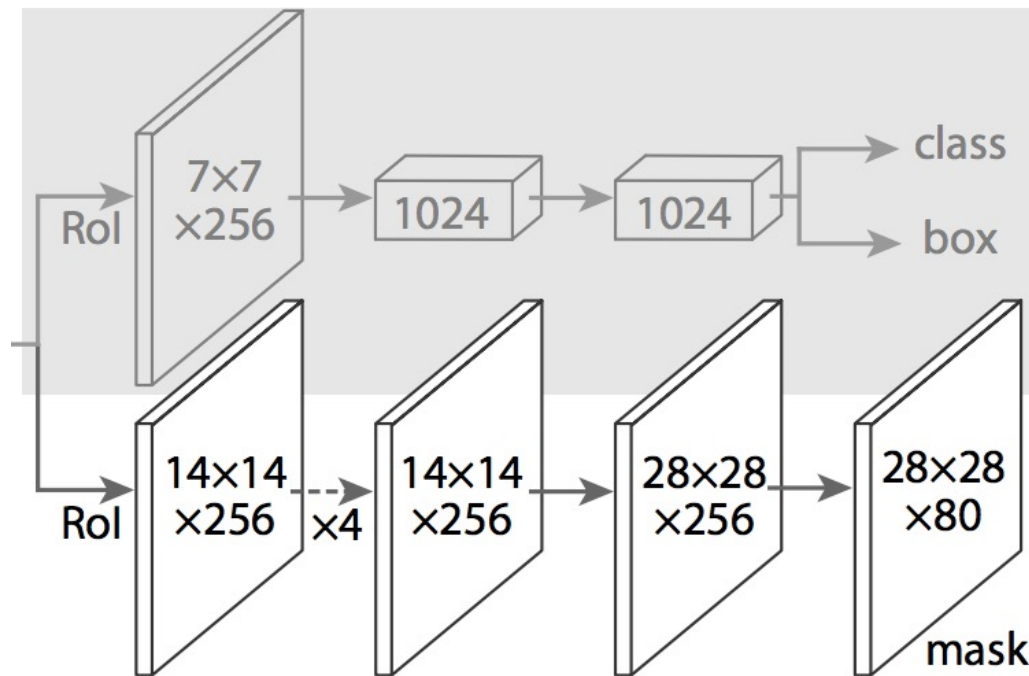
Mask R-CNN



Mask branch: separately predict segmentation for each possible class

Mask R-CNN

- From RoIAlign features, predict class label, bounding box, and segmentation mask



Classification/regression head from an established object detector

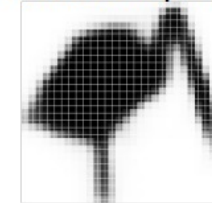
Separately predict binary mask for each class with per-pixel sigmoids, use average binary cross-entropy loss

Mask R-CNN



Validation image with box detection shown in red

28x28 soft prediction



Resized Soft prediction



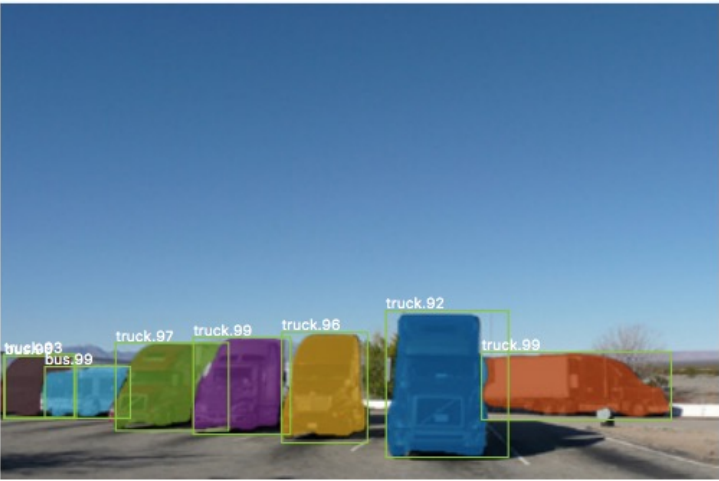
Final mask



Example results



Example results



Instance segmentation results on COCO

| | backbone | AP | AP ₅₀ | AP ₇₅ | AP _S | AP _M | AP _L |
|--------------------|-----------------------|-------------|------------------|------------------|-----------------|-----------------|-----------------|
| MNC [10] | ResNet-101-C4 | 24.6 | 44.3 | 24.8 | 4.7 | 25.9 | 43.6 |
| FCIS [26] +OHEM | ResNet-101-C5-dilated | 29.2 | 49.5 | - | 7.1 | 31.3 | 50.0 |
| FCIS+++ [26] +OHEM | ResNet-101-C5-dilated | 33.6 | 54.5 | - | - | - | - |
| Mask R-CNN | ResNet-101-C4 | 33.1 | 54.9 | 34.8 | 12.1 | 35.6 | 51.1 |
| Mask R-CNN | ResNet-101-FPN | 35.7 | 58.0 | 37.8 | 15.5 | 38.1 | 52.4 |
| Mask R-CNN | ResNeXt-101-FPN | 37.1 | 60.0 | 39.4 | 16.9 | 39.9 | 53.5 |

AP at different IoU
thresholds

AP for different
size instances

Keypoint prediction

- Given K keypoints, train model to predict K $m \times m$ one-hot maps with cross-entropy losses over m^2 outputs



Outline

- Faster R-CNN
 - Region proposal network (RPN)
 - RoI pooling
- Mask R-CNN
- Other detectors

Fully convolutional one-stage detector (FCOS)

- “Anchor-free” approach

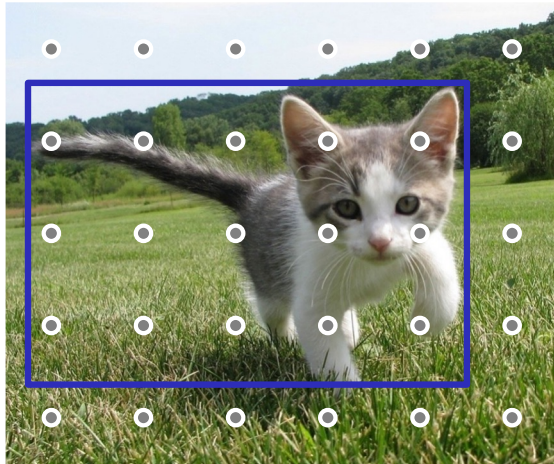


Figure source:
[J. Johnson](#)

Run backbone CNN to get
features aligned to input image

Fully convolutional one-stage detector (FCOS)

- “Anchor-free” approach

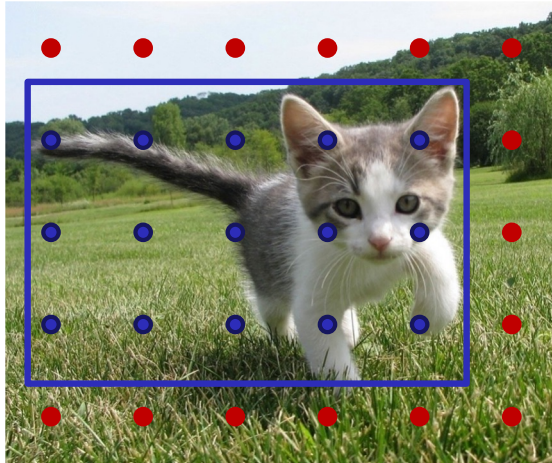


Figure source:
[J. Johnson](#)

For each class, predict whether location falls inside a GT bounding box

Fully convolutional one-stage detector (FCOS)

- “Anchor-free” approach

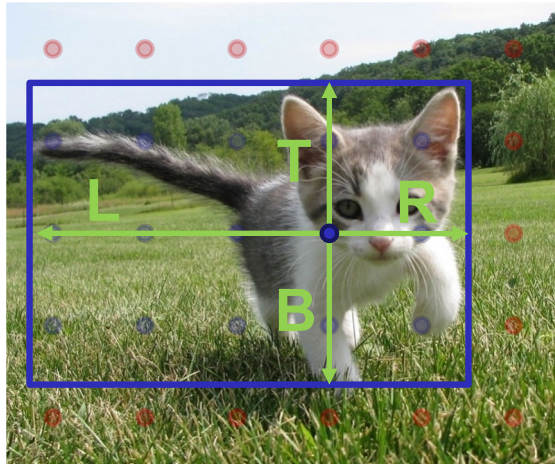


Figure source:
[J. Johnson](#)

For positive points, also regress distance to left, right, top, and bottom of GT box (with L2 loss)

Fully convolutional one-stage detector (FCOS)

- “Anchor-free” approach

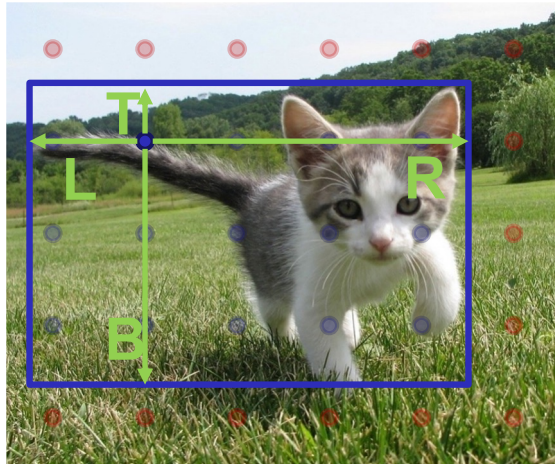
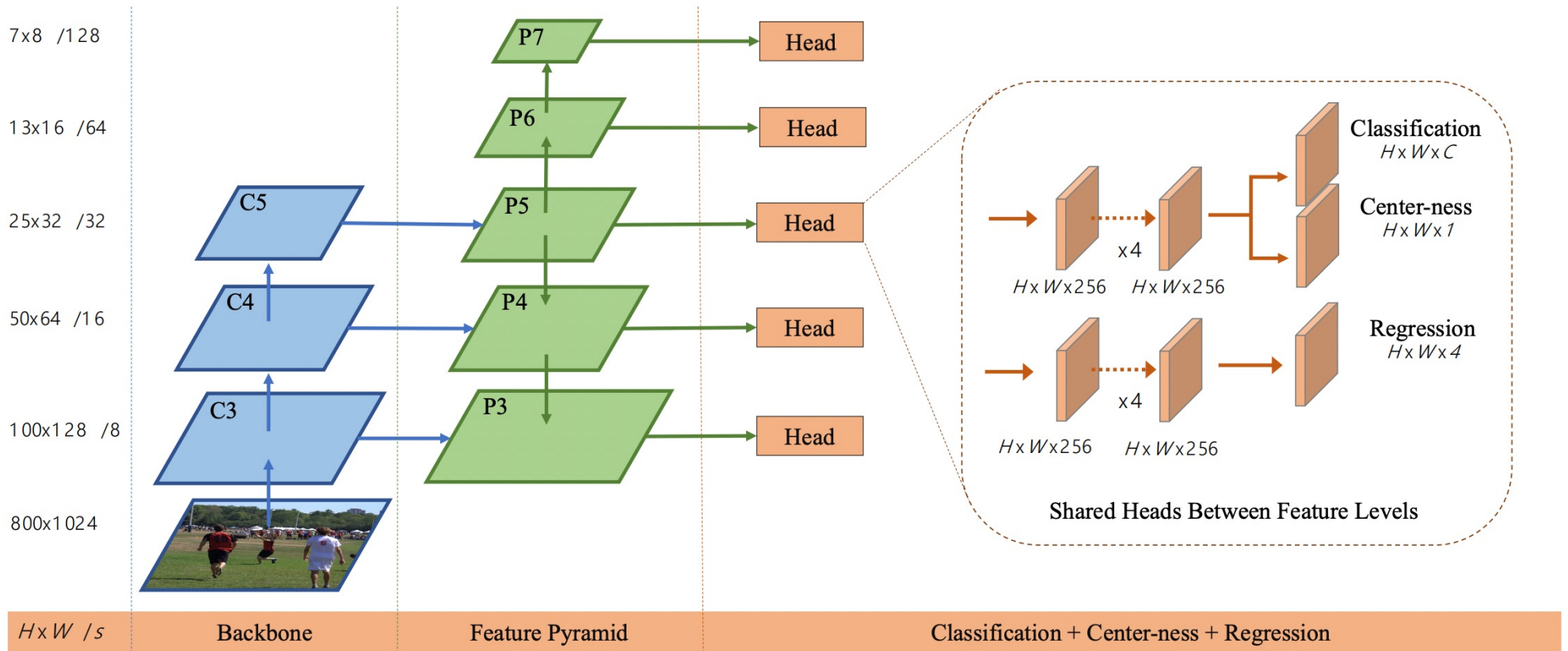


Figure source:
[J. Johnson](#)

For positive points, also regress distance to left, right, top, and bottom of GT box (with L2 loss)

Weight detections by “centerness” and confidence, perform NMS

Fully convolutional one-stage detector (FCOS)



Tian et al., [FCOS: Fully Convolutional One-Stage Object Detection](#), ICCV 2019

CornerNet

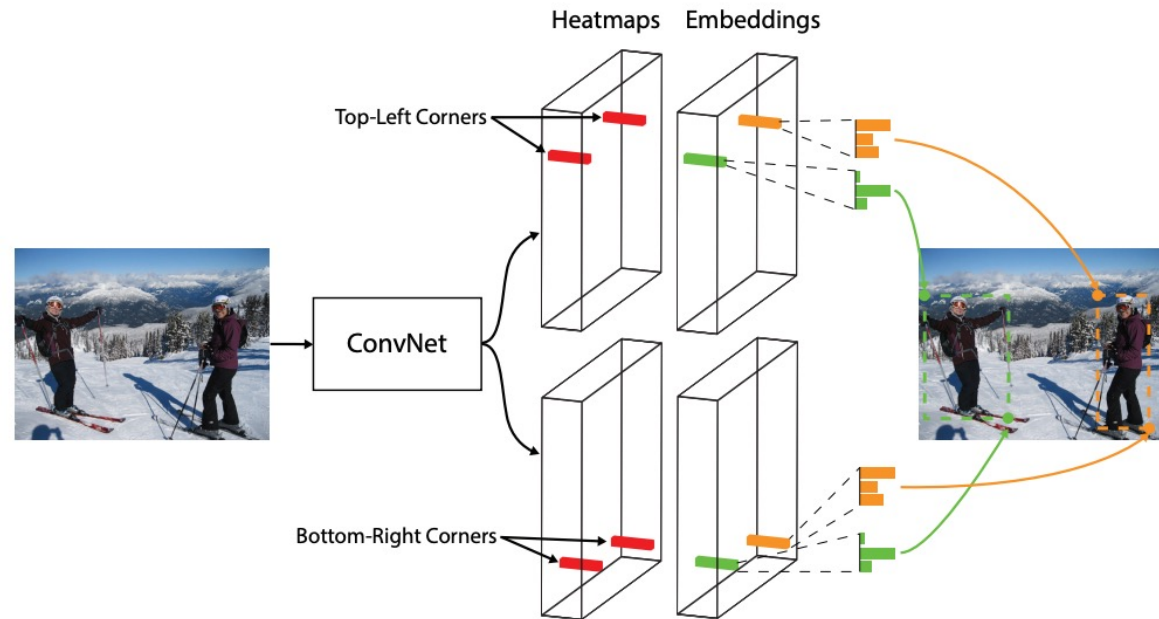


Fig. 1. We detect an object as a pair of bounding box corners grouped together. A convolutional network outputs a heatmap for all top-left corners, a heatmap for all bottom-right corners, and an embedding vector for each detected corner. The network is trained to predict similar embeddings for corners that belong to the same object.

CornerNet

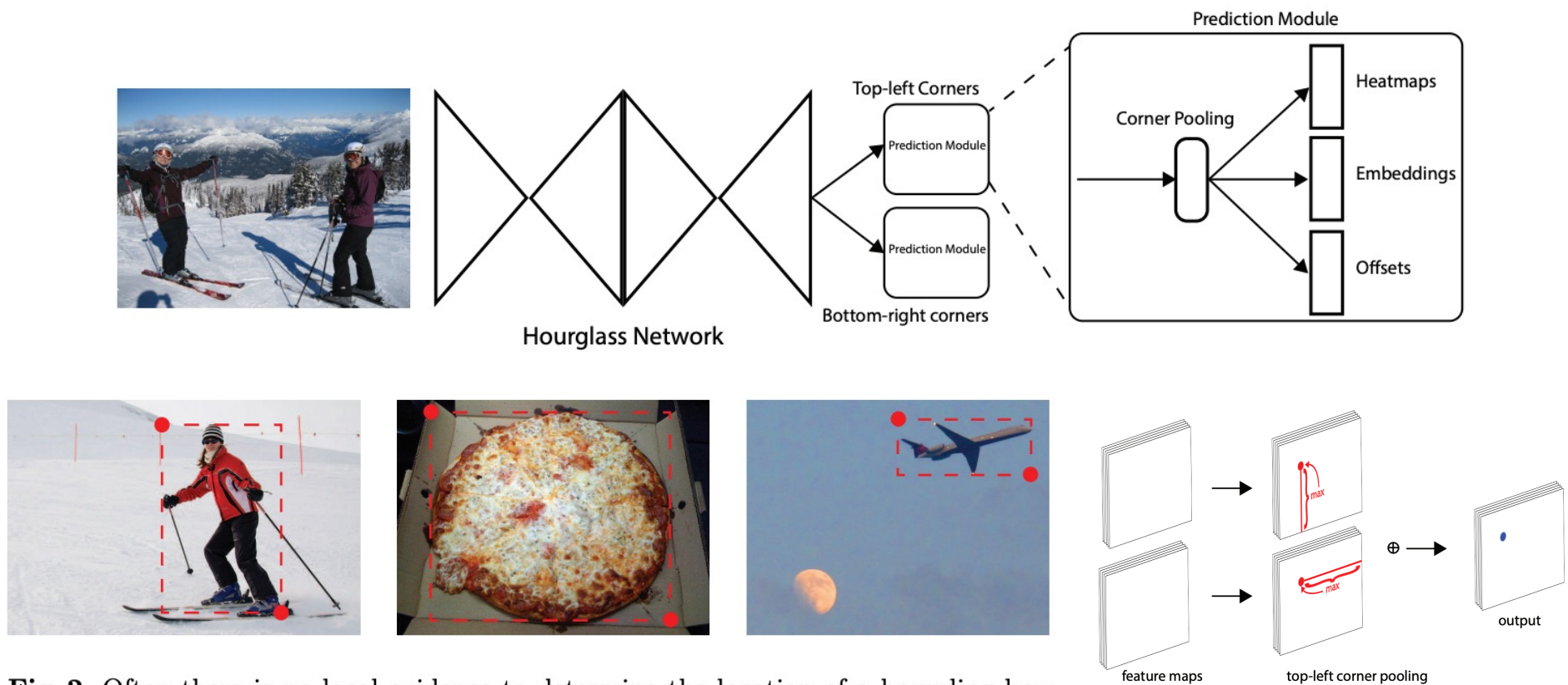
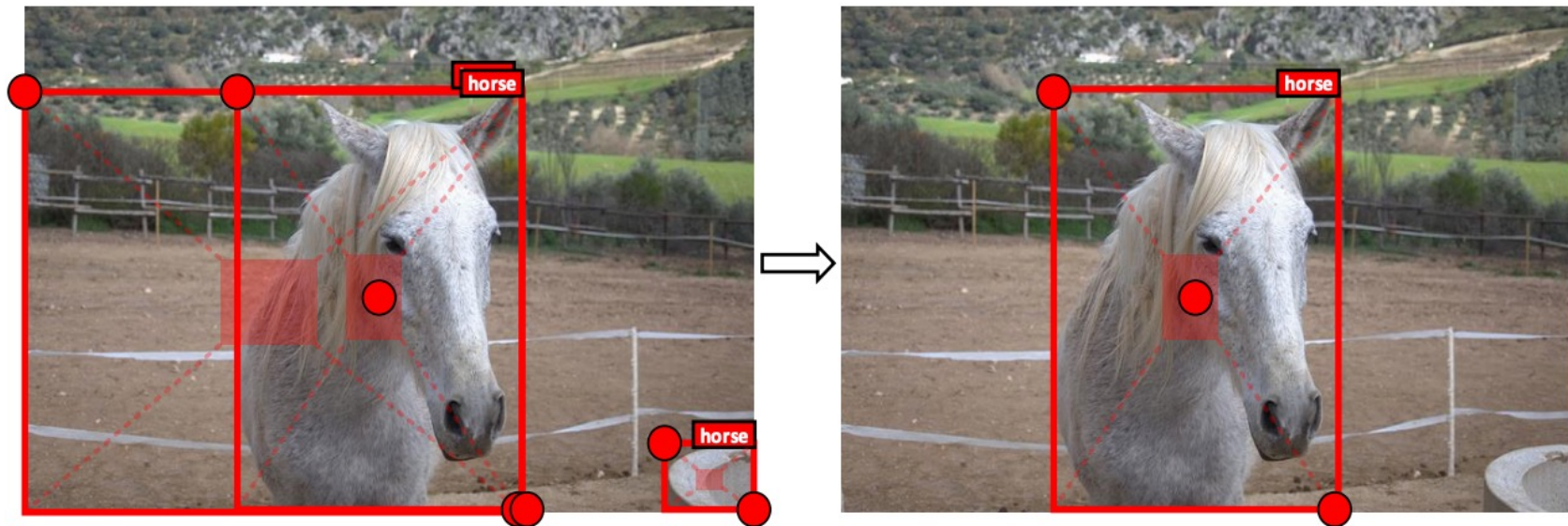


Fig. 2. Often there is no local evidence to determine the location of a bounding box corner. We address this issue by proposing a new type of pooling layer.

CenterNet

- Use an additional center point to verify predictions:



CenterNet

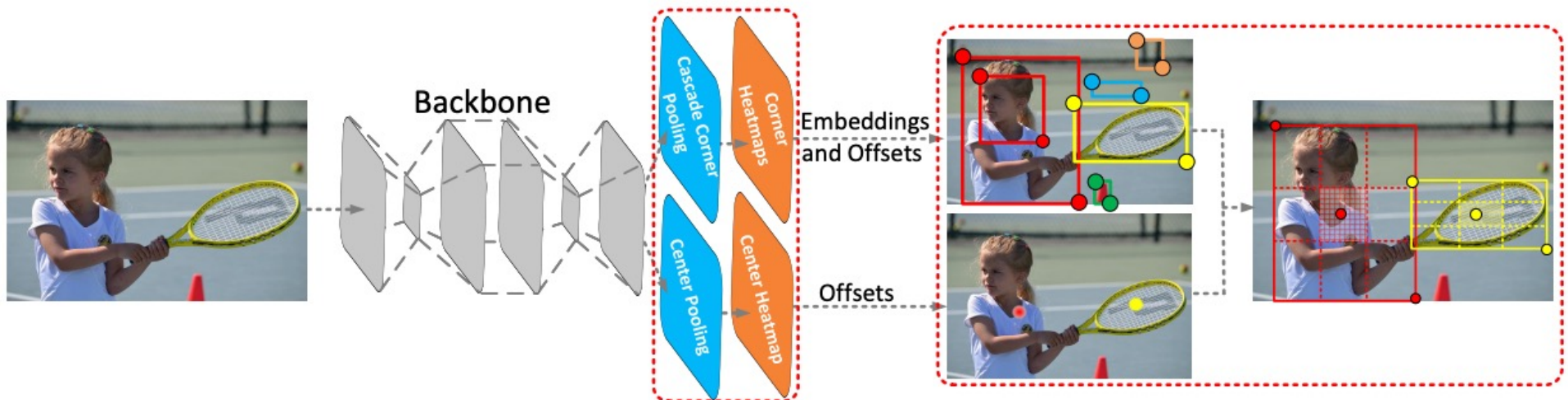


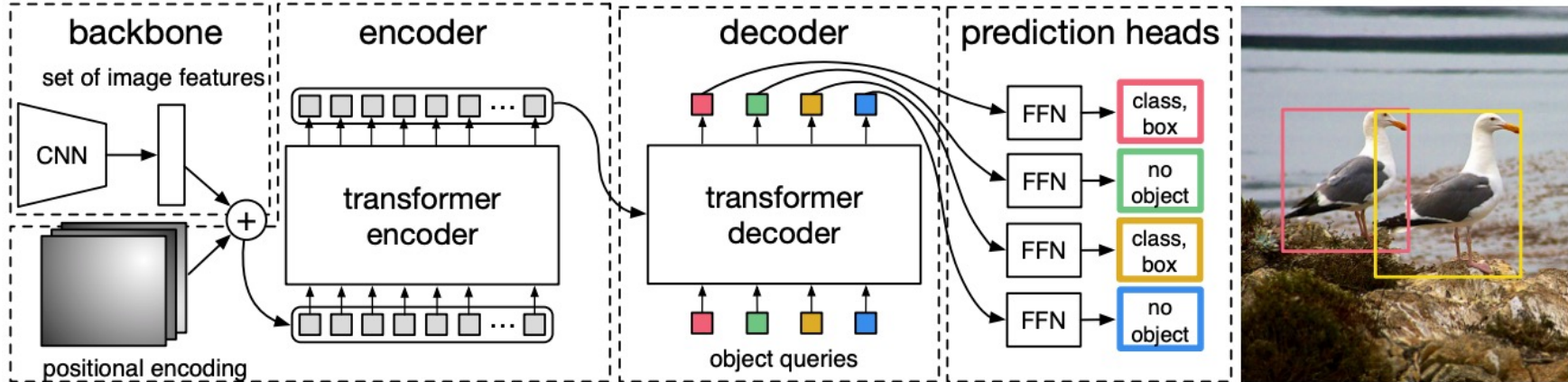
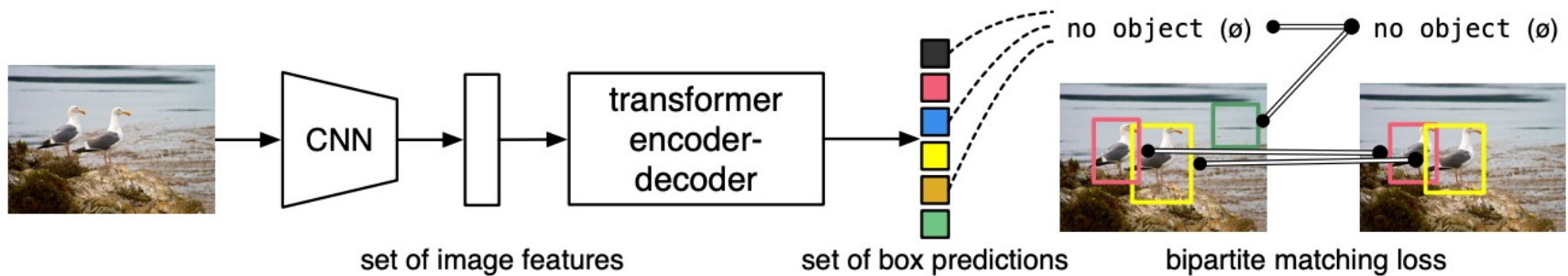
Figure 2: Architecture of CenterNet. A convolutional backbone network applies cascade corner pooling and center pooling to output two corner heatmaps and a center keypoint heatmap, respectively. Similar to CornerNet, a pair of detected corners and the similar embeddings are used to detect a potential bounding box. Then the detected center keypoints are used to determine the final bounding boxes.

CenterNet

| Method | FD | FD ₅ | FD ₂₅ | FD ₅₀ | FD _S | FD _M | FD _L |
|------------------|-------------|-----------------|------------------|------------------|-----------------|-----------------|-----------------|
| CornerNet511-52 | 40.4 | 35.2 | 39.4 | 46.7 | 62.5 | 36.9 | 28.0 |
| CenterNet511-52 | 35.1 | 30.7 | 34.2 | 40.8 | 53.0 | 31.3 | 24.4 |
| CornerNet511-104 | 37.8 | 32.7 | 36.8 | 43.8 | 60.3 | 33.2 | 25.1 |
| CenterNet511-104 | 32.4 | 28.2 | 31.6 | 37.5 | 50.7 | 27.1 | 23.0 |

Table 3: Comparison of the false discovery rates (%) of CornerNet and CenterNet on the MS-COCO validation dataset. The results suggest that CenterNet avoids a large number of incorrect bounding boxes, especially for small incorrect bounding boxes.

Detection Transformer (DETR)



N. Carion et al., [End-to-end object detection with transformers](#), ECCV 2020