Object detection

Vlad Shakhuro

1 November 2023

Outline

1. Task statement, datasets and metrics
2. Object detection via classification
3. R-CNN, Fast R-CNN, Faster R-CNN
4. YOLO
5. RetinaNet
6. Anchor-free detection

One-class object detection

Find all objects of a fixed class in an image. Output a set of bounding boxes:

$$\{(x_i, y_i, w_i, h_i)\}_{i=1}^{N}$$

One-class object detection

Find all objects of a fixed class in an image. Output a set of bounding boxes:

$$\{(x_i, y_i, w_i, h_i)\}_{i=1}^{N}$$

Instead of bboxes may also use:

- rotated bboxes
- ellipses
- pixel mask

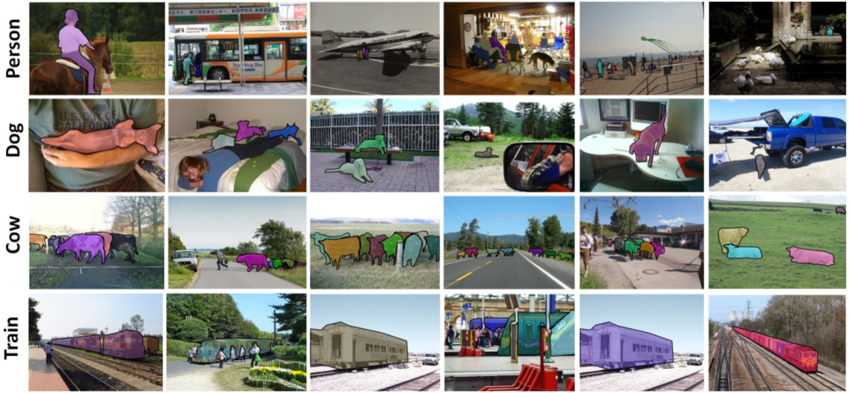
Multiclass object detection

Find all objects of a fixed set of classes in an image. Output a set of bounding boxes with classes:

$$\{(x_i, y_i, w_i, h_i, c_i)\}_{i=1}^{N}$$

N.B. we aim to find things (people, cars), not stuff (sky, road)

MS COCO dataset



200k images, 80 classes, 500k objects with masks

Lin et al. Microsoft COCO: Common Objects in Context. ECCV 2014

LVIS labelling for COCO

Additional fine labelling for COCO

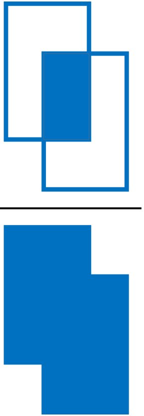
> 1000 object classes

2 M object masks

Gupta et al. LVIS: A Dataset for Large Vocabulary Instance Segmentation. CVPR 2019

IoU matching criterion

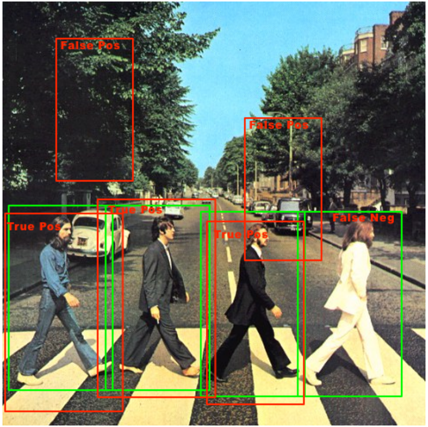
$$\text{Score} = \frac{\text{Area of overlap}}{\text{Area of union}}$$



Detection is correct if IoU > p (i.e. 0.5)

Computing precision and recall

Match predicted bboxes with ground truth bboxes using predicted confidences to compute TP, FP, FN

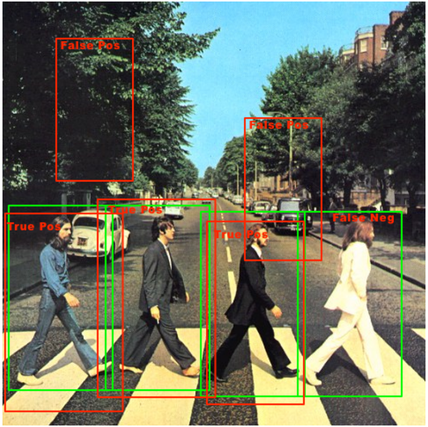


Computing precision and recall

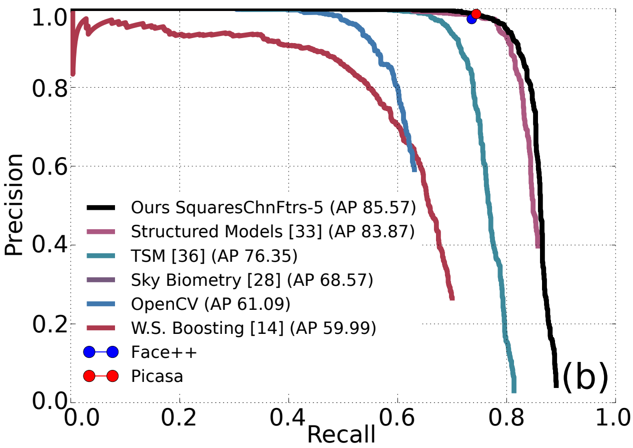
Match predicted bboxes with ground truth bboxes using predicted confidences to compute TP, FP, FN

$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{recall} = \frac{TP}{TP + FN}$$



Precision-Recall curve



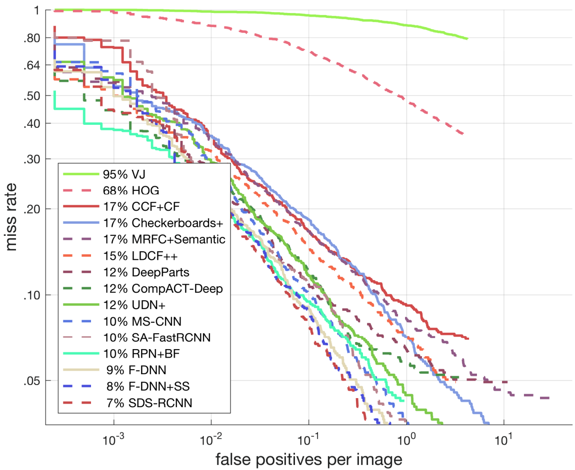
Precision-Recall values w.r.t. model hyperparameters

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} p(r)$$

mAP — averaged AP over all classes

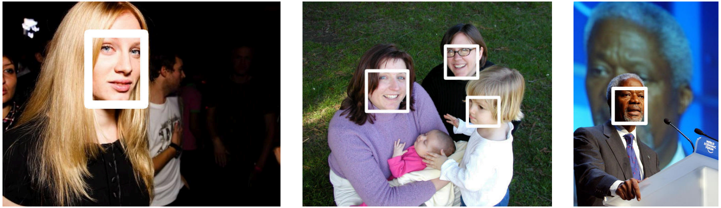
mAP may also be averaged over IoU thresholds

Miss rate vs FPPI

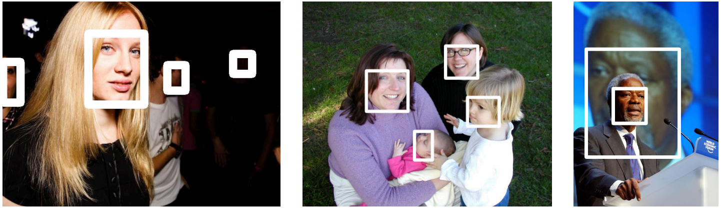


Caltech Pedestrian Detection Benchmark

Annotation protocol



(a) Original annotations



(b) Updated annotations

Mathias et al. Face detection without bells and whistles. ECCV 2014

Outline

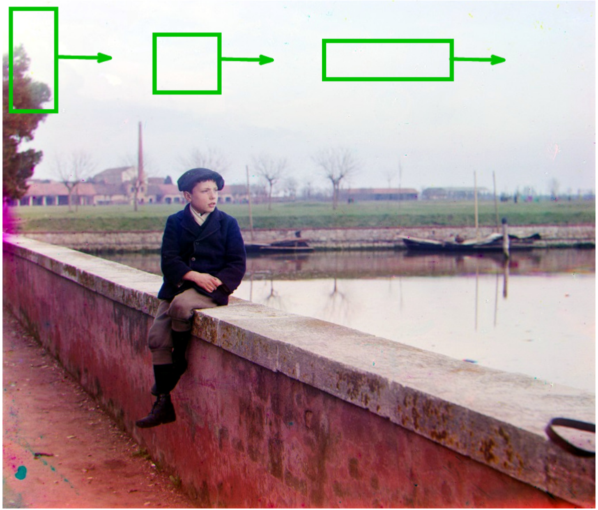
1. Task statement, datasets and metrics
2. Object detection via classification
3. R-CNN, Fast R-CNN, Faster R-CNN
4. YOLO
5. RetinaNet
6. Anchor-free detection

Sliding window

Multiscale sliding windows

Mutiresolution pyramid

Mutiple aspect ratios



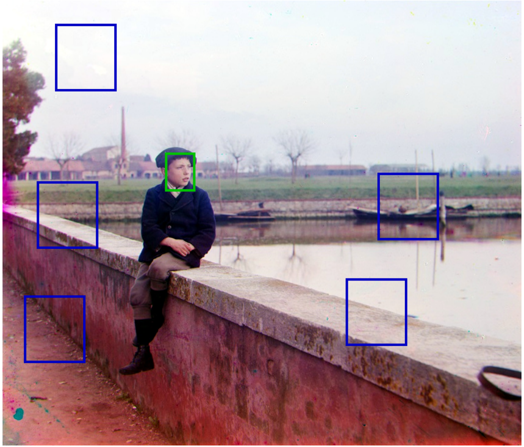
Non-maximum suppression (NMS)

Loop:

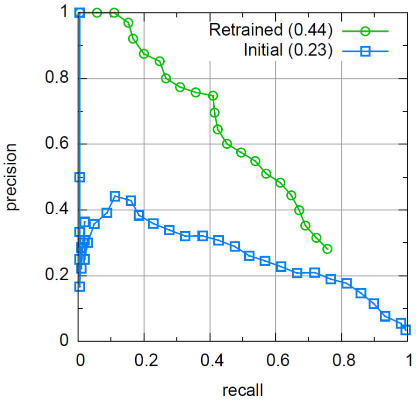
choose window with max confidence

remove all windows that instersect with chosen window

Imbalanced classes



Hard negative mining

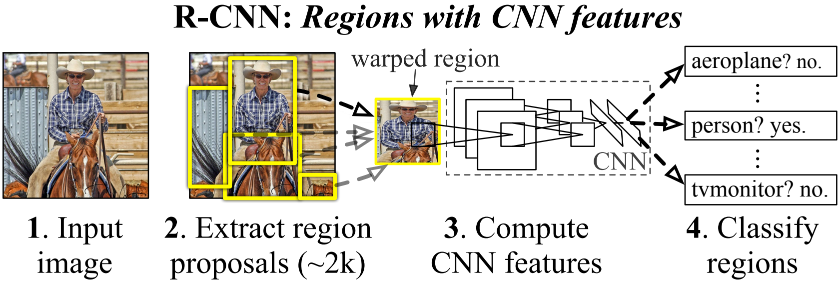


1. Choose random background samples
2. Train classifier
3. Loop:
   3.1 Evaluate detector on train images
   3.2 Choose hard false positive samples, add to classifier training sample
   3.3 Retrain classifier

Outline

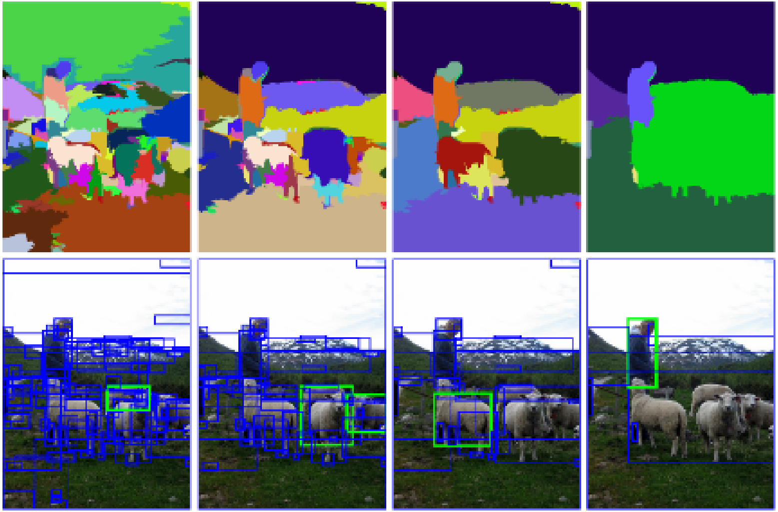
1. Task statement, datasets and metrics
2. Object detection via classification
3. R-CNN, Fast R-CNN, Faster R-CNN
4. YOLO
5. RetinaNet
6. Anchor-free detection

R-CNN



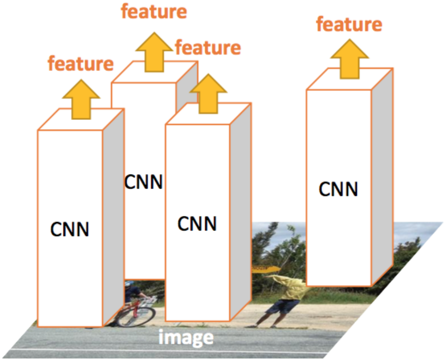
Girshick et al. Rich feature hierarchies for accurate object detection and semantic segmentation. CVPR 2014

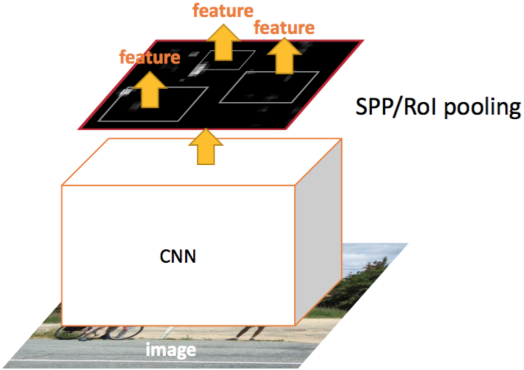
Selective Search



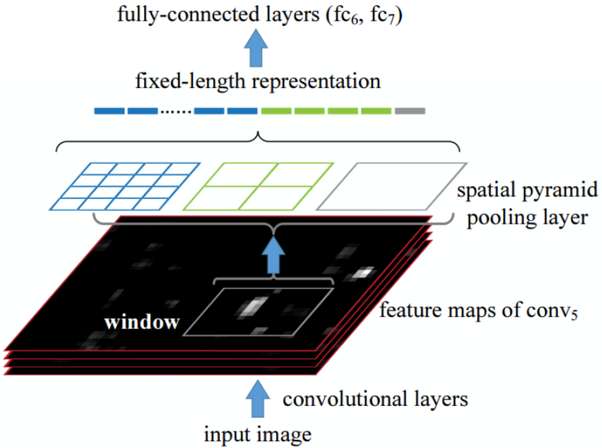
Uijlings et al. Selective Search for Object Recognition. IJCV 2013

Fast R-CNN


feature
feature
feature
feature
CNN
CNN
CNN
CNN
image

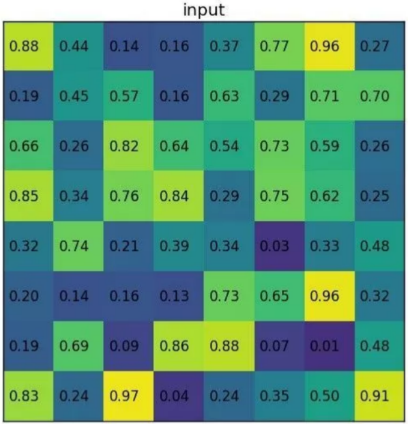

feature
feature
feature
SPP/RoI pooling
CNN
image

Spatial Pyramid Pooling

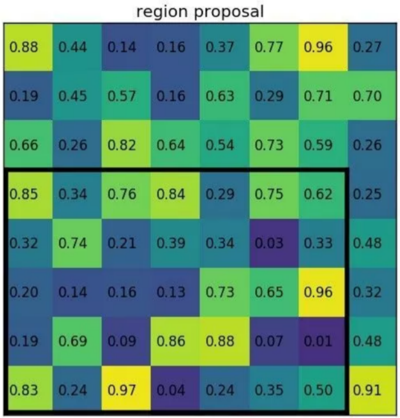


He et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. TPAMI 2015

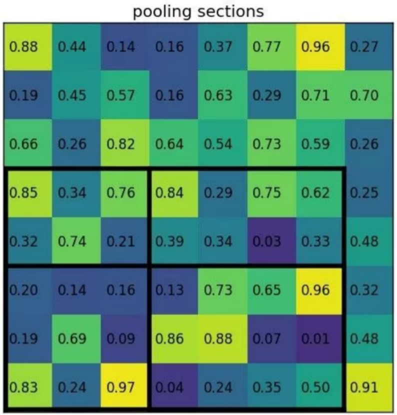
RoI pooling


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

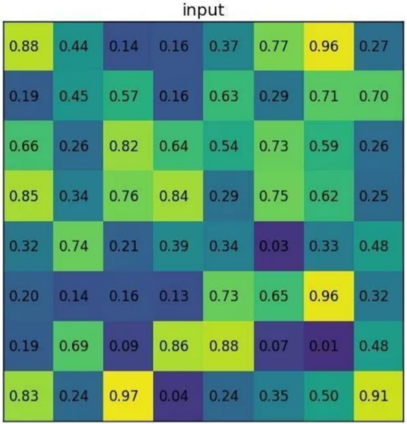
RoI pooling


region proposal
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

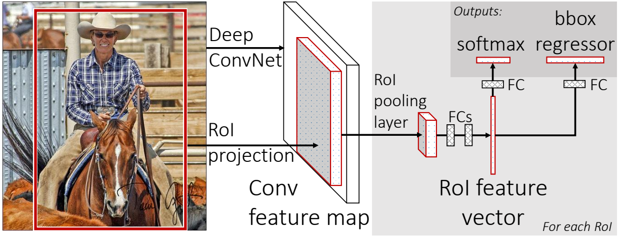
RoI pooling


pooling sections
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

RoI align


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

Fast R-CNN architecture



Key ideas:

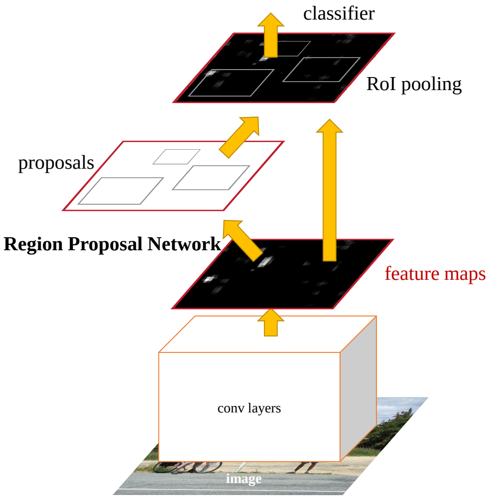
- compute CNN features over whole image
- use RoI pooling to compute features for region
- train a neural network on top of features for bbox classification and regression

R. Girshick. Fast R-CNN. ICCV 2015

R-CNN and Fast R-CNN comparison

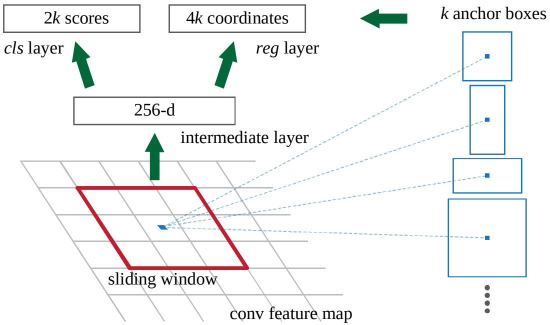
	R-CNN	Fast R-CNN
Training time	84h	8.75h
Testing per image	47s	0.32s
+ selective search	49s	2.32s
Test mAP	66.0%	68.1%

Faster R-CNN



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

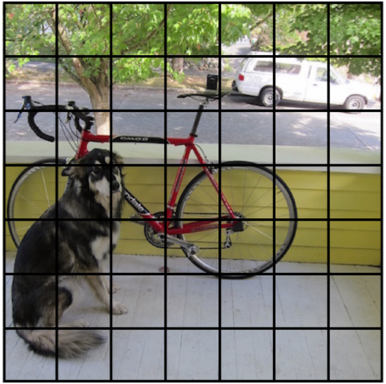
Region Proposal Network



Outline

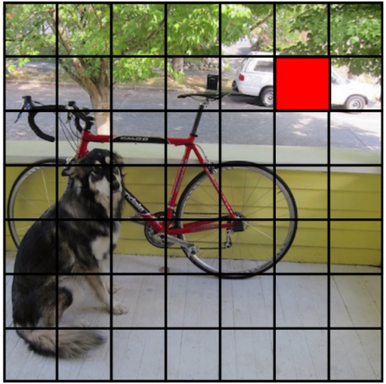
1. Task statement, datasets and metrics
2. Object detection via classification
3. R-CNN, Fast R-CNN, Faster R-CNN
4. YOLO
5. RetinaNet
6. Anchor-free detection

You Only Look Once (YOLO)



Split image into cells

You Only Look Once (YOLO)

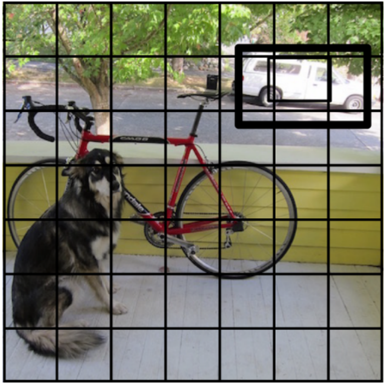


For every cell predict P(Object) and bboxes

You Only Look Once (YOLO)

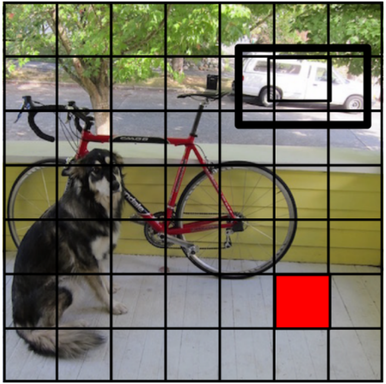
For every cell predict P(Object) and bboxes

You Only Look Once (YOLO)



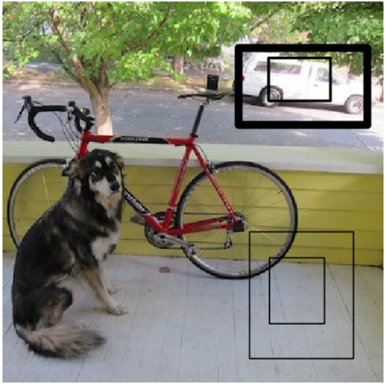
For every cell predict P(Object) and bboxes

You Only Look Once (YOLO)



For every cell predict P(Object) and bboxes

You Only Look Once (YOLO)

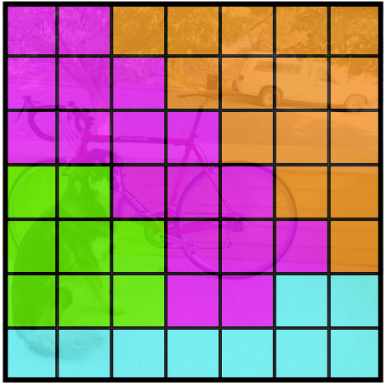


For every cell predict P(Object) and bboxes

You Only Look Once (YOLO)

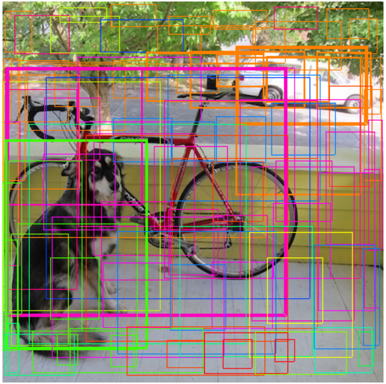
For every cell predict P(Object) and bboxes

You Only Look Once (YOLO)



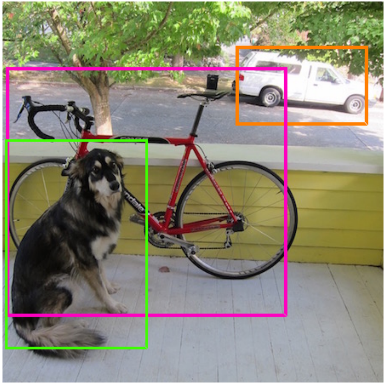
For every cell also predict P(Class)

You Only Look Once (YOLO)



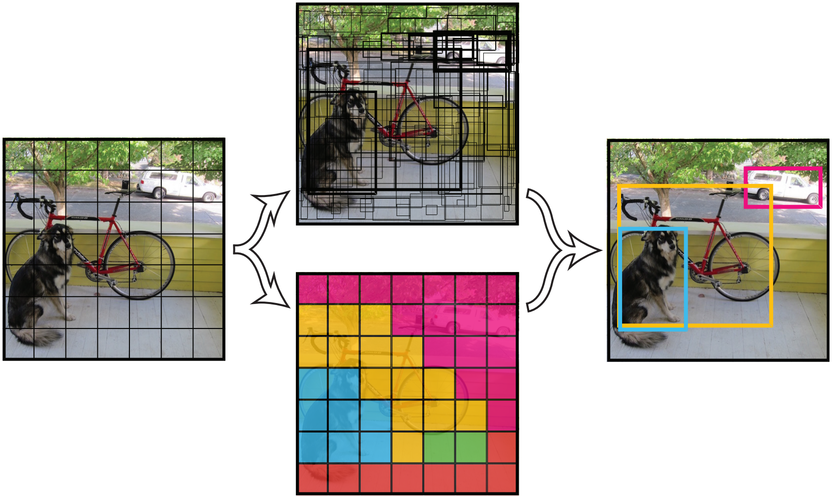
Combine bboxes and class probabilities

You Only Look Once (YOLO)

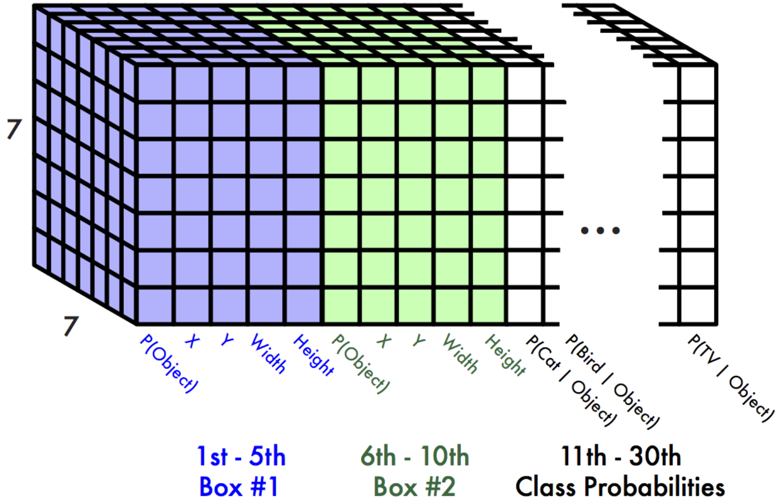


Apply NMS and probability thresholding

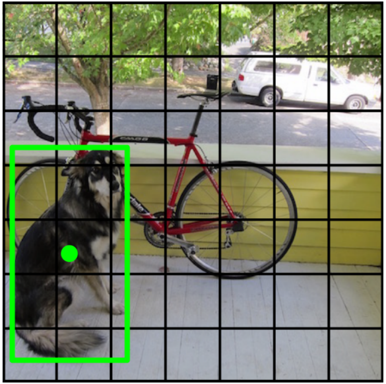
You Only Look Once (YOLO)



YOLO outputs

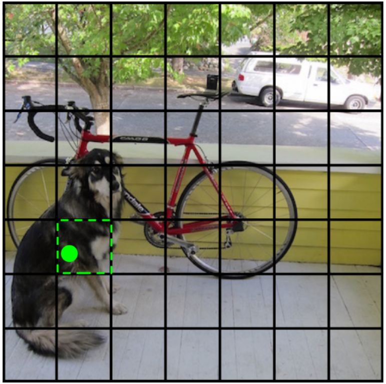


YOLO training



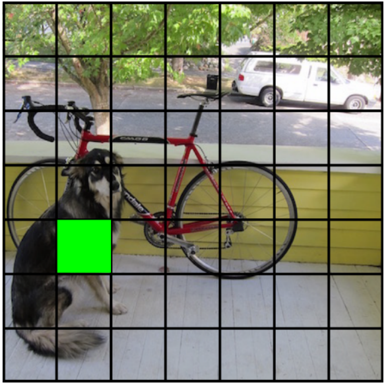
Find a cell for a training sample

YOLO training



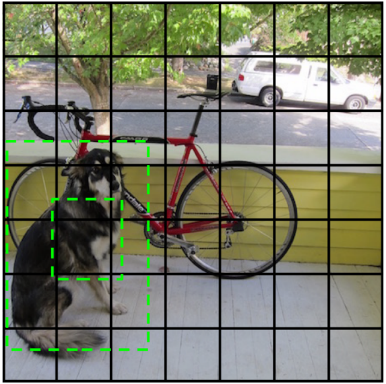
Find a cell for a training sample

YOLO training



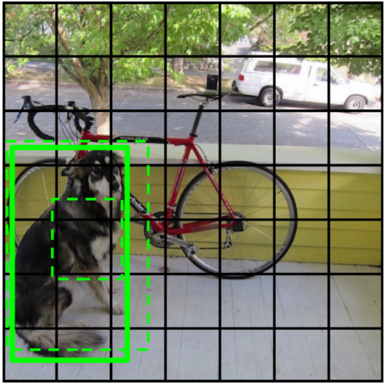
Define probability vector using that training sample

YOLO training



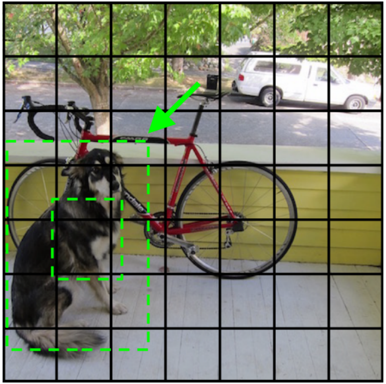
Look at predicted bboxes for that cell

YOLO training



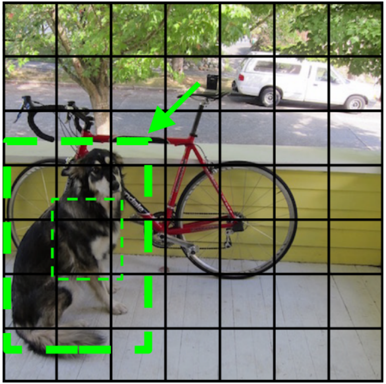
Find nearest bbox

YOLO training



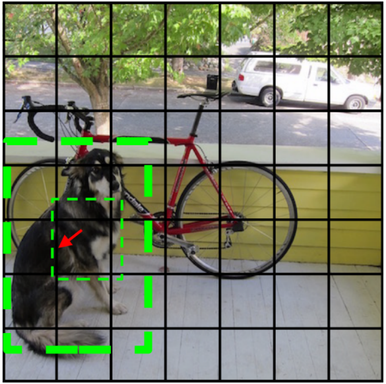
Find nearest bbox

YOLO training



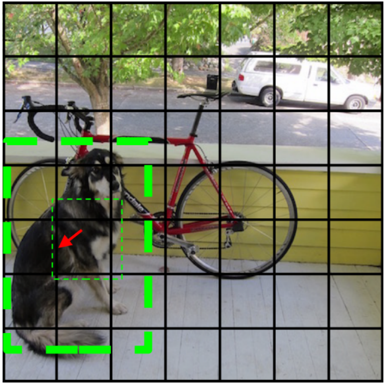
And increase P(Object) for that bbox

YOLO training



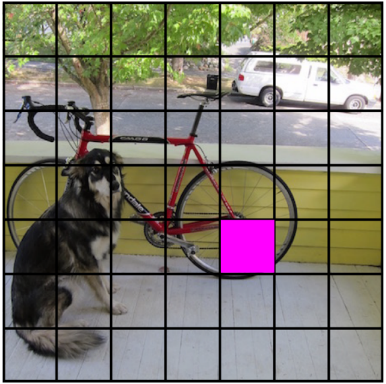
Lower P(Object) for other bboxes

YOLO training



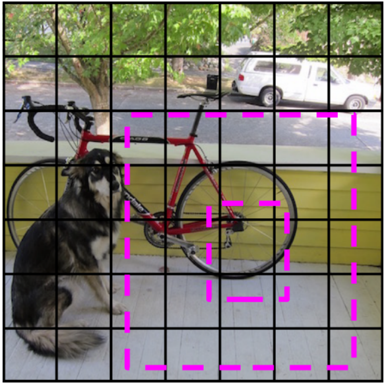
Lower P(Object) for other bboxes

YOLO training



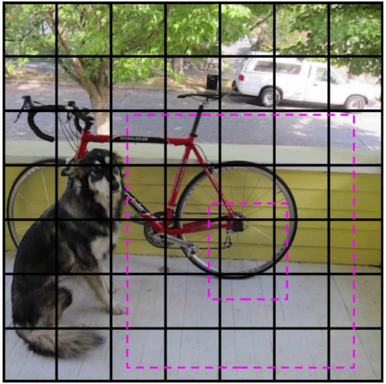
Some cells don’t have corresponding training samples

YOLO training



Lower P(Object) for bboxes in these cells

YOLO training



Lower P(Object) for bboxes in these cells

Comparison with other methods

	Pascal 2007 mAP	Speed
DPM v5	33.7	0.07 FPS
R-CNN	66.0	0.05 FPS
Fast R-CNN	70.0	0.5 FPS
Faster R-CNN	73.2	7 FPS
YOLO	69.0	45 FPS

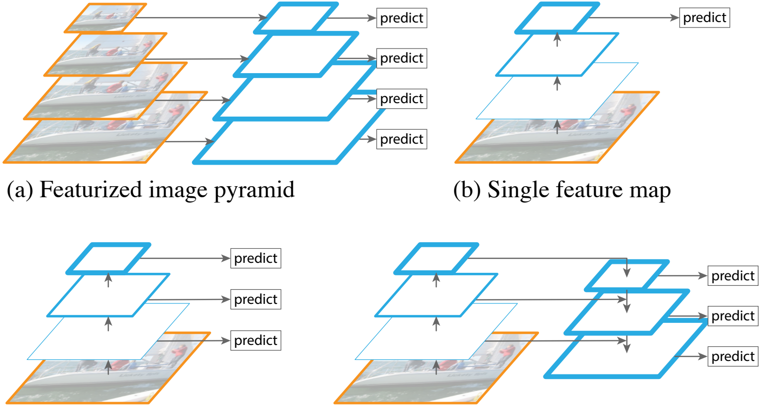
Outline

1. Task statement, datasets and metrics
2. Object detection via classification
3. R-CNN, Fast R-CNN, Faster R-CNN
4. YOLO
5. RetinaNet
6. Anchor-free detection

Feature pyramids

Network receptive field size isn't always similar to object size

Feature pyramids



(a) Featurized image pyramid

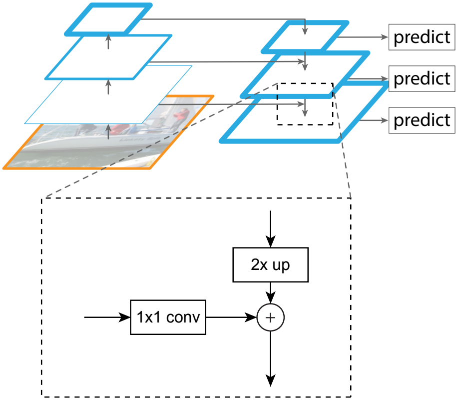
(b) Single feature map

(c) Pyramidal feature hierarchy

(d) Feature Pyramid Network

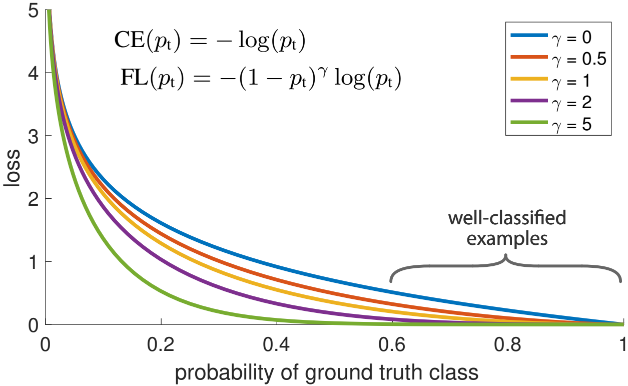
Lin et al. Feature Pyramid Networks for Object Detection. CVPR 2017

Feature pyramids



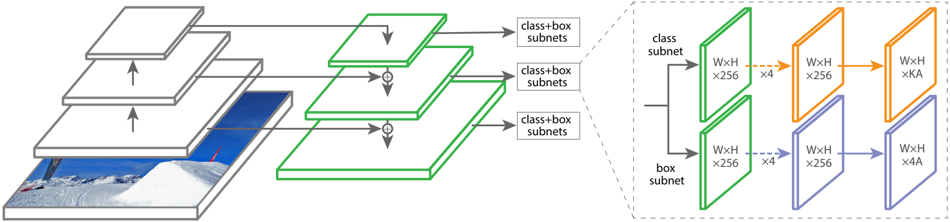
Lin et al. Feature Pyramid Networks for Object Detection. CVPR 2017

Focal loss



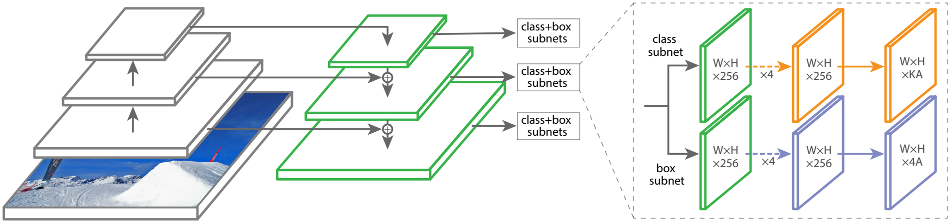
Lin et al. Focal Loss for Dense Object Detection. ICCV 2017

RetinaNet

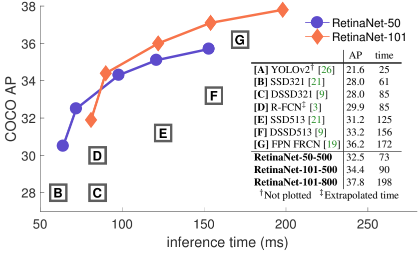


(a) ResNet (b) feature pyramid net (c) class subnet (top) (d) box subnet (bottom)

RetinaNet



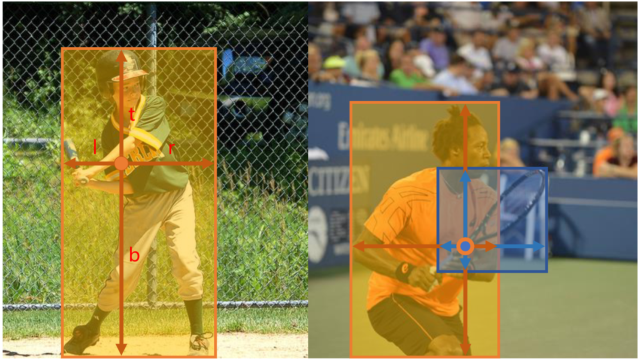
(a) ResNet (b) feature pyramid net (c) class subnet (top) (d) box subnet (bottom)



Outline

1. Task statement, datasets and metrics
2. Object detection via classification
3. R-CNN, Fast R-CNN, Faster R-CNN
4. YOLO
5. RetinaNet
6. Anchor-free detection

FCOS bbox regression

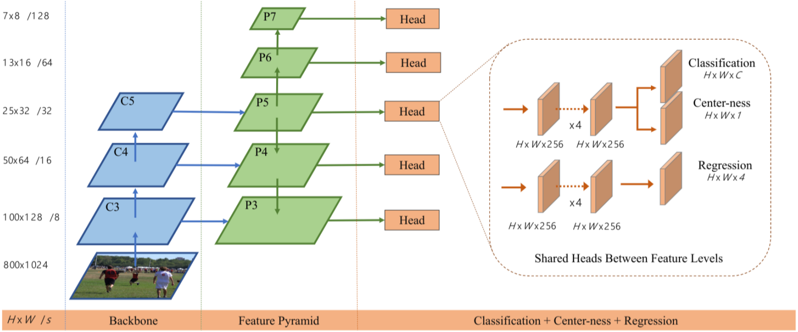


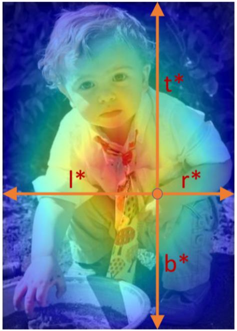
Regress (l,t,r,b) vector in every pixel

Train to predict smallest bbox in case of overlapping bboxes

Tian et al. Fully Convolutional One-Stage Object Detection. ICCV 2019

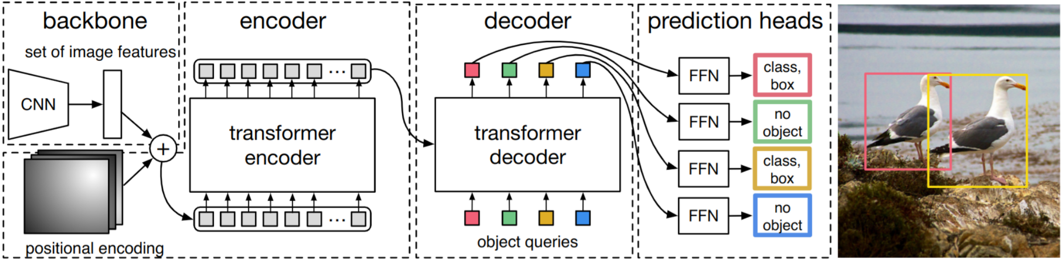
FCOS architecture





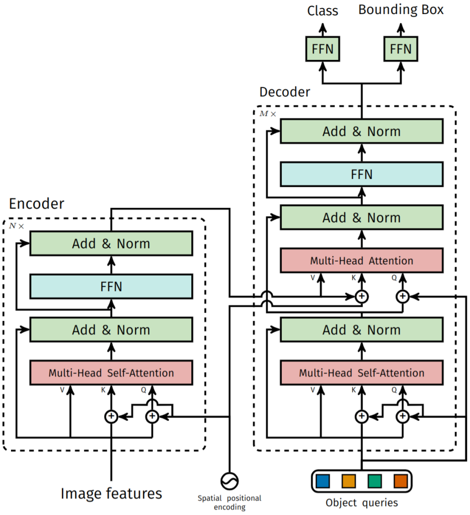
$$\text{centerness}^* = \sqrt{\frac{\min(l^*, r^*)}{\max(l^*, r^*)} \times \frac{\min(t^*, b^*)}{\max(t^*, b^*)}}$$

DETR



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

DETR



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

Conclusion

We reviewed following topics:

- object detection task, metrics and datasets
- development of two-stage R-CNN detector
- single stage detector YOLO
- using feature pyramids for improving detection quality on different object resolutions
- anchor-free detectors FCOS and DETR