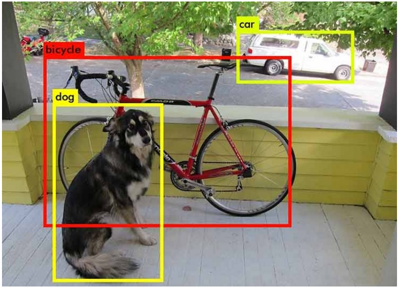
Object detection

Victor Kitov

v.v.kitov@yandex.ru

Problem statement



Training dataset:

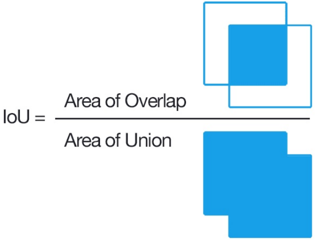
- img1.jpg: X(1), Y(1), X(2), Y(2), class
- img2.jpg: X(1), Y(1), X(2), Y(2), class
- ....

Popular object detection datasets

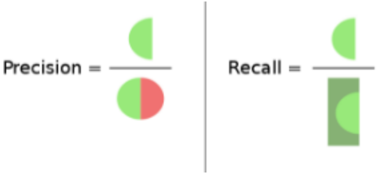
Dataset	train		validation		trainval		test	
	images	objects	images	objects	images	objects	images	objects
VOC-2007	2,501	6,301	2,510	6,307	5,011	12,608	4,952	14,976
VOC-2012	5,717	13,609	5,823	13,841	11,540	27,450	10,991	-
ILSVRC-2014	456,567	478,807	20,121	55,502	476,688	534,309	40,152	-
ILSVRC-2017	456,567	478,807	20,121	55,502	476,688	534,309	65,500	-
MS-COCO-2015	82,783	604,907	40,504	291,875	123,287	896,782	81,434	-
MS-COCO-2018	118,287	860,001	5,000	36,781	123,287	896,782	40,670	-
OID-2018	1,743,042	14,610,229	41,620	204,621	1,784,662	14,814,850	125,436	625,282

Quality measures

- Intersection over union (IoU) = Jaccard similarity



- Precision$=TP/\widehat{P}$, Recall$=TP/P$

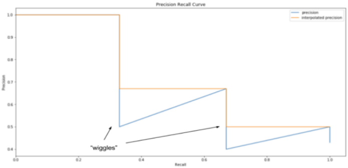


Precision-recall curve

- Fix confidence threshold and minimal IoU threshold.
- Precision-recall curve = Precision(Recall)
- To make it smoother, *interpolated precision* is used:

$$Pr_{interp}(Rec) = \max_{\tilde{Rec} \leq Rec} Pr(\tilde{Rec})$$

1	TP/FP	Precision	Recall	Precision_inter
2	TP	1/1 = 1	1/3 = 0.33	1
3	FP	1/2 = 0.5	1/3 = 0.33	1
4	TP	2/3 = 0.67	2/3 = 0.67	0.67
5	FP	2/4 = 0.5	2/3 = 0.67	0.67
6	FP	2/5 = 0.4	2/3 = 0.67	0.67
7	TP	3/6 = 0.5	3/3 = 1	0.5
8	FP	3/7 = 0.43	3/3 = 1	0.5



Average precision, mean average precision

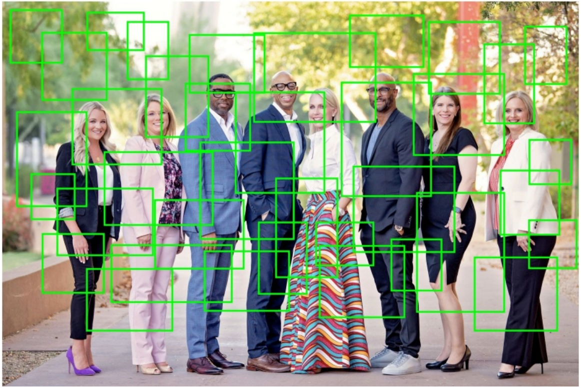
- Average precision (AP) - area under the curve:

$$AP = \frac{1}{K+1} \sum_{Rec \in \{0, \frac{1}{K}, \frac{2}{K}, \dots \frac{K}{K}\}} Pr_{interp}(Rec)$$

- Mean average precision - averaged AP over object classes:

$$mAP = \frac{1}{C} \sum_{c=1}^{C} AP(c)$$

Naive approach



Naive approach: extract patches at all locations at different sizes and apply classifier to each patch.

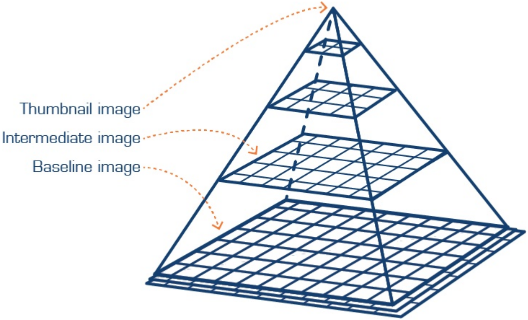
- inefficient, but represents the idea of later approaches.

Extracting bounding boxes of different size

To extract bounding boxes of different size:

- extract bboxes of the same size applied to rescaled versions of image.

Gaussian pyramid



Need also to check bboxes of different shape.

How to solve object detection?

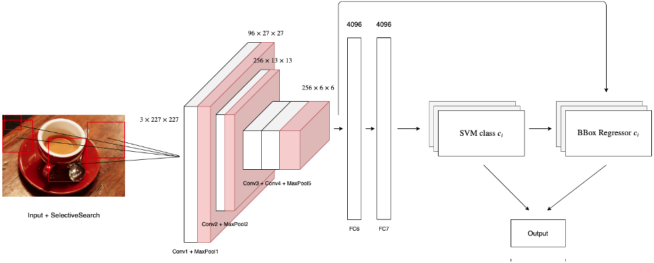
- Selective search algorithm[1] generates region proposals.

- Idea: cluster RGB+XY information into clusters of similar color and spatial position. Join neighboring segments similar in color. Draw a boundary around each segment.

[1] http://www.huppelen.nl/publications/selectiveSearchDraft.pdf

R-CNN[2]

R-CNN scheme:



[2]Girshik et al. Rich feature hierarchies for accurate object detection and semantic segmentation.

R-CNN algorithm

1. SelectiveSearch algorithm generates region proposals ~ 2000
2. Region proposals rescaled to 224x224 (to match AlexNet)
3. AlexNet convolutional layers (except FC layers) extract 4096 features for each region.
4. SVM classifier is trained on $C+1$ class (+1 for background)
5. Regression is trained to correct coordinates of region proposal:

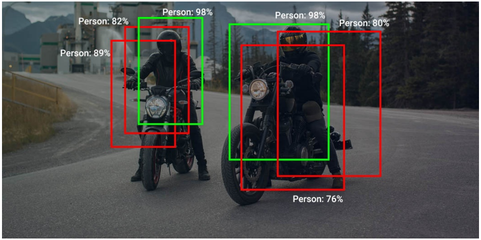
$$\left(\widehat{x},\widehat{y},\widehat{h},\widehat{w}\right)\text{-predicted bbox},\ (x,y,h,w)\text{-true bbox}$$

$$\text{regr. predicts: } \left(\frac{x-\widehat{x}}{w},\frac{y-\widehat{y}}{h},\ln\left(\frac{\widehat{w}}{w}\right),\ln\left(\frac{\widehat{h}}{h}\right)\right)$$

trained on bboxes with IoU>0.3 with true bbox

6. Eliminate redundant bboxes (non-maximum suppression)

Non-maximum supression

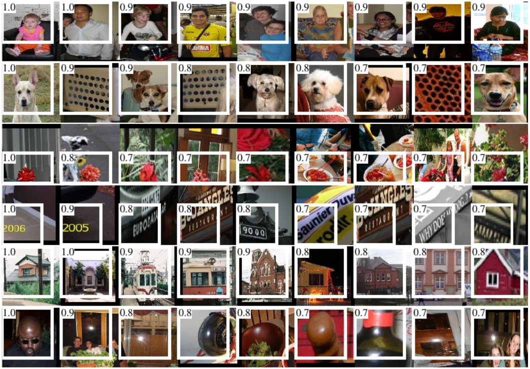


Non-maximum suppression:

- drop low confidence regions
- order bboxes by decreasing confidence
- starting from max confident region downwards:
    - if region has high IoU with other bbox of smaller confidence, drop the latter

Neurons interpretation

Visualization of patches, activating certain neurons inside R-CNN the most.



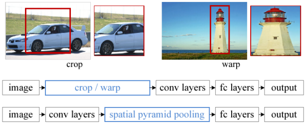
Drawbacks of R-CNN

Drawbacks of R-CNN:

- need separate training for
    - CNN (finetuning)
    - classification (SVM)
    - regression
- for each region proposal need to run CNN
    - and many regions overlap

SPP-net

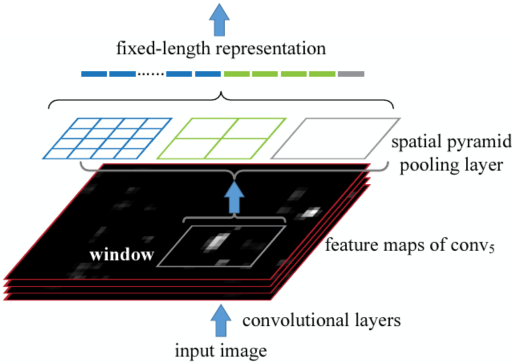
- Problems of R-CNN:
    - CNN reapplied to each region proposal
    - region proposal requires rescaling to 224x224 (image deformation & information loss due to cropping)



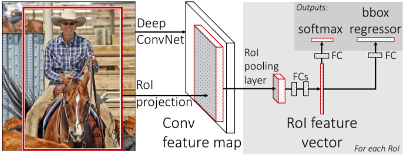
Ideas of SPP-net (spatial pyramid pooling net):

- apply CNN one to whole image (instead of applying for each region proposal)
    - speedup 10x - 100x times.
- spatial pooling allows to output fixed size image representation for image of arbitrary size.

Spatial pyramid pooling



Fast R-CNN



- Regions extracted using SelectiveSearch.
- Image passed through CNN once as a whole.
- Region proposals rescaled to CNN final feature map.
- Pyramid pooling with single layer (grid 7x7) used to match tensor->fixed size representation ("ROI pooling").
- Classifier and bbox regression implemented using additional layers of NN.
  - end2end loss=classification loss+bbox regression loss