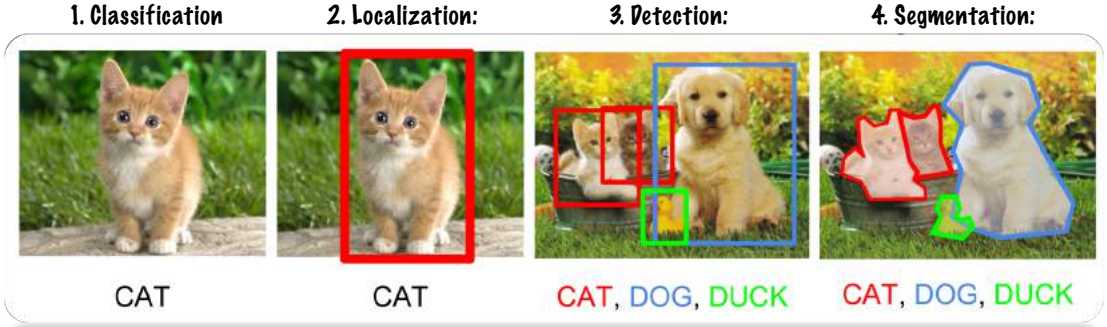
Deep Learning 4 Object Detection

Michal Drozdzal
mdrozdzal@fb.com

About me

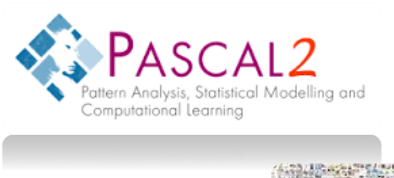
Index

Computer Vision tasks



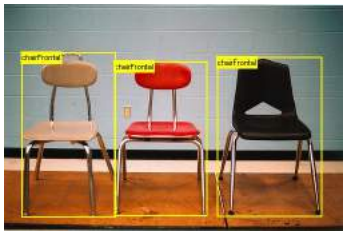
https://chaosmail.github.io/deeplearning/2016/10/22/intro-to-deep-learning-for-computer-vision/

Datasets



2005-2012

Classification Detection Segmentation

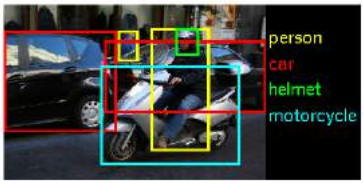


- 20 categories
- 6k training images (17k objects)
- 6k validation + 10k test

Localization

- 1000 categories
- 1.2M training images
- 150k validation + test images

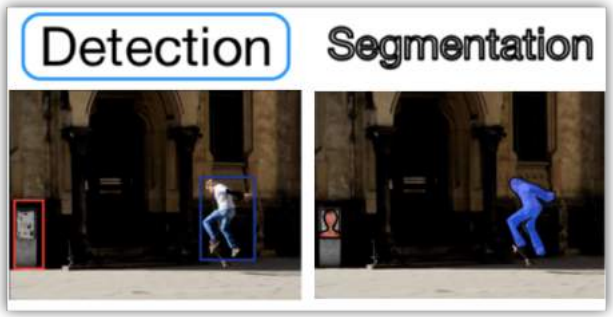
Detection



- 200 categories
- 456k training images
- 60k validation + test images

Detection from video

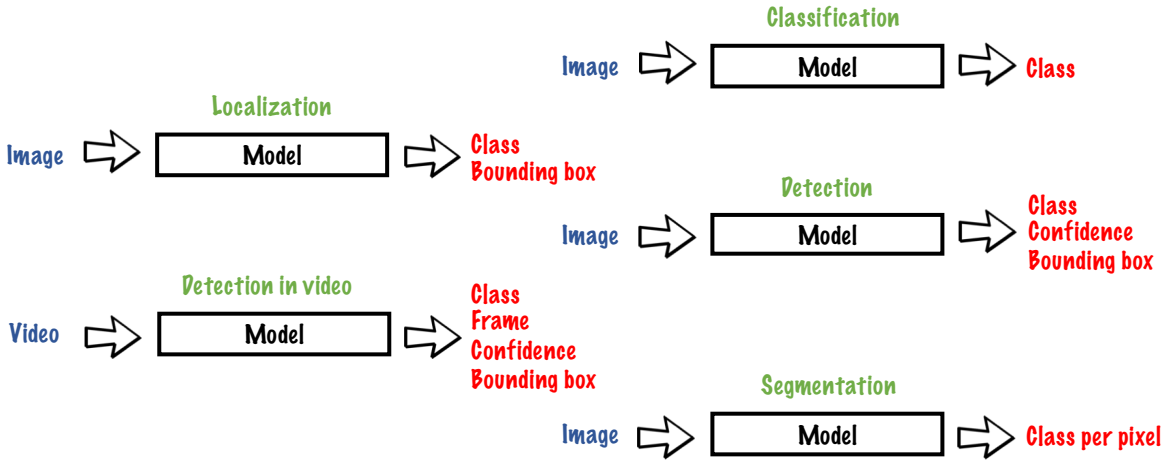
- 30 categories
- 6k videos



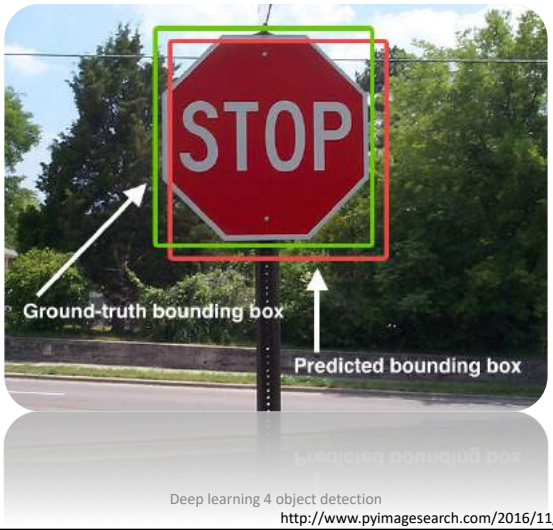
- 80 categories
- 160k images
- 1M instances (350k people)

http://presentations.cocodataset.org/COCO17-Detect-Overview.pdf

Let's define an input and an output for each task



Evaluation



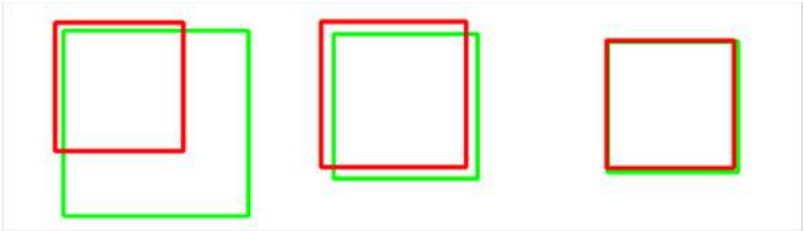
http://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

Intersection over Union (IoU)

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

http://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

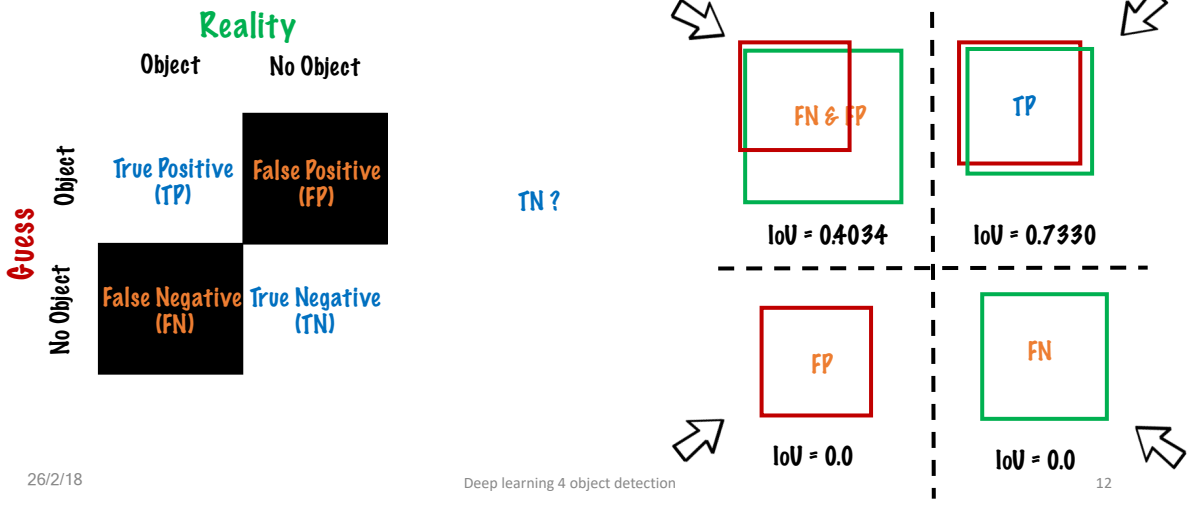
Intersection over Union (IoU)



IoU = 0.4034 IoU = 0.7330 IoU = 0.9264

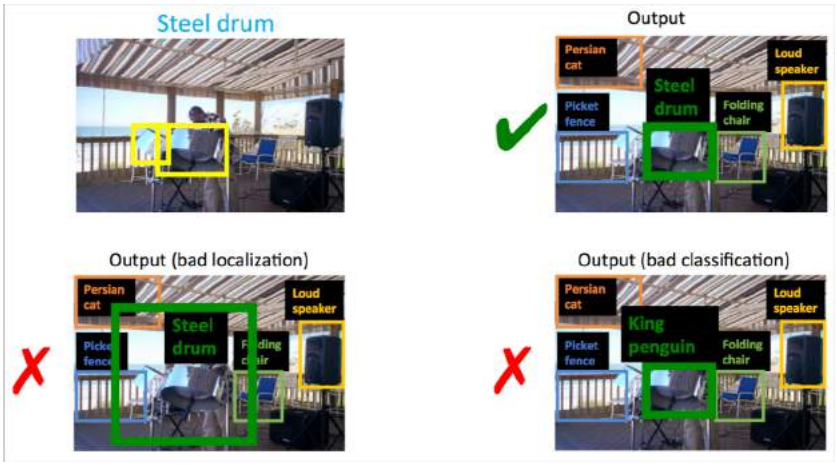
http://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

Being correct, being wrong



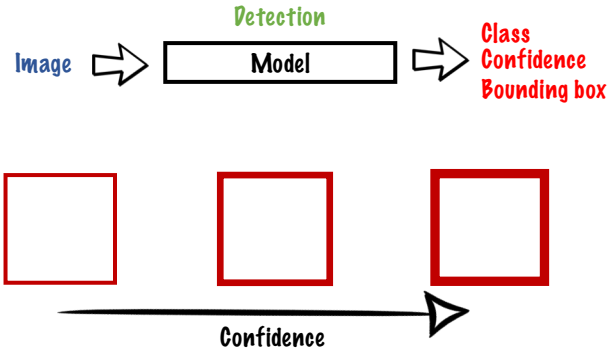
Localization

Localization error:
Wrong class or bad localization (IoU<0.5)



http://image-net.org/challenges/talks/2016/ILSVRC2016_10_09_clsloc.pdf

Detection



Metrics:
1) Average Precision (AP)
2) Average Recall (AR)

Computing Average Precision (A four step procedure)

1. Order the predictions using confidence
2. Compute Precision and Recall
3. Plot Precision Recall plot (optional)
4. Compute Average Precision (AP) and Average Recall (AR)

1. Predictions ordering

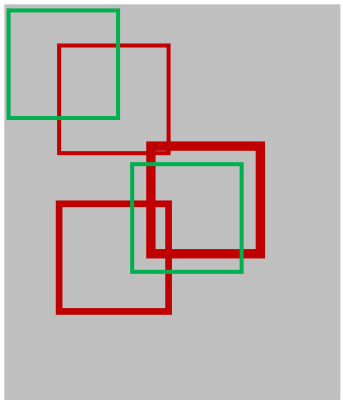
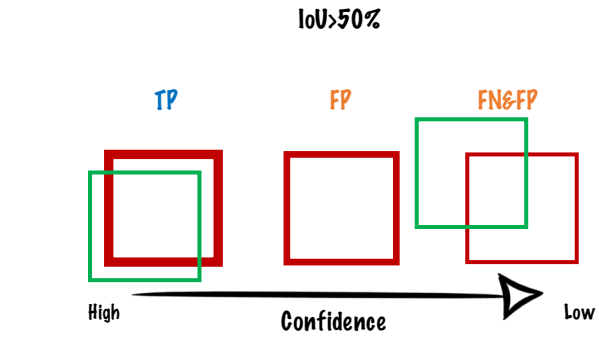


Image
Annotations
Predictions



TP	1	1	1
FP	0	1	2
FN	0	0	1

1. Predictions ordering

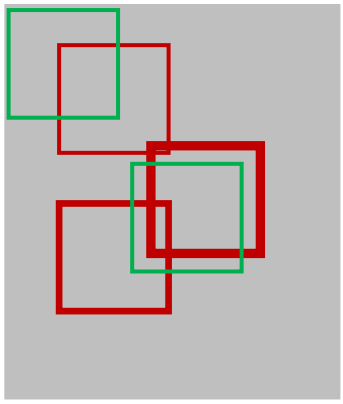
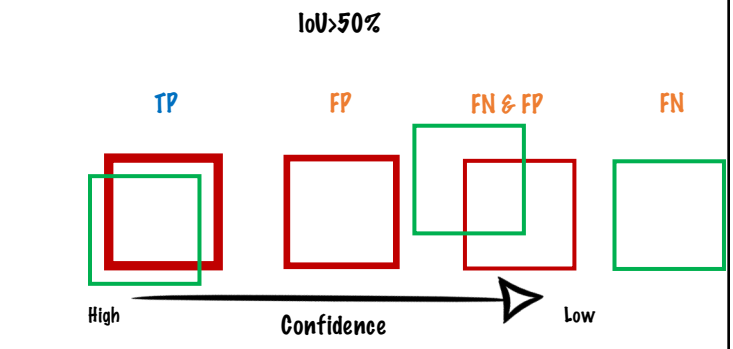
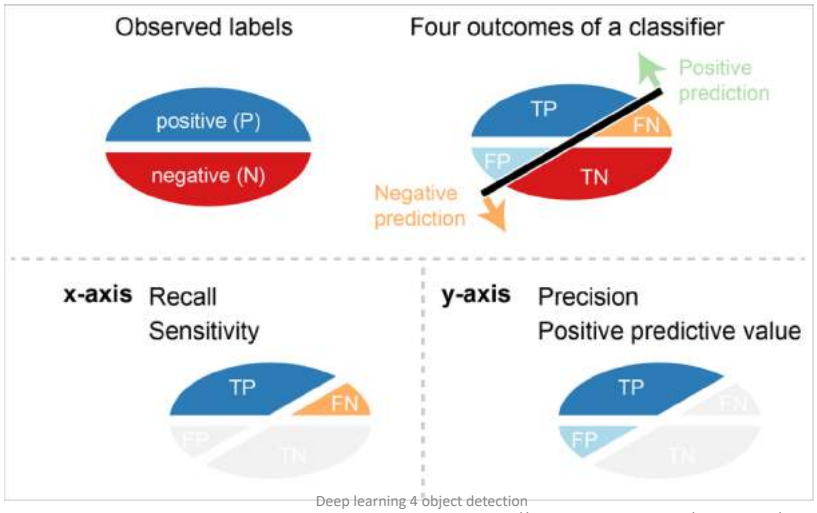


Image
Annotations
Predictions



TP	1	1	1
FP	0	1	2
FN	0	0	1

2. Precision & Recall



https://classeval.wordpress.com/introduction/introduction-to-the-precision-recall-plot/

2. Precision & Recall

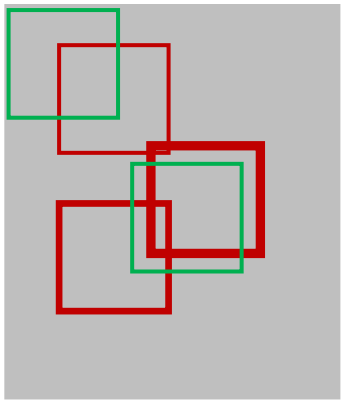
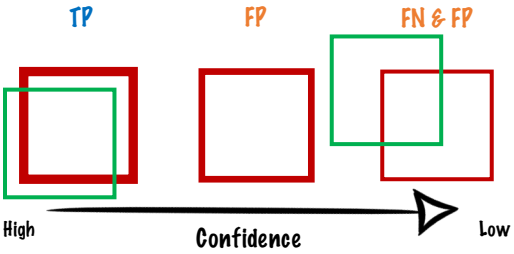
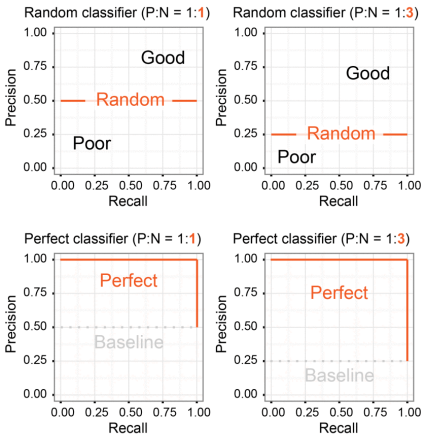


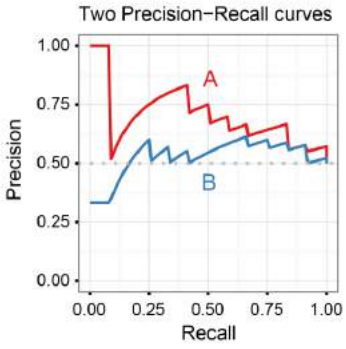
Image
Annotations
Predictions



TP	1	1	1
FP	0	1	2
FN	0	0	1
Precision	1	.5	.3
Recall	.5	.5	1

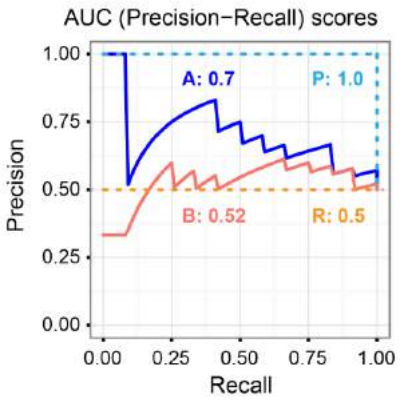
3. Precision - Recall plots

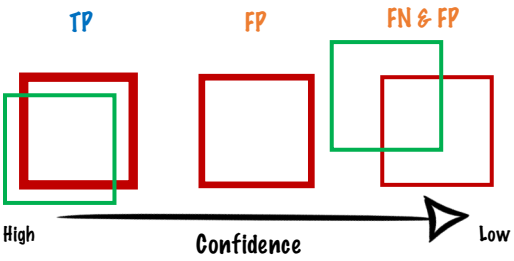




https://classeval.wordpress.com/introduction/introduction-to-the-precision-recall-plot/

4. AP & AR





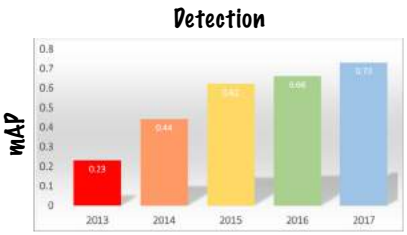
TP	1	1	1
FP	0	1	2
FN	0	0	1
Precision	1	.5	.3
Recall	.5	.5	1

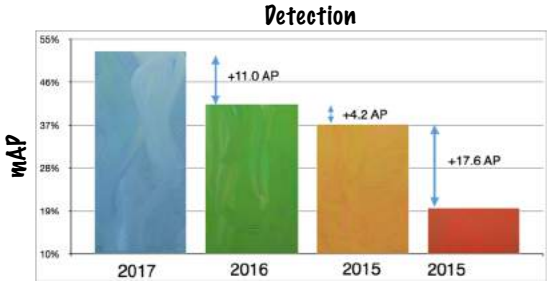
AP@(IoU=50%) = ? R@(1pred) = 0.5

https://classeval.wordpress.com/introduction/introduction-to-the-precision-recall-plot/


```
Average Precision (AP):% AP at IoU=.50:.05:.95 (primary challenge
  AP                   metric)
  AP^IoU=.50           % AP at IoU=.50 (PASCAL VOC metric)
  AP^IoU=.75           % AP at IoU=.75 (strict metric)
AP Across Scales:
  AP^small             % AP for small objects: area < 32^2
  AP^medium            % AP for medium objects: 32^2 < area < 96^2
  AP^large             % AP for large objects: area > 96^2
Average Recall (AR):
  AR^max=1             % AR given 1 detection per image
  AR^max=10            % AR given 10 detections per image
  AR^max=100           % AR given 100 detections per image
AR Across Scales:
  AR^small             % AR for small objects: area < 32^2
  AR^medium            % AR for medium objects: 32^2 < area < 96^2
  AR^large             % AR for large objects: area > 96^2
```


http://mscoco.org/dataset/#detections-eval





http://image-net.org/challenges/talks_2017/ILSVRC2017_overview.pdf
http://presentations.cocodataset.org/COCO17-Detect-Overview.pdf

Index

Part 1
Introduction

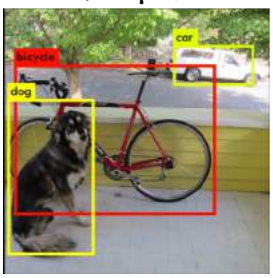
Part 2
Basic blocks & concepts

Part 3
Models

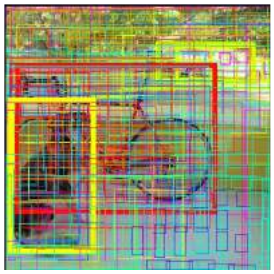
Bonus material
Weakly supervised localization

What is an outcome of an object detector?

We expect:



We get:



Lots of FP.

Image credit: https://pjreddie.com/darknet/yolo/

The unbalanced nature of detection. Hard Negative Mining

Build a head detector.

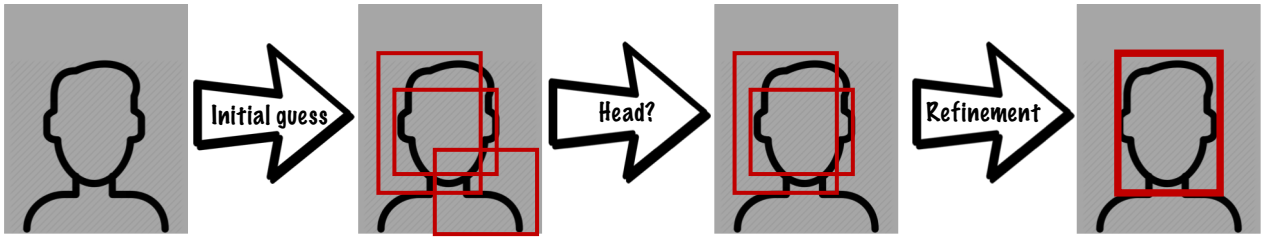
Positive	1
Negative	70
Hard negative	6

Image credit: https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721#.dnbyjd6zg

Object detection pipeline

Given the unbalanced nature of detection

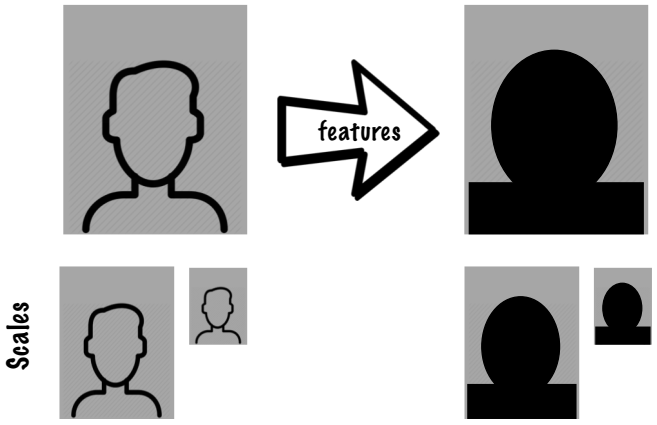
What do we need?



Fast & approximately correct

Slow but precise

Initial guess



Features (in the past)

Histogram of Oriented Gradients

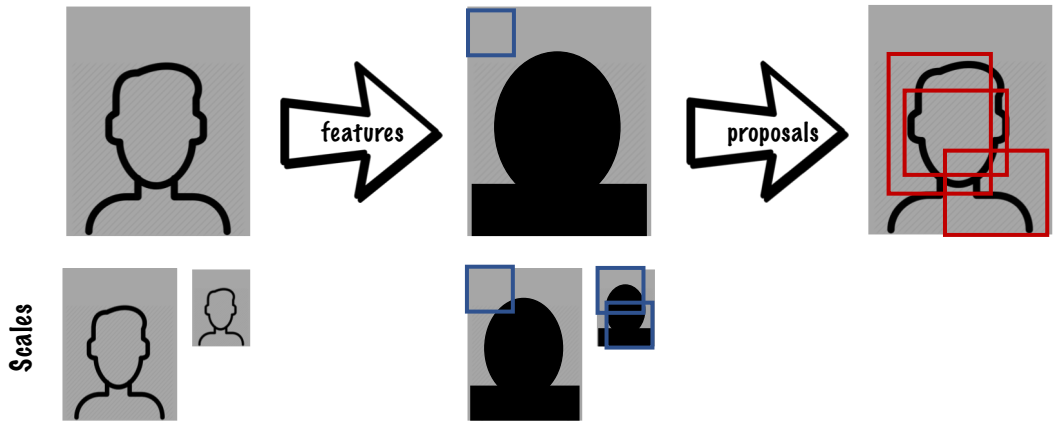
http://sharky93.github.io/docs/gallery/auto_examples/plot_hog.html

Features (currently)

Use pretrained neural networks!

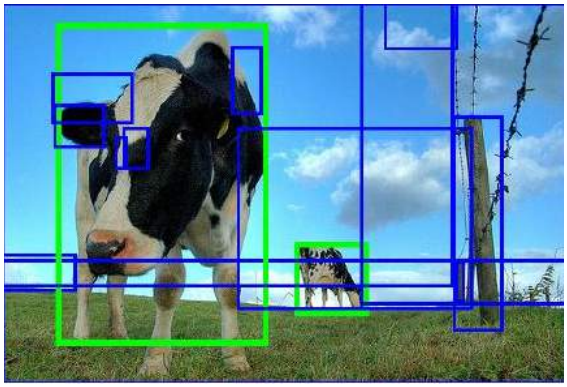
Architectures for Image Classification

Region proposals (Class agnostic classifiers)



Region Proposals (Class agnostic classifiers)

Objectness [1]
Selective search [2]
Category-independent object proposals [3]
Constrained parametric min-cuts (CPMC) [4]
Multi-scale combinatorial grouping [5]



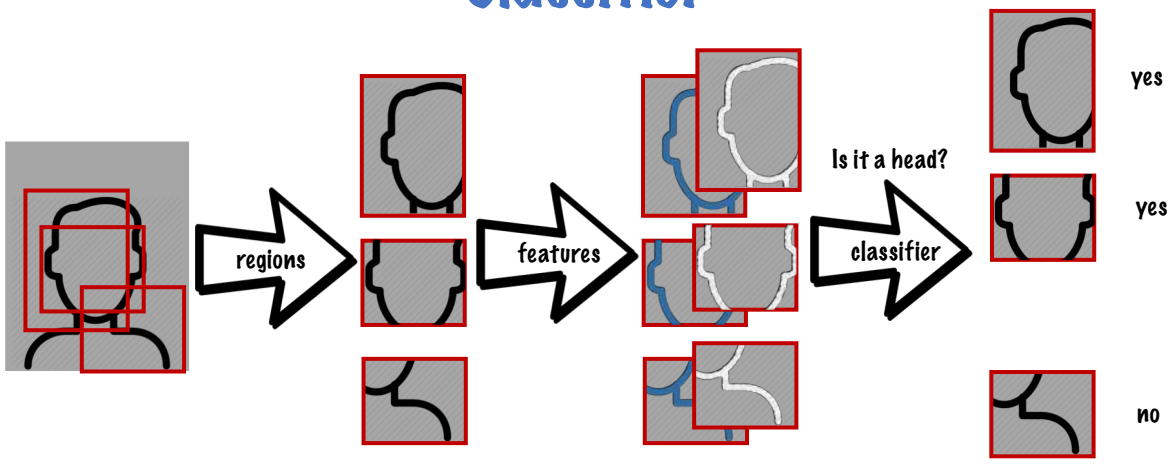
[1] B.Alexe et al.Measuring the objectness of image windows.
[2] J. Uijlings et al. Selective search for object recognition
[3] I. Endres and D. Hoiem. Category independent object proposals
[4] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts
[5] P.Arbelaez et al. Multiscale combinatorial grouping

Image: https://ivi.fnwi.uva.nl/isis/publications/bibtexbrowser.php?key=UijlingsIJCV2013&bib=all.bib

Region Proposals Selective search

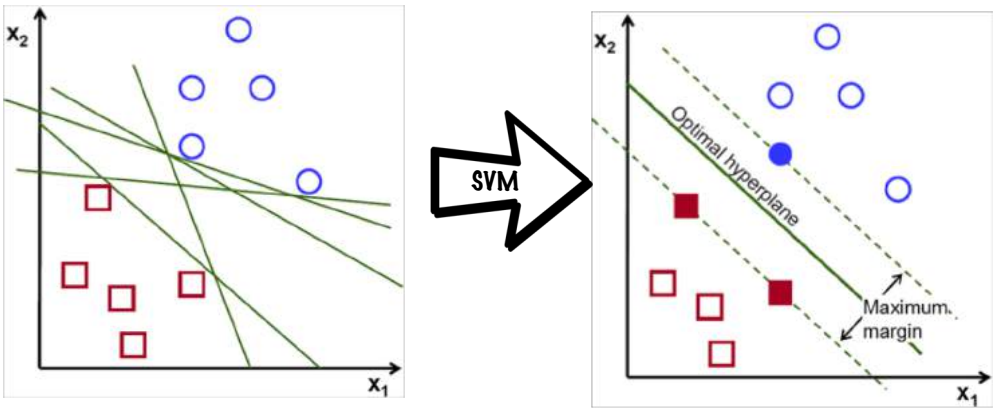
http://koen.me/research/pub/vandesande-iccv2011-poster.pdf

(Object specific) Classifier



Support Vector Machines (in the past)

Head: ☐ No head: ○

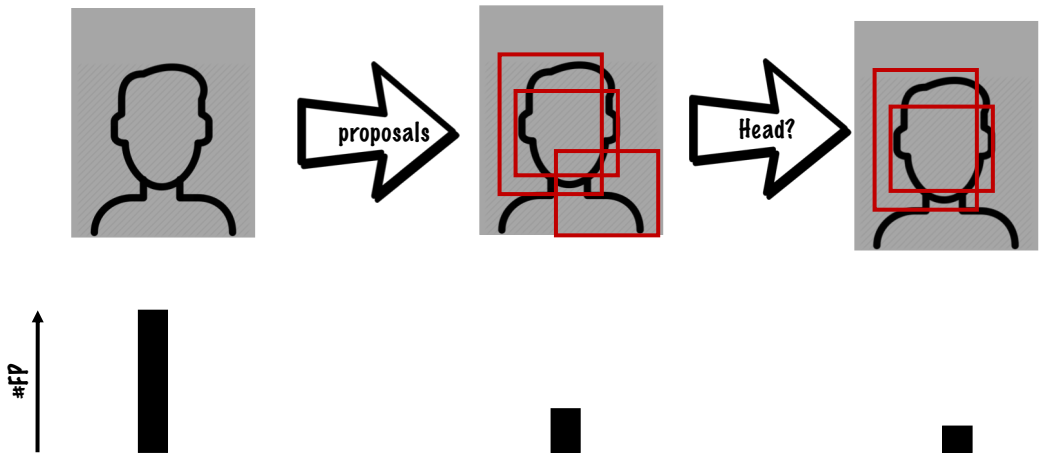


http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

Classifier (Currently)

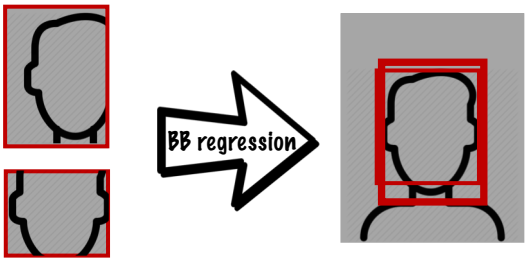
Neural Network!

What happens with FP?

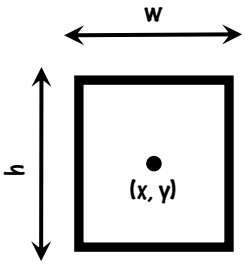


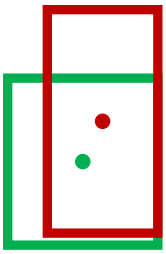
Regressor (Refinement)

Regressor is used to adjust the position of class specific bounding boxes.



Bounding box regression





Annotation (A_x, A_y, A_w, A_h)

Prediction (P_x, P_y, P_w, P_h)

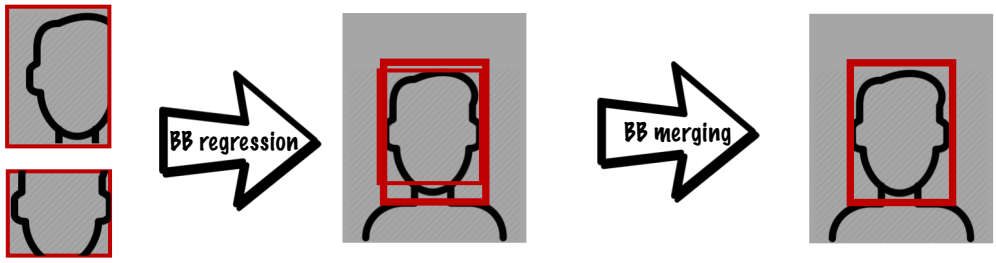
Regression error (t_x, t_y, t_w, t_h)

$t_x = (A_x - Px)/P_w$
$t_y = (A_y - Py)/P_h$
$t_w = \log(A_w/P_w)$
$t_h = \log(A_h/P_h)$

Why divide by P_w and P_h?

Regression loss = $\text{loss}(t_x) + \text{loss}(t_y) + \text{loss}(t_w) + \text{loss}(t_h)$

Regressor (Refinement)



BB merging
Non-Maximum Suppression (NMS)

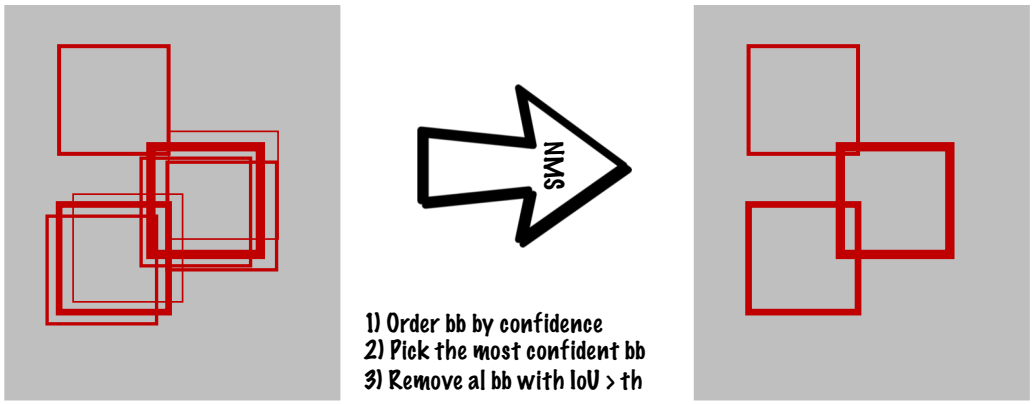


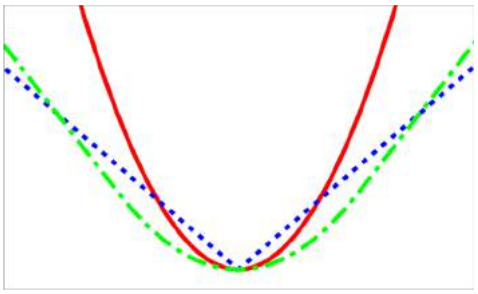
Image
Predictions

Image
Predictions

http://www.computervisionblog.com/2011/08/blazing-fast-nmsm-from-exemplar-svm.html

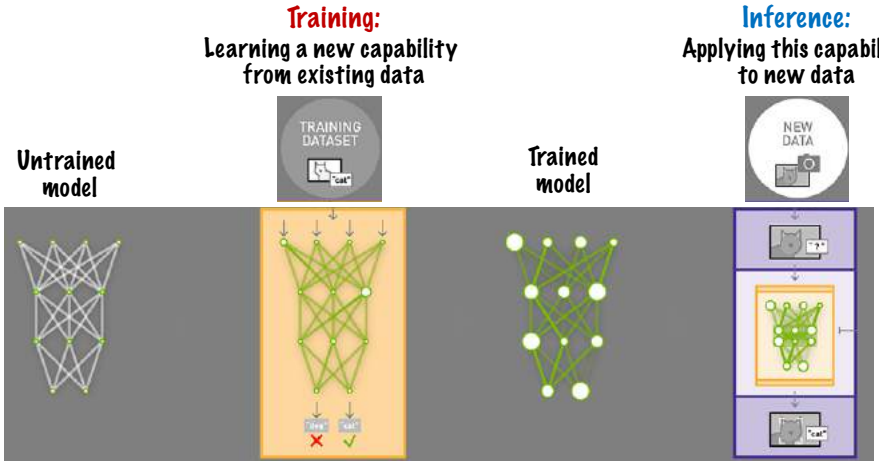
Detection: Loss Functions

- Classification losses:
  - Cross entropy (softmax)
  - Hinge loss (SVM)
- Regression losses:
  - L1
  - Smooth L1
  - L2



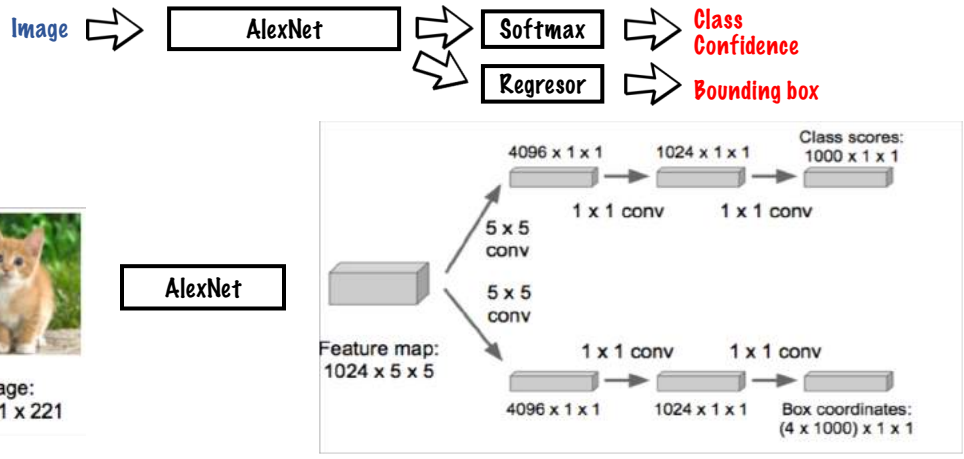
Index

Training vs. Inference



https://blogs.nvidia.com/blog/2016/08/22/difference-deep-learning-training-inference-ai/

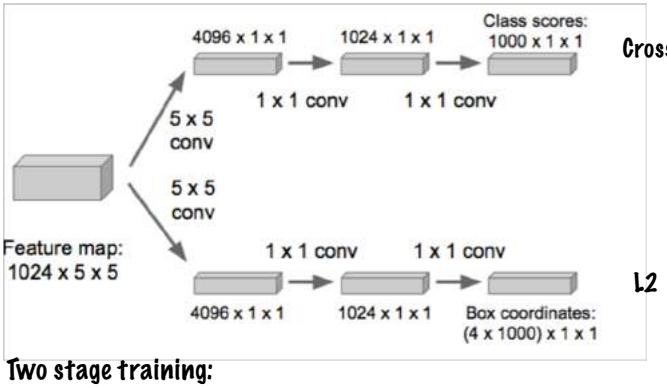
OverFeat (2013, ICLR2014)



http://next.sh/blog/2016/03/24/lecture-8-localization--detection-stanford-cs231n/

OverFeat (2013, ICLR2014): training

AlexNet



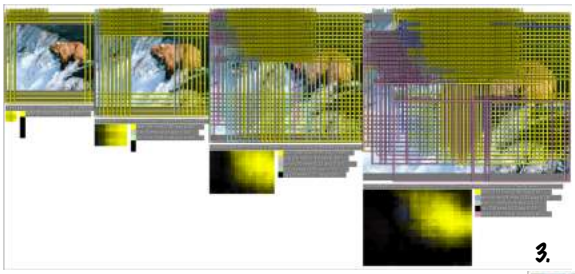
Two stage training:

1. Train the classifier (cross entropy)
2. Train the regressor (L2)

OverFeat (2013, ICLR2014): inference

Apply the network at all positions and scales and predict:

1. Class

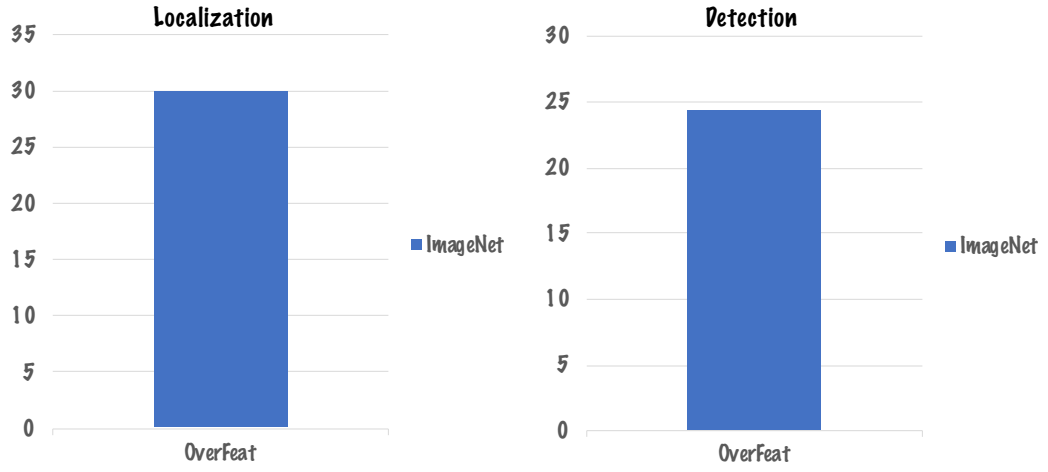


2. Bounding Boxes

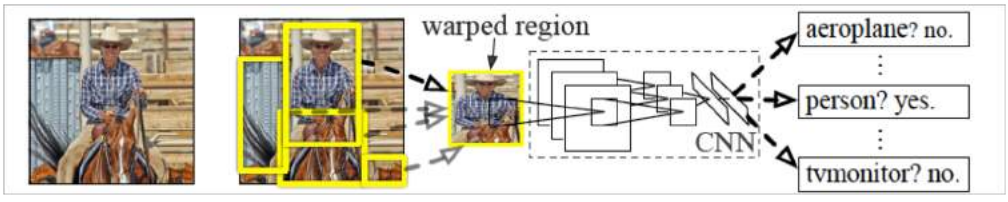
3. Combine predictions

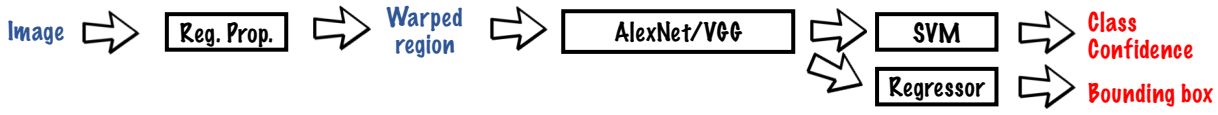
Inference is slow!

OverFeat (2013, ICLR2014): results

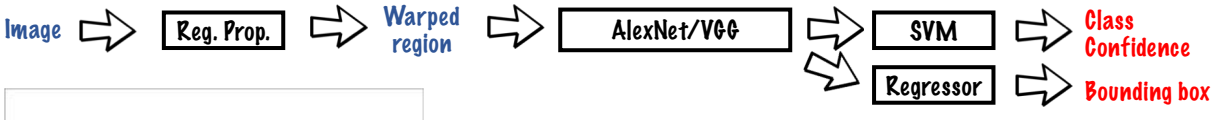


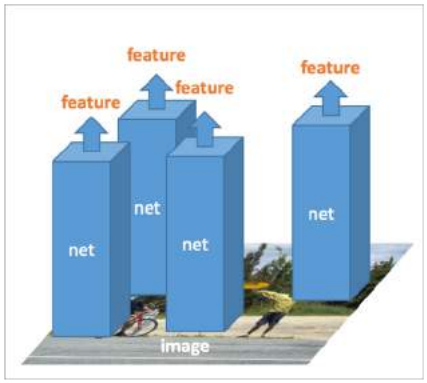
R-CNN (2013, CVPR2014)





R-CNN (2013, CVPR2014): training



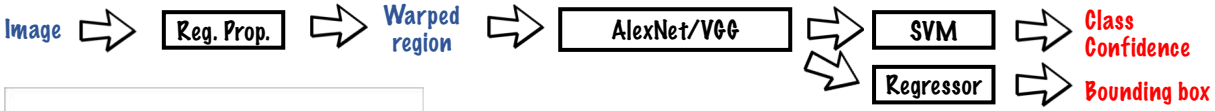


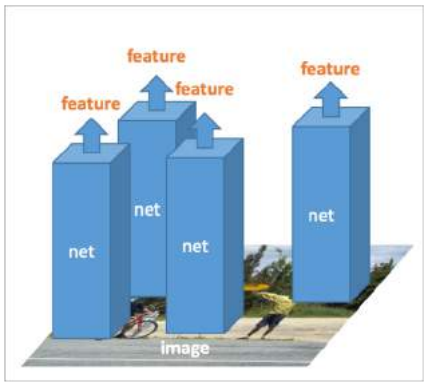
Training:
1. Pre-train network on Imagenet (image classification task)
2. Finetune network with softmax classifier (log loss)
3. Extract features
4. Train linear SVMs with hard negative mining (hinge loss)
5. Train bounding box regressions (least squares)

Training is slow (84h). Why point #2?

http://kaiminghe.com/eccv14sppnet/sppnet_ilsvrc2014.pdf

R-CNN (2013, CVPR2014): inference



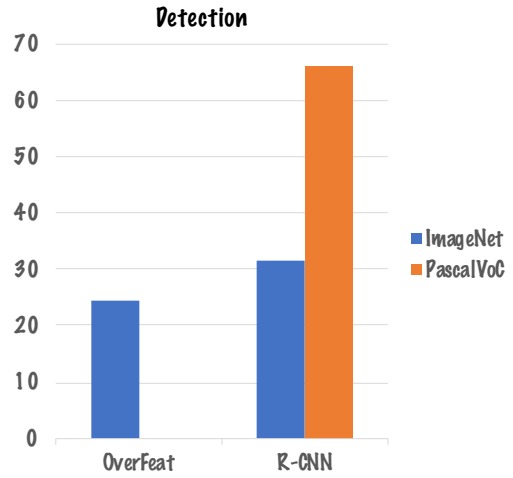


Inference:
1. Extract 2000 region proposals per image
2. Extract features for each proposal
3. Infer class, confidence and bounding box for each proposal

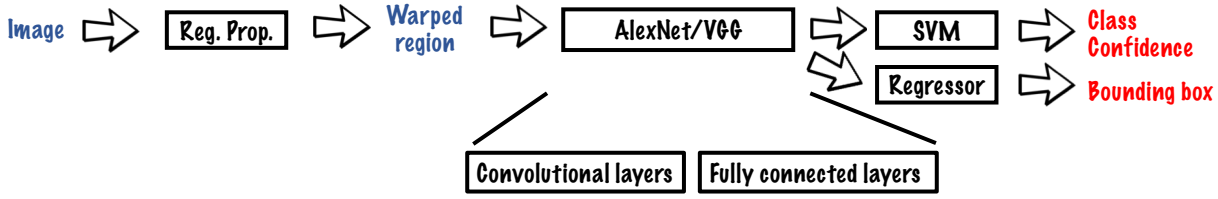
Inference is slow (2k passes of CNN per image).

http://kaiminghe.com/eccv14sppnet/sppnet_ilsvrc2014.pdf

R-CNN (2013, CVPR2014): results

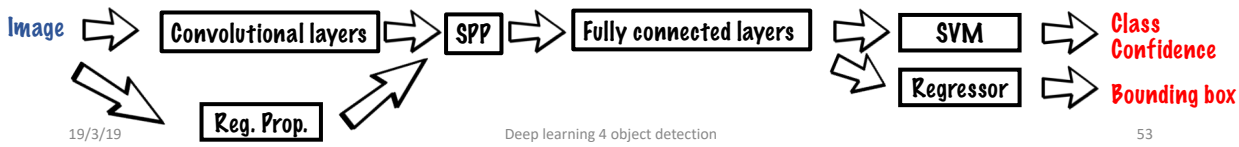


R-CNN (2013, CVPR2014)

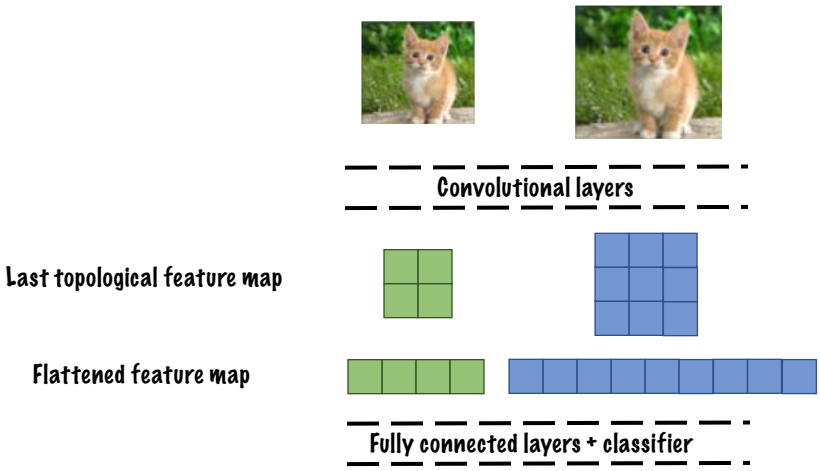


Spatial Pyramid Pooling

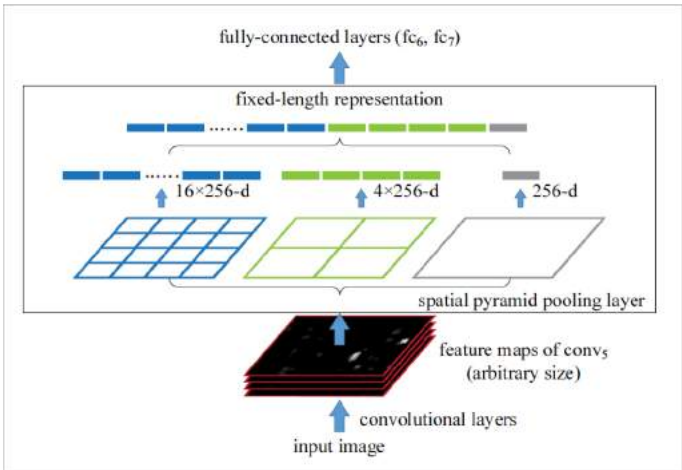
SPP (2014, TPAMI2015)



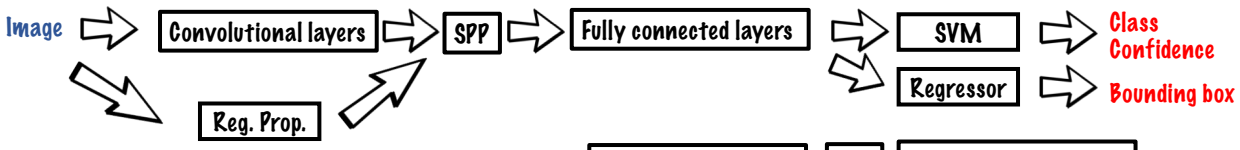
Why do we need to warp images?

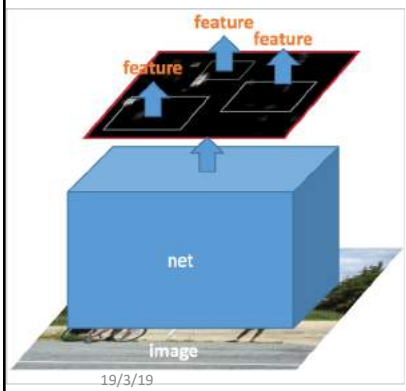


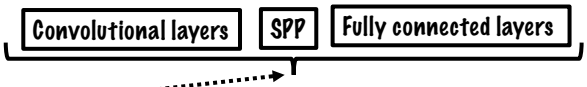
SPP (2014, TPAMI2015)



SPP (2014, TPAMI2015): training





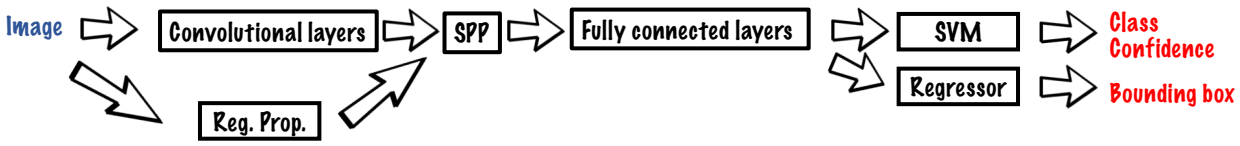


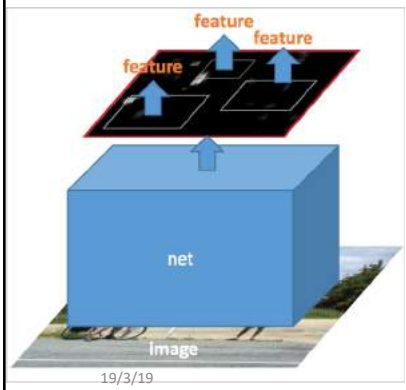
Training:
1. Pre-train network on Imagenet (image classification task)
2. Finetune network with softmax classifier (log loss)
3. Extract features
4. Train linear SVMs with hard negative mining (hinge loss)
5. Train bounding box regressions (least squares)

Training is slow.

http://kaiminghe.com/eccv14sppnet/sppnet_ilsvrc2014.pdf

SPP (2014, TPAMI2015): inference



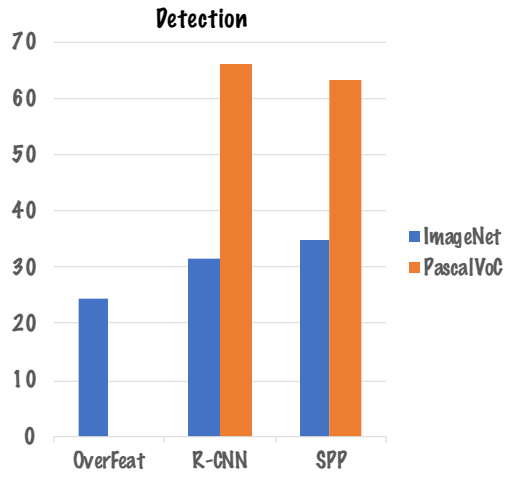


Inference:
1. Compute the feature map of an image
2. Compute region proposals (2k per image)
3. Project region proposals into feature map
4. Infer class, confidence and bounding box for each proposal

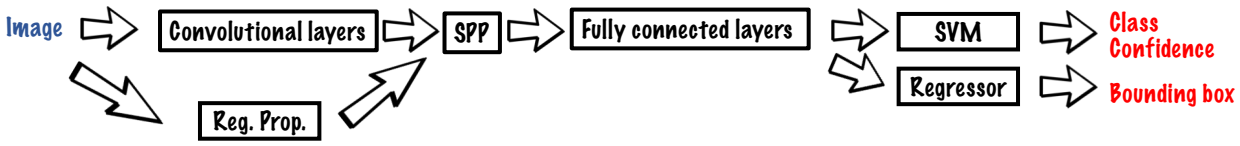
Inference time is OK (up to 100x faster than R-CNN).

http://kaiminghe.com/eccv14sppnet/sppnet_ilsvrc2014.pdf

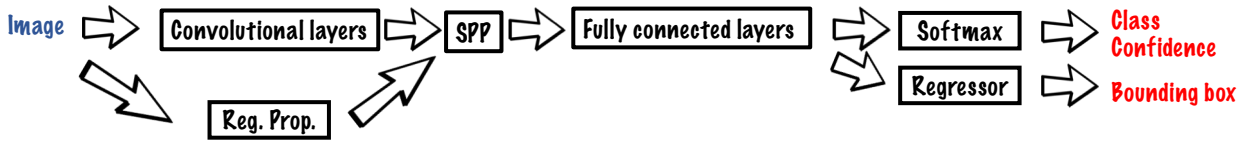
SPP (2014, TPAMI2015): results



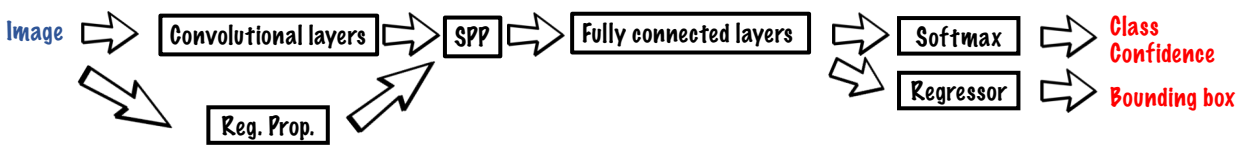
SPP (2014, TPAMI2015)

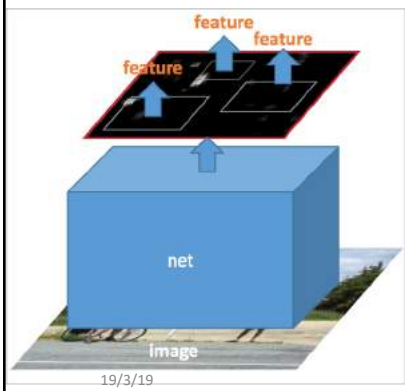


Fast R-CNN (2015, ICCV2015)



Fast R-CNN (2015, ICCV2015): training



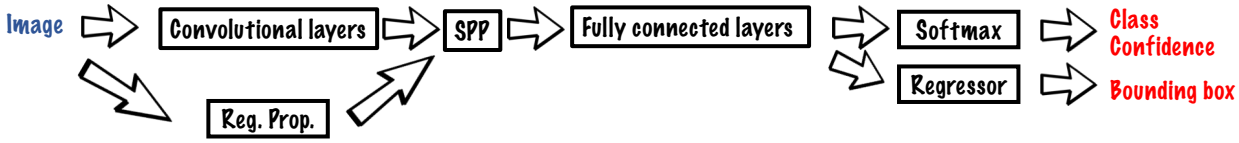


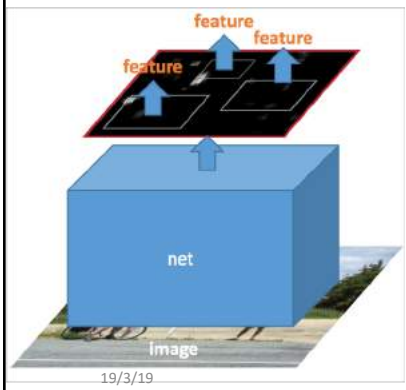
Joint loss: log loss + smooth L1 loss

Training:
1. Pre-train network on Imagenet classification task
2. Train the model with joint loss

Training is elegant and fast.
Region Proposals are still required....

Fast R-CNN (2015, ICCV2015): infrence



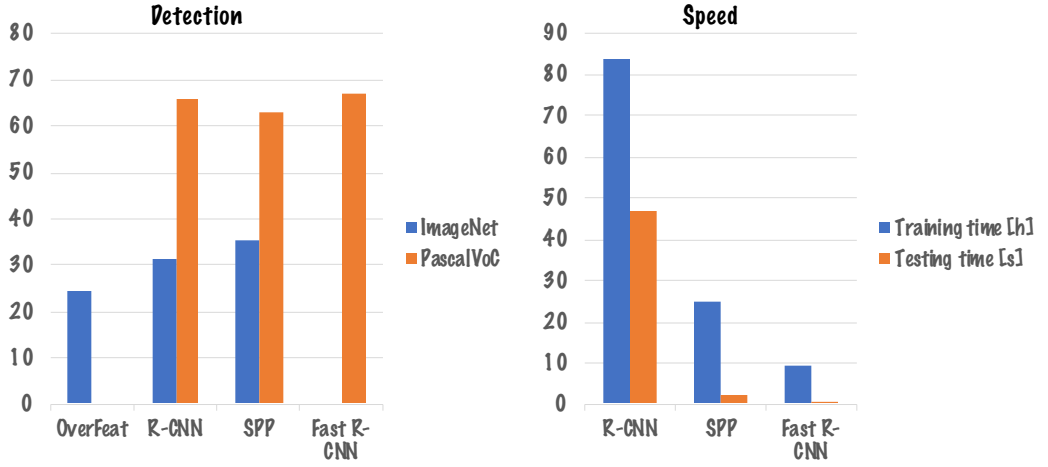


Inference:

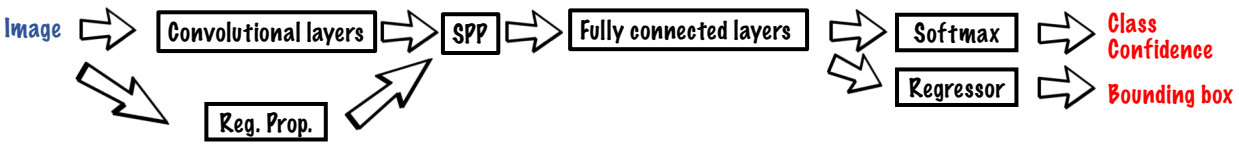
1. Extract feature map
2. Extract region proposals
3. Infer class, confidence and bounding box for each proposal

Inference is fast.

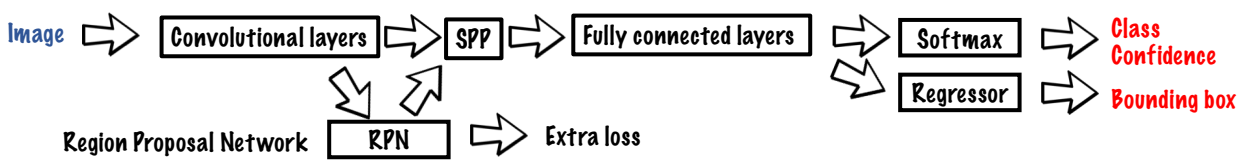
Fast R-CNN (2015, ICCV2015): results



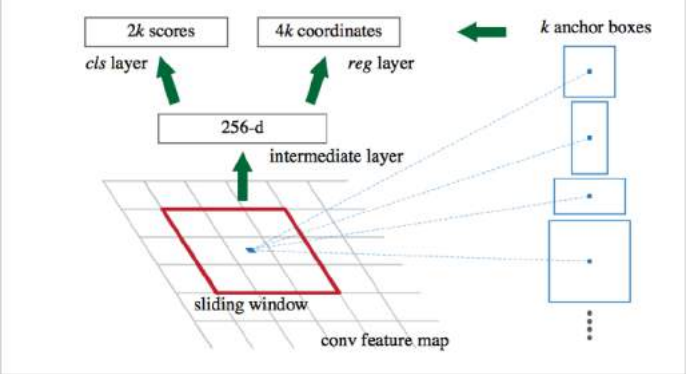
Fast R-CNN (2015, ICCV2015)



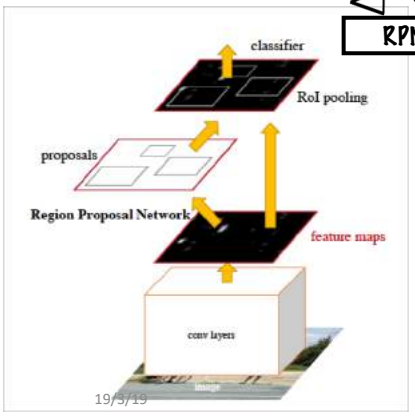
Faster R-CNN (2015, NIPS2015)



Faster R-CNN (2015, NIPS2015): Region Proposal Network (RPN)

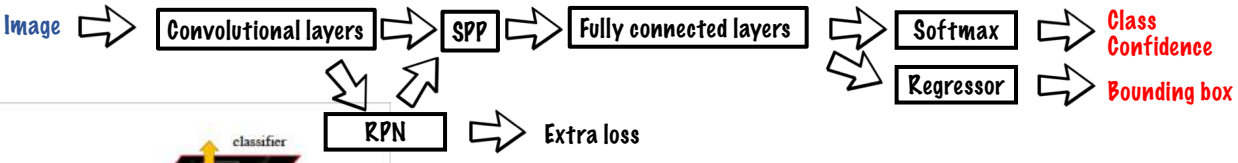


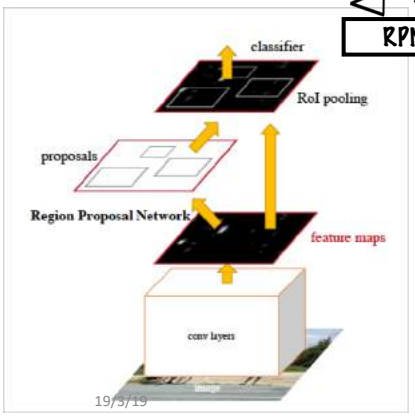
Faster R-CNN (2015, NIPS2015): training



One network, four losses (TPAMI version)
1. RPN classification (anchor good / bad)
2. RPN regression (anchor -> proposal)
3. Fast R-CNN classification (over classes)
4. Fast R-CNN regression (proposal -> box)

Faster R-CNN (2015, NIPS2015): inference

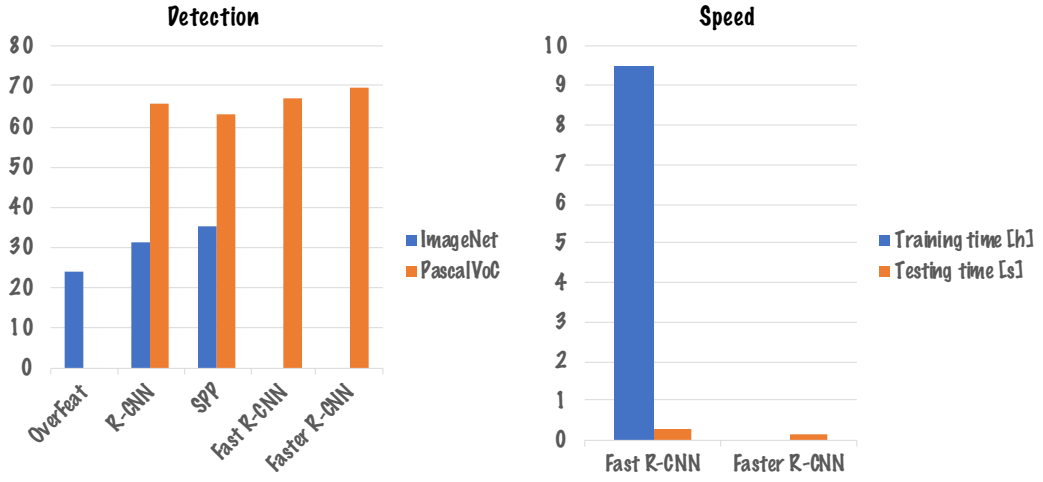




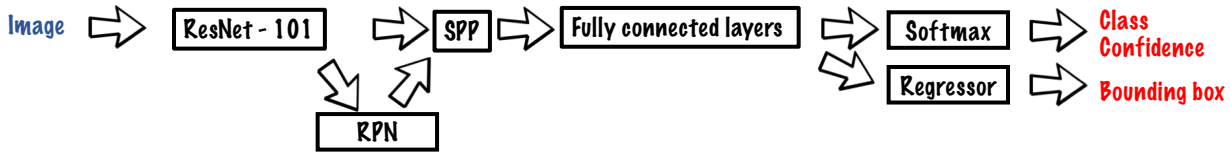
Inference:

1. Extract feature map and region proposals
2. Infer class, confidence and bounding box for each proposal

Faster R-CNN (2015, NIPS2015): results



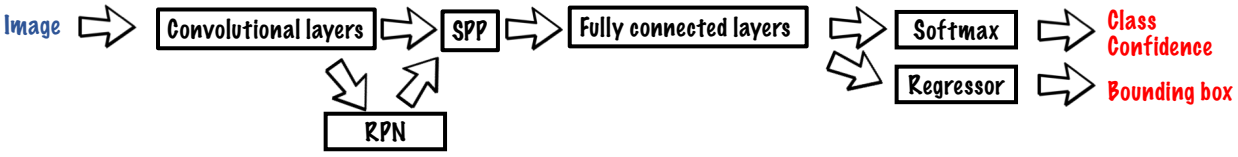
ResNets (2015, CVPR2016)



Faster R-CNN baseline	mAP@.5	mAP@.5:.95
VGG-16	41.5	21.5
ResNet-101	**48.4**	**27.2**

COCO detection results

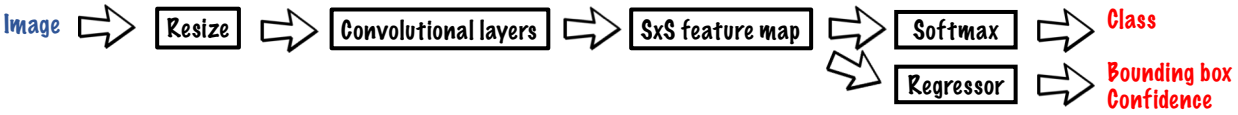
Faster R-CNN (2015, NIPS2015)



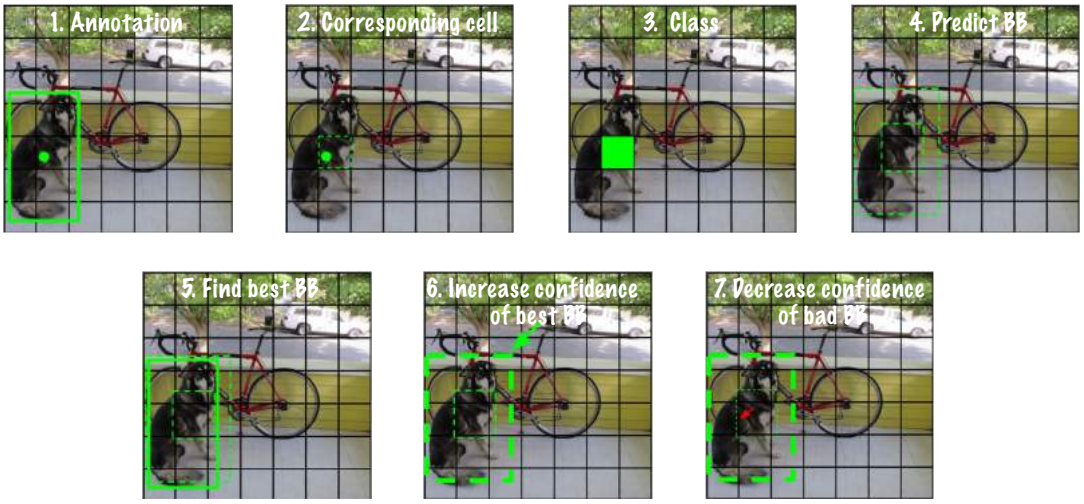
You Only Look Once

YOLO (2015, CVPR2016)

No need for region proposals.



YOLO (2015, CVPR2016): key idea

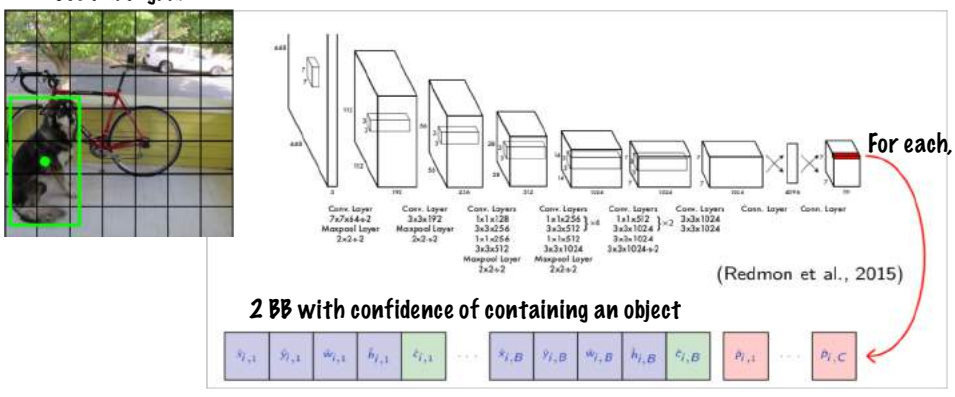


https://docs.google.com/presentation/d/1kAa7NOamBt4calBU9iHgT8a86RRHz9Yz2oh4-GTdX6M/edit#slide=id.g151008b386_0_0

YOLO (2015, CVPR2016): architecture

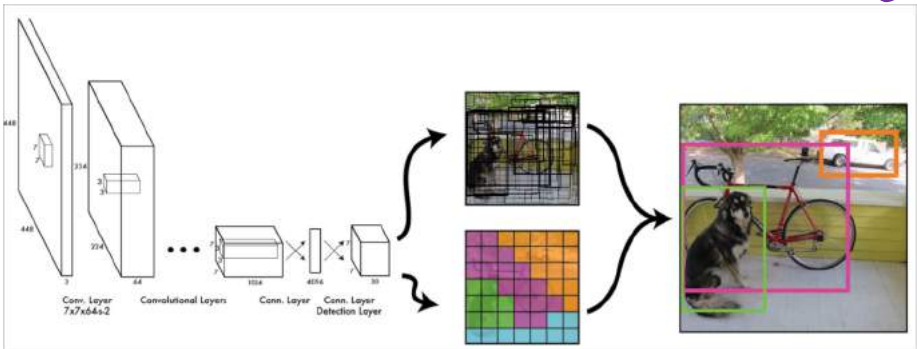
For PascalVOC we have:

Use 7 x 7 grid



Francois Fleuret : Deep Learning Course, https://fleuret.org/dlc/

YOLO (2015, CVPR2016): training



Training:

1. Pre-train network on Imagenet classification task
2. Train the model with joint loss (quite engineered loss function)

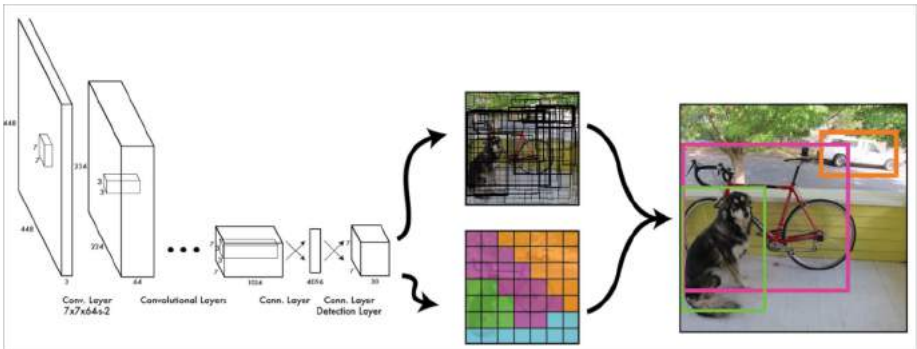
https://docs.google.com/presentation/d/1kAa7NOamBt4calBU9iHgT8a86RRHz9Yz2oh4-GTdX6M/edit#slide=id.g151008b386_0_0

YOLO (2015, CVPR2016): training tricks

1. use 448×448 input for detection, instead of 224×224,
2. use Leaky ReLU for all layers,
3. dropout after the first fully connected layer,
4. normalize bounding boxes parameters in $[0, 1]$,
5. use a quadratic loss not only for the bounding box coordinates, but also for the confidence and the class scores,
6. reduce the weight of large bounding boxes by using the square roots of the size in the loss,
7. reduce the importance of empty cells by weighting less the confidence-related loss on them,
8. use momentum 0.9, decay $5e-4$,
9. data augmentation with scaling, translation, and HSV transformation.

Francois Fleuret : Deep Learning Course, https://fleuret.org/dlc/

YOLO (2015, CVPR2016): inference

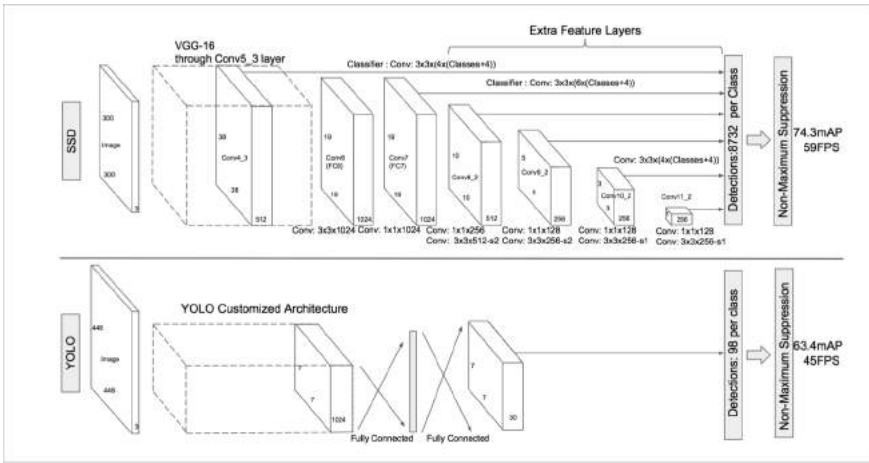


Inference:
Single pass through the network.

Inference is very fast.

https://docs.google.com/presentation/d/1kAa7NOamBt4calBU9iHgT8a86RRHz9Yz2oh4-GTdX6M/edit#slide=id.g151008b386_0_0

SSD (2015, ECCV2016)

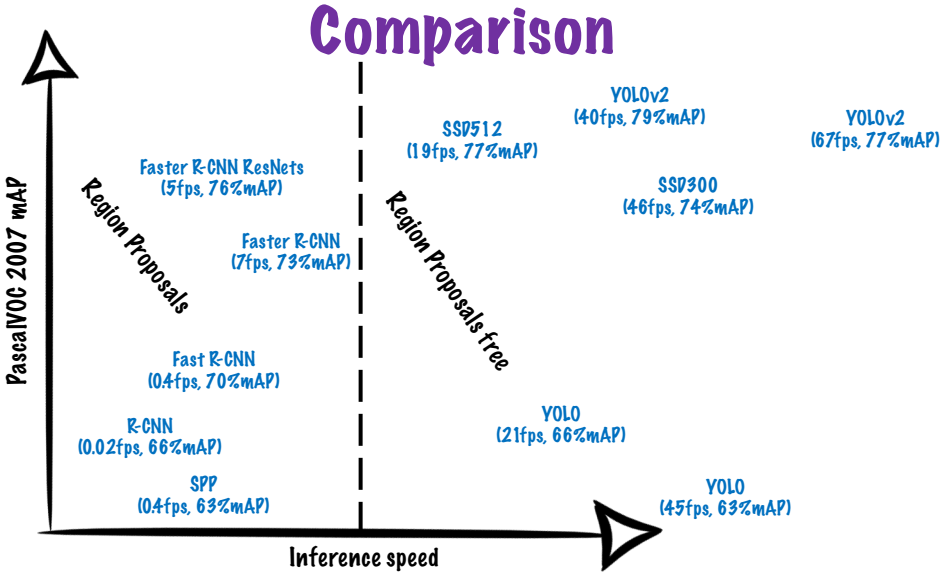


YOLOv2 (2016)

	YOLO							YOLOv2
batch norm?		✓	✓	✓	✓	✓	✓	✓
hi-res classifier?			✓	✓	✓	✓	✓	✓
convolutional?				✓	✓	✓	✓	✓
anchor boxes?				✓	✓			
new network?					✓	✓	✓	✓	✓
dimension priors?						✓	✓	✓	✓
location prediction?						✓	✓	✓	✓
passthrough?							✓	✓	✓
multi-scale?								✓	✓
hi-res detector?									✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	**78.6**

There are a lot of tricks to get a good architecture for object detection...

Comparison



Interesting papers (not covered in the class)

Feature Pyramid Networks for Object Detection

Tsung-Yi Lin[1,2], Piotr Dollár[1], Ross Girshick[1],
Kaiming He[1], Bharath Hariharan[1], and Serge Belongie[2]

[1]Facebook AI Research (FAIR)
[2]Cornell University and Cornell Tech

Focal Loss for Dense Object Detection

Tsung-Yi Lin Priya Goyal Ross Girshick Kaiming He Piotr Dollár
Facebook AI Research (FAIR)

Mask R-CNN

Kaiming He Georgia Gkioxari Piotr Dollár Ross Girshick
Facebook AI Research (FAIR)

Index

Weakly supervised localization

We have:

We want:

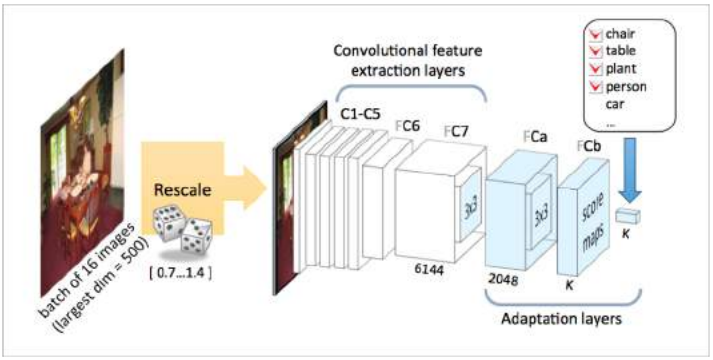
Labeling is expensive!

Is object localization for free?

Let us have a look at one paper [1]

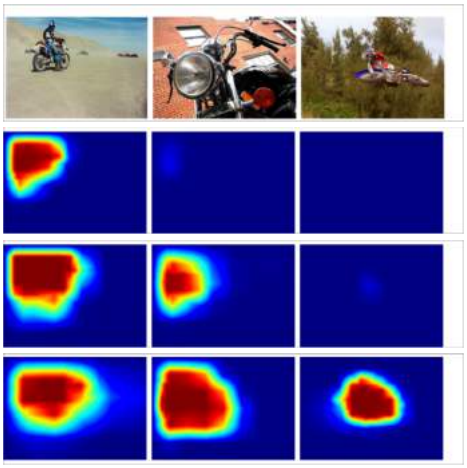
[1] http://www.di.ens.fr/~josef/publications/Oquab15.pdf

How is it done?



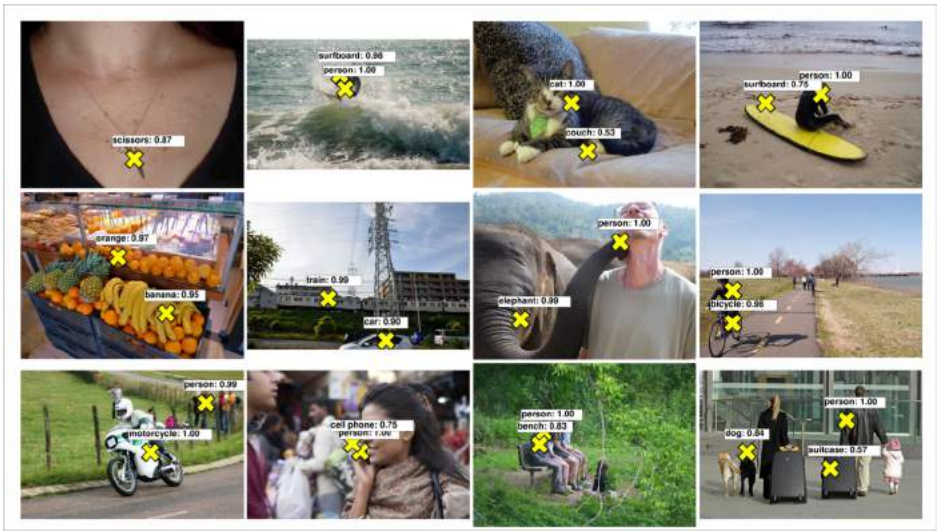
http://www.di.ens.fr/willow/research/weakcnn/

Is object localization for free?



http://www.di.ens.fr/willow/research/weakcnn/

Results



Wrap up

Datasets
Evaluation

Part 1
Introduction

Part 2
Basic blocks & concepts

Object detection pipeline
Hard Negative Mining
Non-Maximum Suppression
Region Proposals

Part 3
Models

SPP
OverFeat
R-CNN
Fast R-CNN
Faster R-CNN
YOLO
SSD

Bonus material
Weakly supervised localization

Is object localization for free?

References

- *Ross B. Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik*; Rich feature hierarchies for accurate object detection and semantic segmentation.
- *Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, Yann LeCun*; OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks.
- *Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun*; Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition.
- *Ross B. Girshick*; Fast R-CNN.
- *Shaoqing Ren, Kaiming He, Ross B. Girshick and Jian Sun*; Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.
- *Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi*; You Only Look Once: Unified, Real-Time Object Detection.
- *Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun*; Deep Residual Learning for Image Recognition.
- *Joseph Redmon and Ali Farhadi*; YOLO9000: Better, Faster, Stronger.
- *M. Oquab and L. Bottou and I. Laptev and J. Sivic*; Is object localization for free? - Weakly-supervised learning with convolutional neural networks.

Deep Learning 4

Michal Drozdzal
mdrozdzal@fb.com