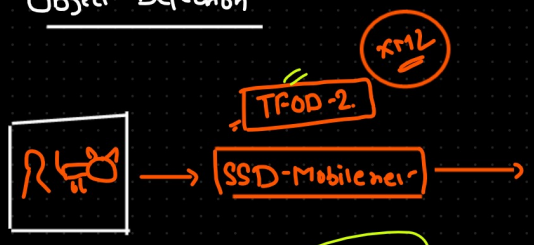
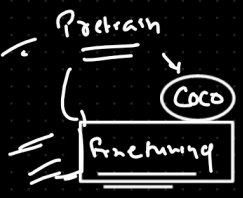
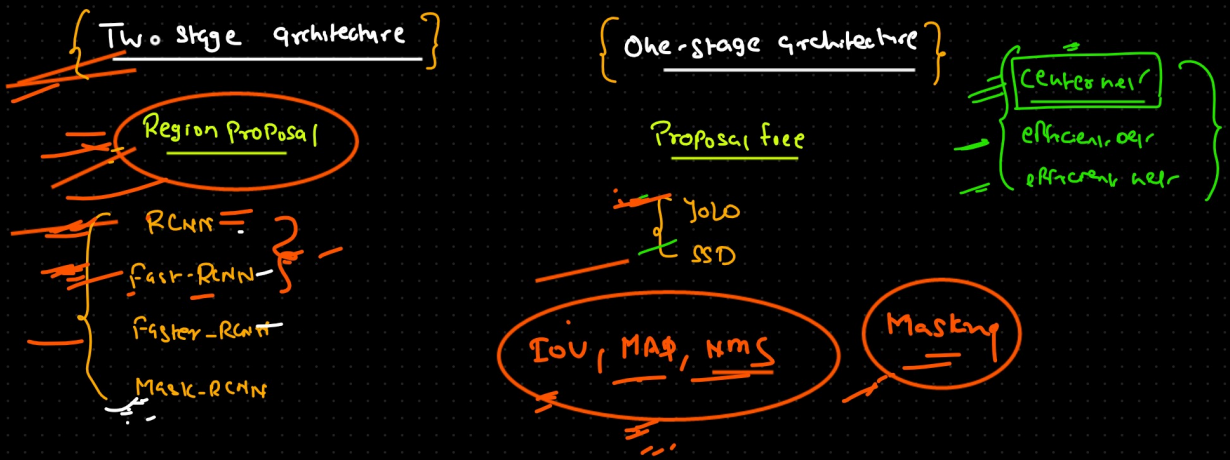
Object Detection



$Y - \hat{Y} \to 0$



1. Supervised learning (ground truth)
2. training
   - Forward Pass → Bbox (Reg), class (classification)
   - Loss → L1 + Logloss
   - Backward Pass → updating the Param
3. Prediction
   - Passing the image
   - in Real time ⇒ Detection is happening



RCNN ⇒ Region-based Convolution

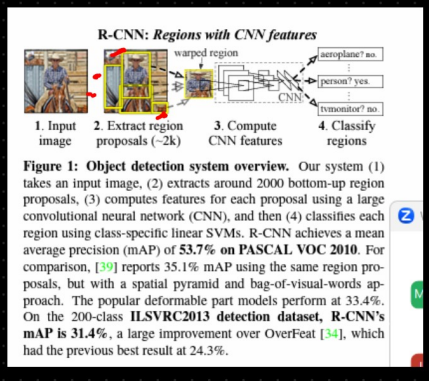
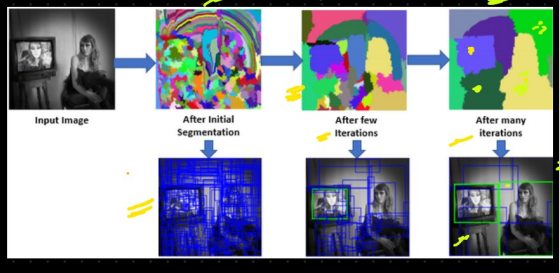


Figure 1: Object detection system overview. Our system (1) takes an input image, (2) extracts around 2000 bottom-up region proposals, (3) computes features for each proposal using a large convolutional neural network (CNN), and then (4) classifies each region using class-specific linear SVMs. R-CNN achieves a mean average precision (mAP) of **53.7% on PASCAL VOC 2010**. For comparison, [39] reports 35.1% mAP using the same region proposals, but with a spatial pyramid and bag-of-visual-words approach. The popular deformable part models perform at 33.4%. On the 200-class **ILSVRC2013 detection dataset, R-CNN's mAP is 31.4%**, a large improvement over OverFeat [34], which had the previous best result at 24.3%.

1) Image
2) extract region (2k) (Selective Search)
3) Warped region = (resize)
4) CNN + FC
5) Bbox + class → $(x_{min}, x_{max}, y_{min}, y_{max})$ sigmoid + Prob (0-1) Softmax

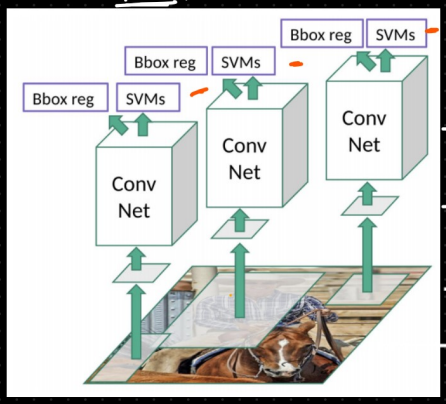
Selective Search algo

1. Color Similarity
2. Texture Sim
3. Size Sim.
4. Fill Sim.



2k → 2k regions → RP

RCNN



Forward Pass

→ classification + Bbox
(class → Prob) ↳ (Xmin, Xmax, Ymin, Ymax)

→ CNN → Flatten (1-D array) → SVM ③ (Alexnet)
②

→ warped region (resizing)

→ region extractor (Selective Search) ① → 2K Similiry

→ image

Loss

L_1 & L_2 and Log loss

Disadvantage of RCNN

Optimizer

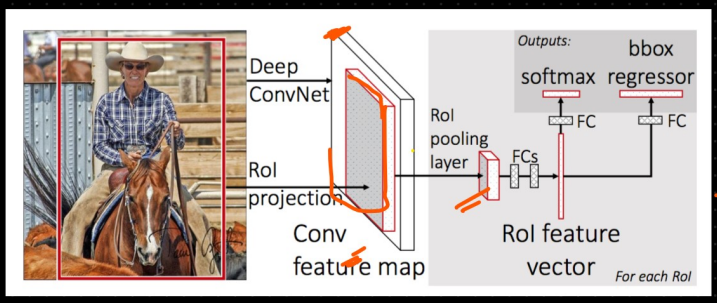
MAP, IOU | NMS Pool Pruning

↓ evaluating the class Pred.

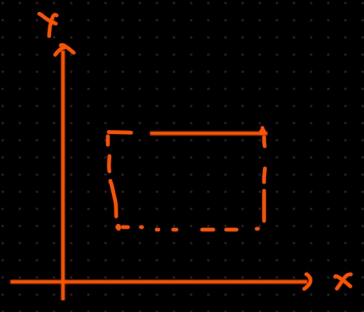
↓ evaluating Bbox

Region Proposal, RoI (Region of interest)

Fast-RCNN ⇒



1. Image
2. Kernels & filter (Conv oper)
3. feature map
4. Selective search (RoI)
5. RoI (max Pooling)
6. CNN + FC
7. Bbox, class
   (Xmin, Xmax, Ymin, Ymax) ↓ Prob



↓ flattening ↓ FC

X, Y, L, W

Bbox ← FC → Clan
↑
CNN

③ faster RCNN ⇒

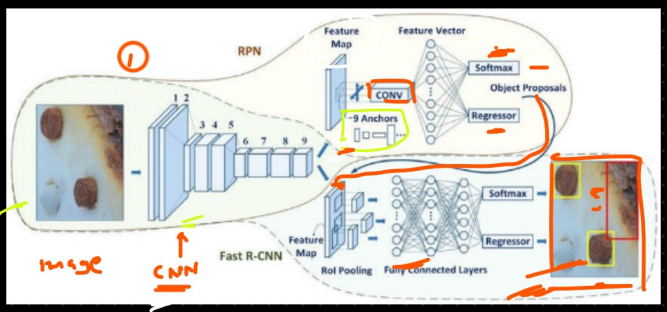
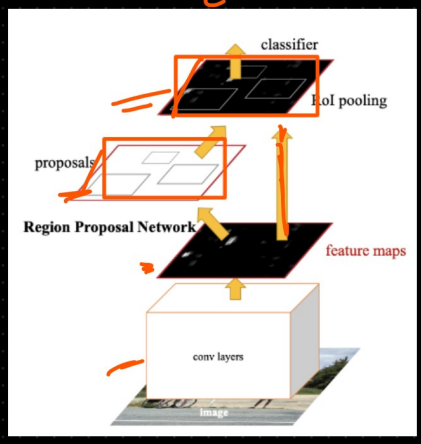
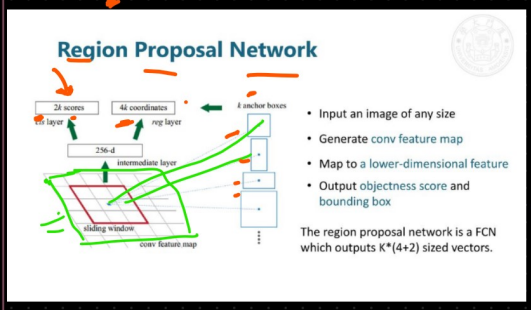


image
CNN



Y/N 2×K 4×K □



Anchor boxes

① Image
② Filter
③ Conv op.
④ FM ⇒

Possibility

□□□ ⇒ might present

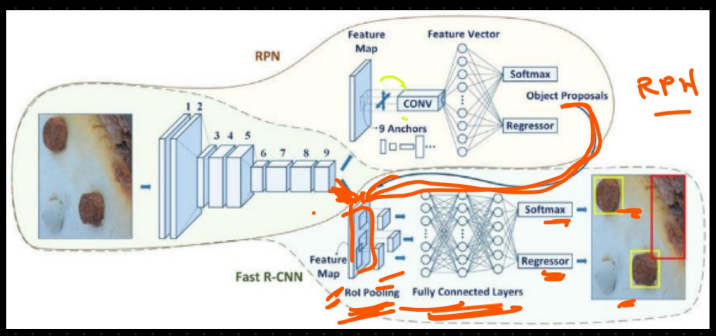
FM ← Sliding window.

K-anchor
↳ 5, 7, 10, 15, 20, 25

↳ Score. Bbox

RPN

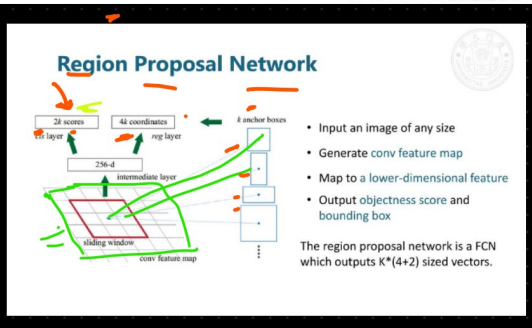
Anchor boxes = Y/N R



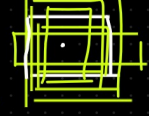
RPN

① Image
② filter × Conv op.
③ FM ⇒ RPN

Region Proposal network
↓
those region whereever object is present



④ FM ⇒ Sliding window



5
k-anchor boxes
(Bbox)

Selective Search

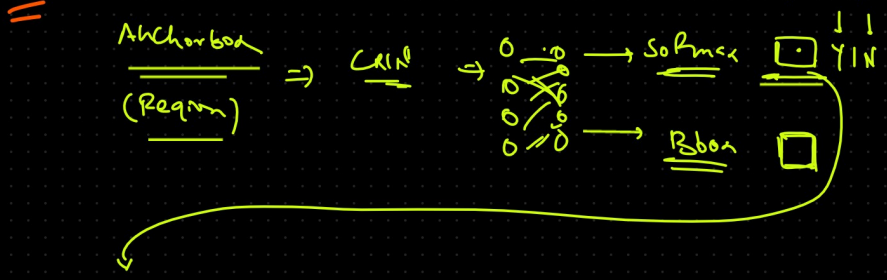
(whether obj is present or not)

Y/N

2K ⇒ 2×5

4K ⇒ 4×5

$x_{min}, x_{max}, y_{min}, y_{max}$



Region Proposal (Region where object is available)

https://stackoverflow.com/questions/50450998/anchor-box-or-bounding-boxes-in-yolo-or-faster-rcnn
https://medium.com/@codeplumber/region-proposal-network-rpn-backbone-of-faster-r-cnn-4a744a38d7f9

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://arxiv.org/pdf/1506.01497.pdf
chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://arxiv.org/pdf/1504.08083.pdf
chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://arxiv.org/pdf/1311.2524v5.pdf
https://paperswithcode.com/task/object-detection
https://github.com/tensorflow/models