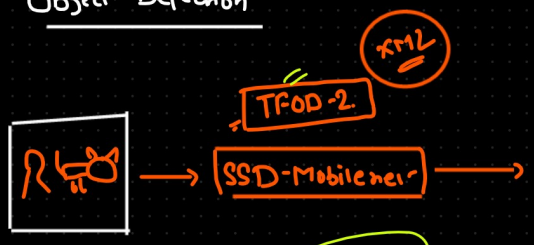
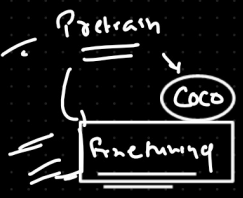
Object Detection



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



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

{Two stage architecture}

Region Proposal

- RCNN
- Fast-RCNN
- Faster-RCNN
- Mask-RCNN

{One-stage architecture}

Proposal free

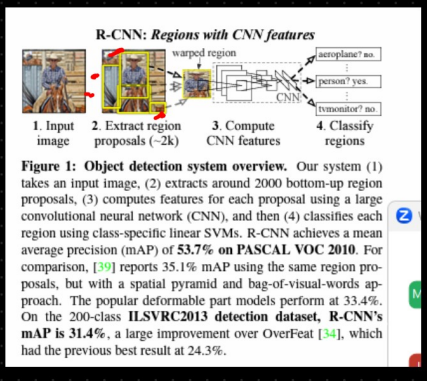
- YOLO
- SSD

IOU, MAP, NMS

Masking

- CenterNet
- efficientdet
- efficientnet

RCNN ⇒ Region-based Convolution



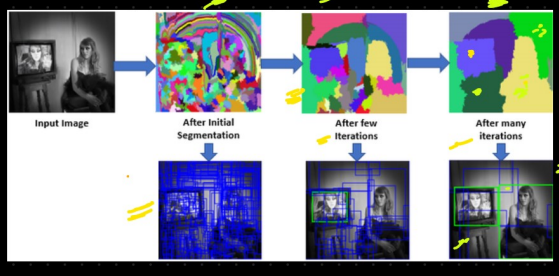
R-CNN: *Regions with CNN features*

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

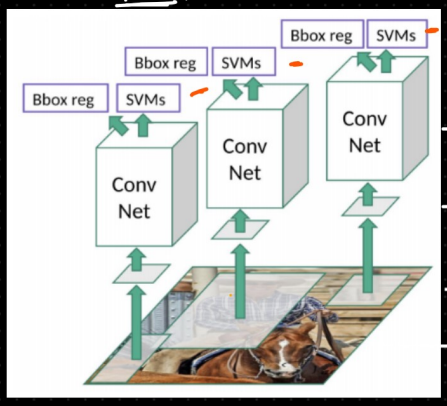
Selective Search algo

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 (4D array) → SVM ③ (Alexnet)

②

→ Warped region (resizing)

①

→ region extractor (Selective Search) → 2k Similar

→ image

Loss

L_1, L_2 and Log Loss

Disadvantage of RCNN

Optimizer

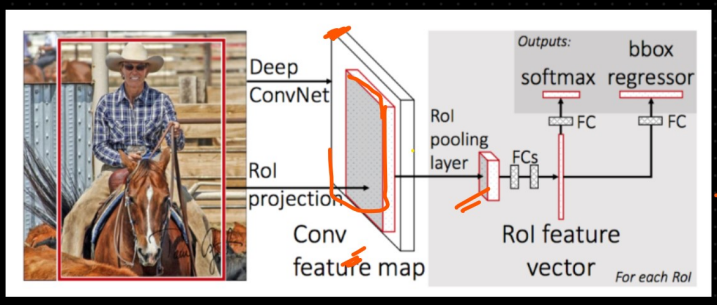
MAP, IOU — NMS — Pooling

MAP → evaluating the class pred.

IOU → evaluating Bbox

Region Proposal , RoI (Region of Interest)

FastRCNN ⇒



① Image

② Kernels & filter (Convoper)

③ feature map

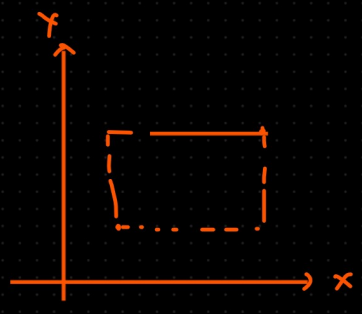
④ Selective search (RoI)

⑤ RoI (max Pooling)

⑥ CNN + FC

⑦ Bbox, class

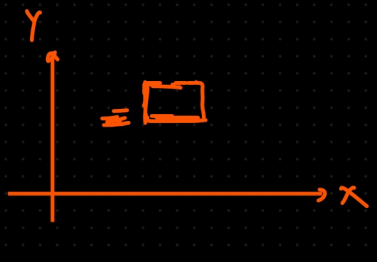
(Xmin, Xmax, Ymin, Ymax) — Prob



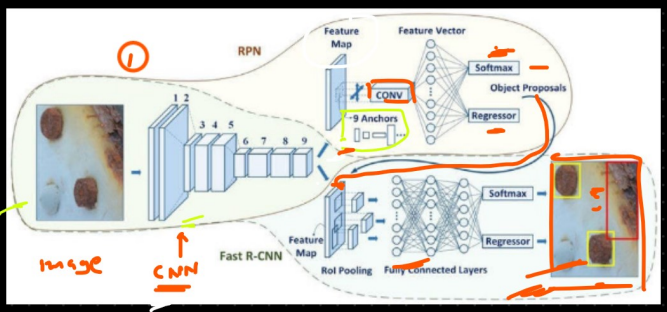
Flattening

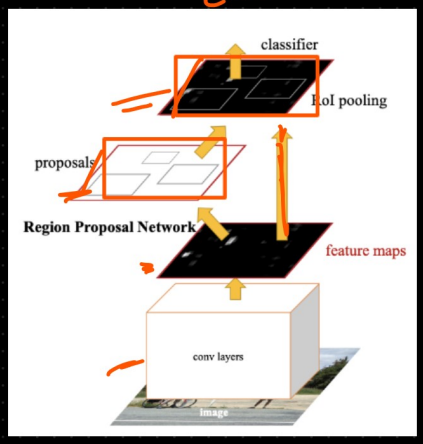
FC

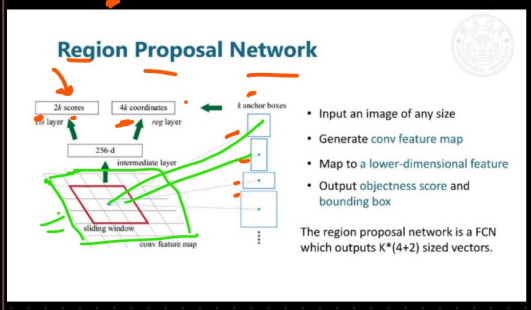
X, Y, h, w



③ faster RCNN ⇒







Anchor boxes

Possibility

⇒ mostly possible

1. Image
2. Filter
3. Conv op.
4. FM ⇒

FM ← sliding window.

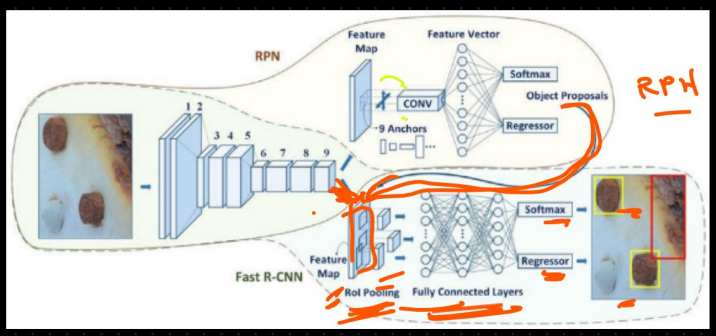
K-anchor

↳ 5, 7, 10, 15, 20, 25

↳ Score. Bbox

RPN

Anchor box → Y/N → R.

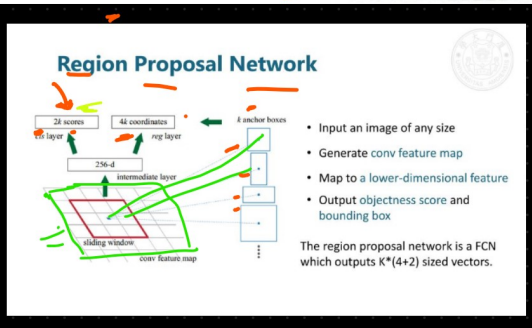


1. Image
2. filter x Conv op.
3. FM ⇒ RPN

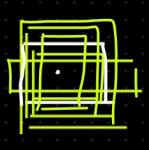
Region Proposal network

↓

those region whereever Object is possible



4. FM ⇒ sliding window



5 k-anchor boxes (Bbox)

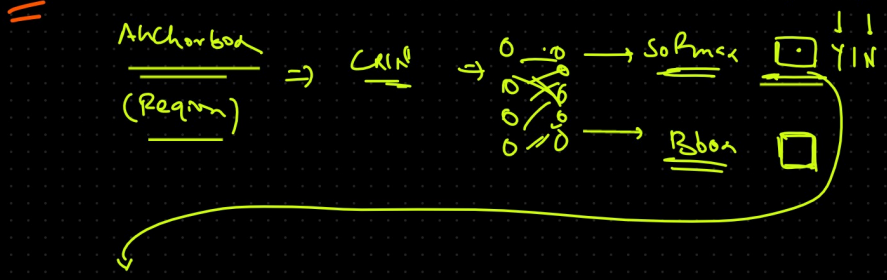
Selective Search

(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