Object Detection using Deep Learning

Dr. Rakesh Kumar Sanodiya

Indian Institute of Information Technology Sri City
rakesh.pcs16@gmail.com

CSE Department, IIIT Sri City
April 19, 2024

Roadmap

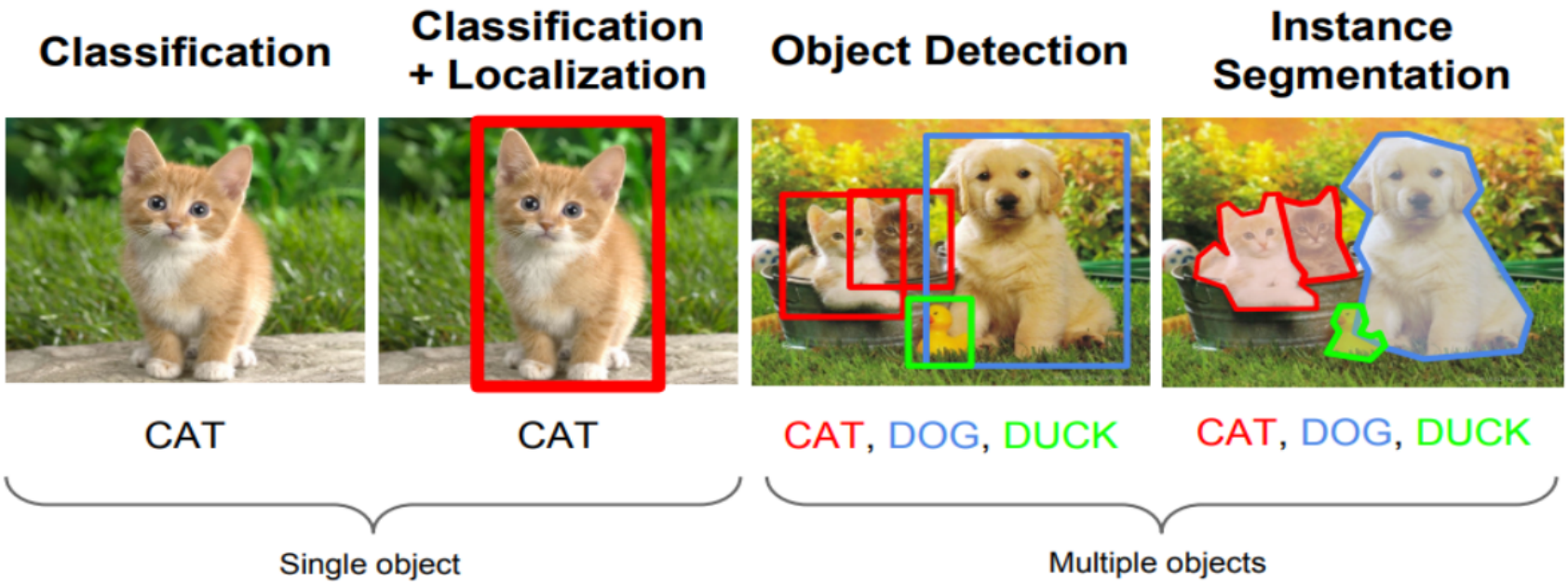


Figure: Source:[1]

[1] A. Karpathy

Object Category Detection

- Focus on object search: "Where is it?"
- Build templates that quickly differentiate object patch from background patch

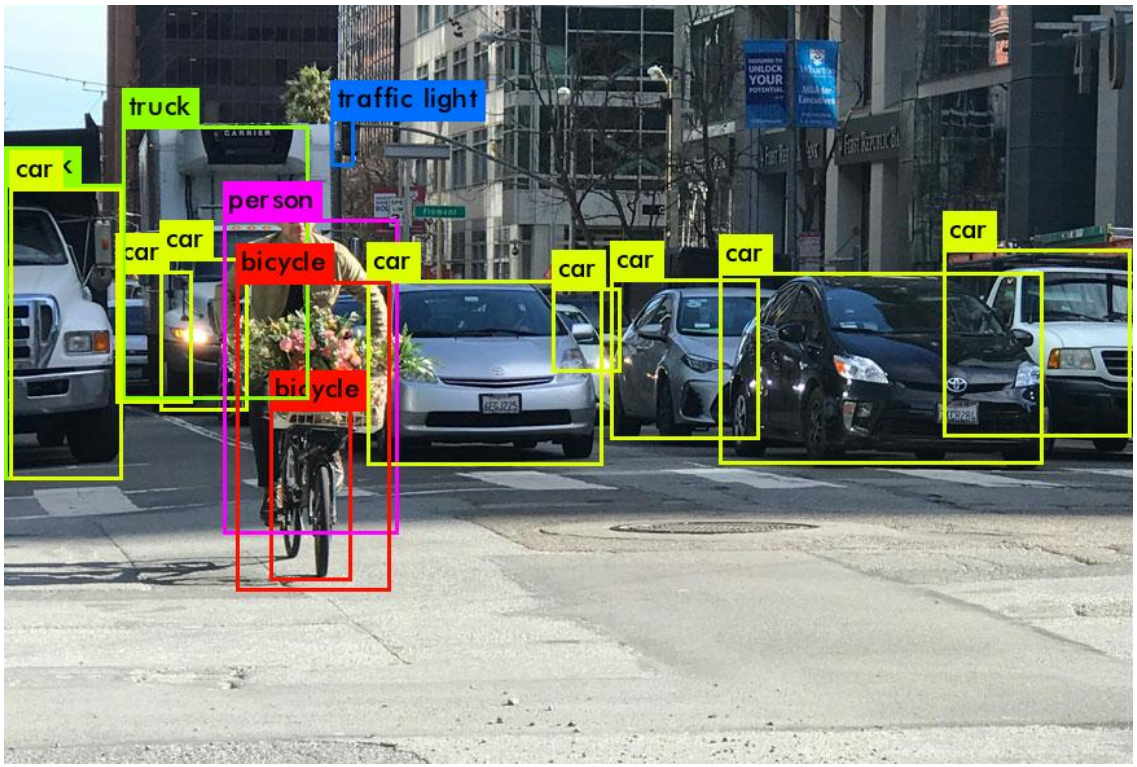
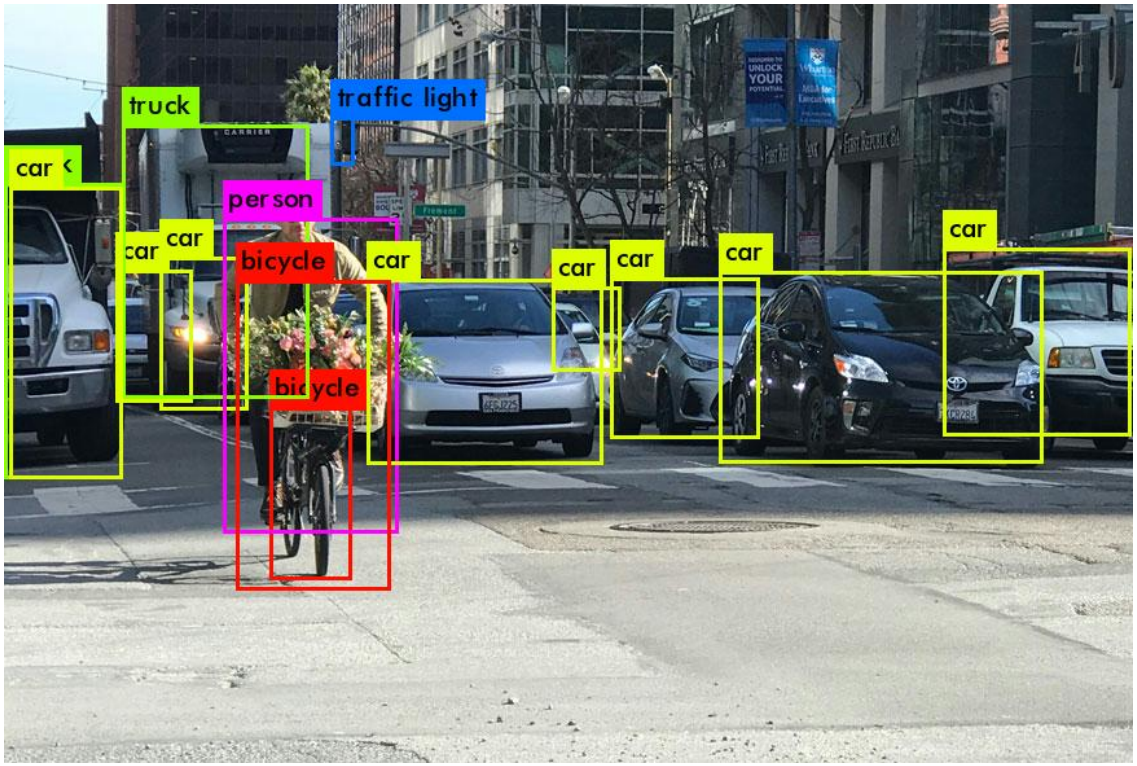


Figure: Source:[2]

[2] A. Karpathy

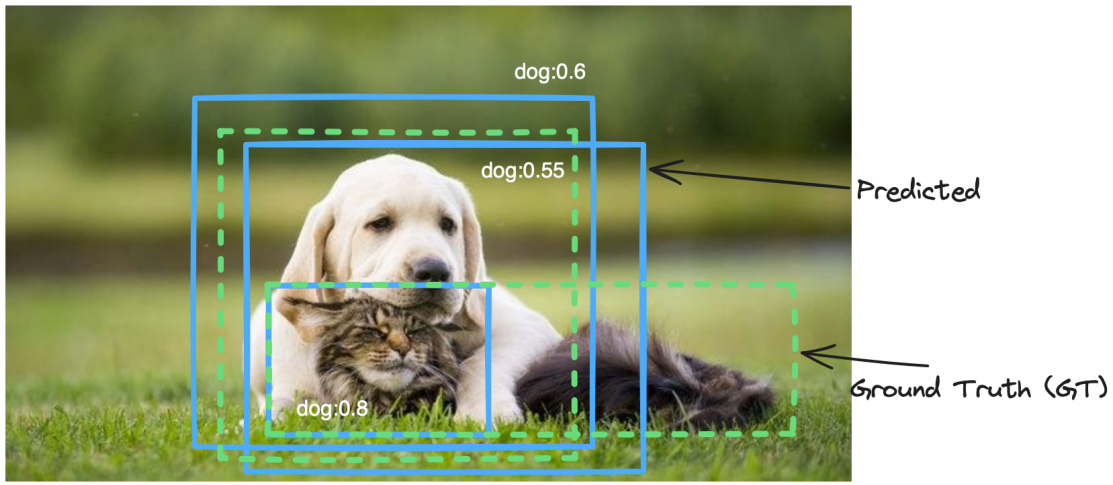
What are the challenges of object detection

- Images may contain more than one class, multiple instances from the same class
- Bounding box localization
- Evaluation



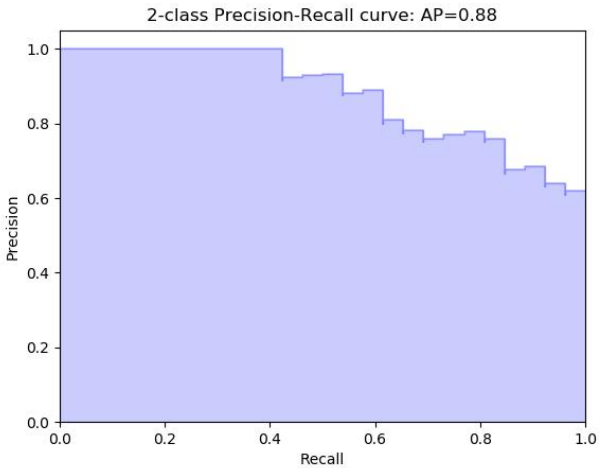
Object detection evalution

- At test time, predict bounding boxes, class labels,and confidence scores
- For each detection, determine whether it is true or false positive
  - PASCAL criterion: AREA(GT$\cap$Det)/Area (GTU Det)$>$0.5
  - For multiple detections of the same ground truth box, only one is considered a true positive



Object detection evaluation

- At test time, predict bounding boxes, class labels, and confidence scores
- For each detection, determine whether it is a ture or false positive
- For each class, sort detections from highest to lowest confidence, plot Recall-Precision curve and compute Average Precision (Area under the curve)
- Take mean of AP over classes to get mAP

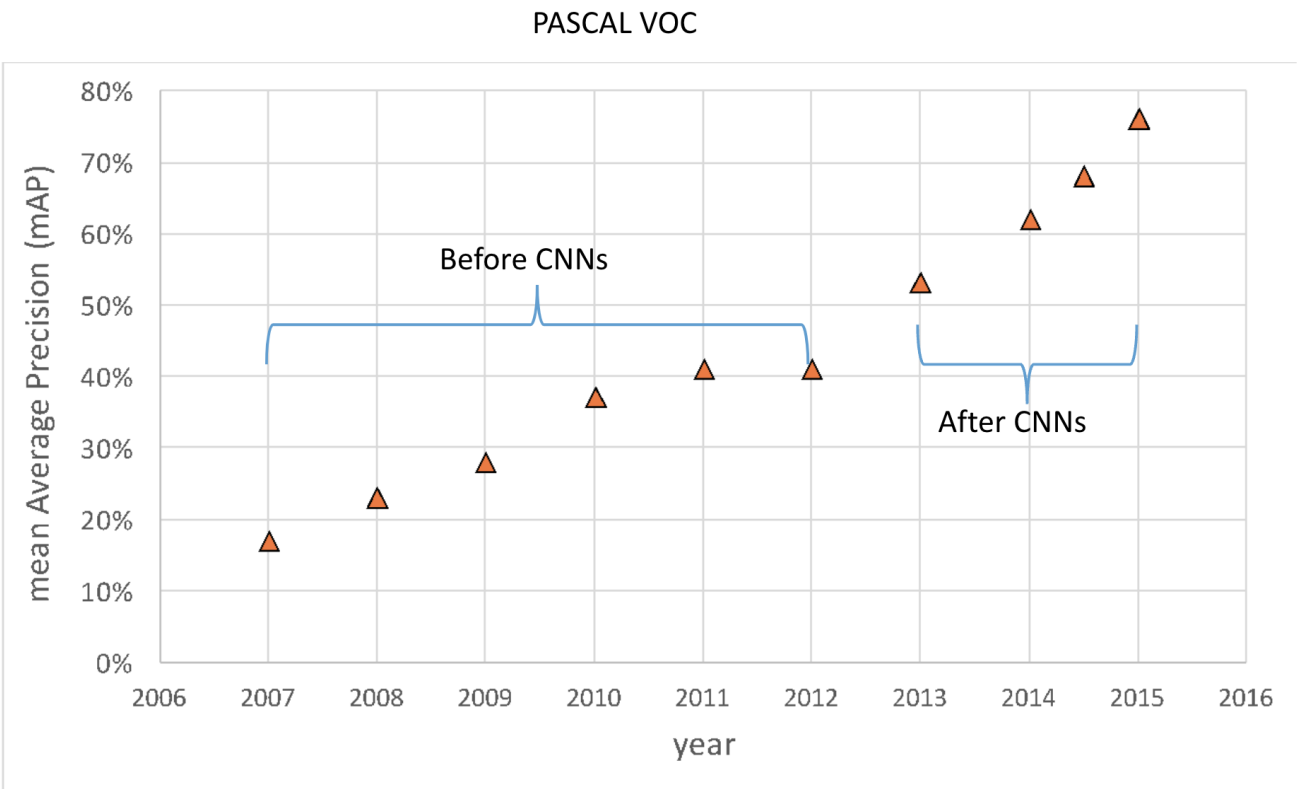


- **Precision:**true positive detections/ total detections
- **Recall:** true positive detections/ total positive test instances

PASCAL VOC challenge (2005-2012)

- 20 challenge classes:
  - Persons
  - **Animals**:bird,cat,cow, dog, horse, sheep
  - **Vehicles**:airplane, bicycle, boat, bus, car, motorbike, train
  - **indoor**:bottle, chair, dining table, potted plant, sofa, tv/monitor
- Dataset size (by 2012):11.5K training/validation images, 27K bounding boxes, 7K segmentations

Progress on PASCAL detection



Current benchmark:COCO

What is COCO?

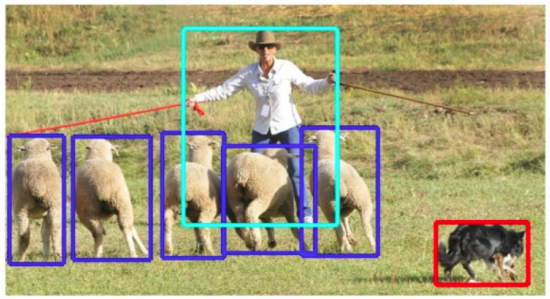
COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- Object segmentation
- Recognition in context
- Superpixel stuff segmentation
- 330K images (>200K labeled)
- 1.5 million object instances
- 80 object categories
- 91 stuff categories
- 5 captions per image
- 250,000 people with keypoints

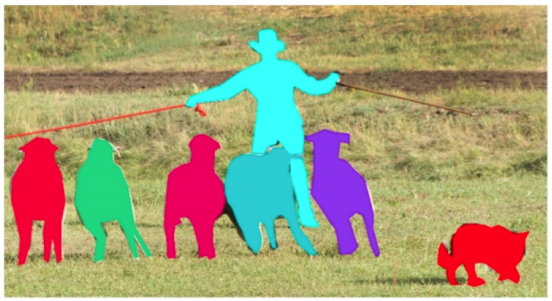
COCO
Common Objects in Context

COCO dataset: Tasks

image classification


object detection


semantic segmentation


instance segmentation

- Also:
  - keypoint prediction,
  - captioning,
  - question answering
  - …

- Leaderboard:http://cocodataset.org/detection-leaderboard
- Official COCO challenges no longer include detection
  - Emphasis has shifted to instance segmentation and dense semantic segmentation

COCO detection metrics

Average Precision (AP):

- AP % AP at IoU=.50:.05:.95 **(primary challenge 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):

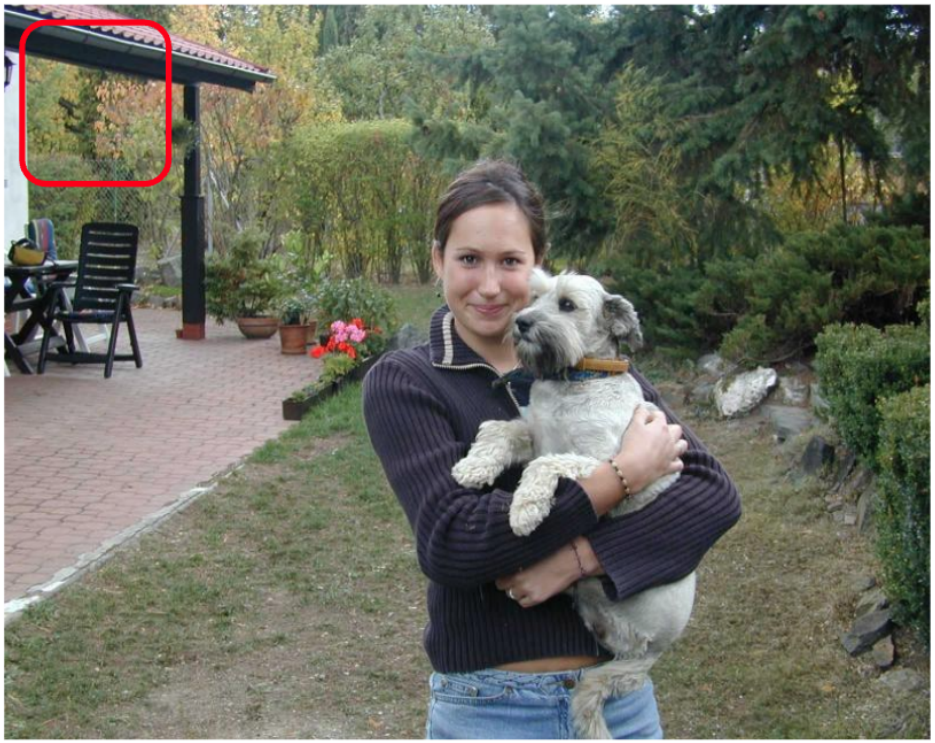
- $AR^{max=1}$ % AR given 1 detection per image
- $AR^{max=10}$ % AR given 10 detections per image
- $AR^{max=100}$ % AR given 100 detections per image

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

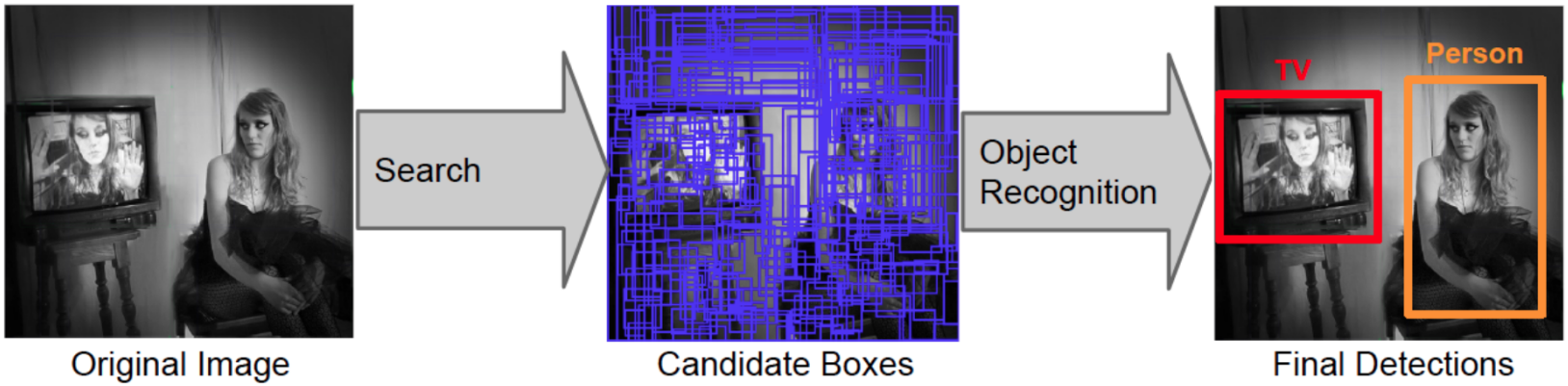
- Leaderboard:http://cocodataset.org/detection-leaderboard
- Official COCO challenges no longer include detection
  - Emphasis has shifted to instance segmentation and dense semantic segmentation

Approaches to detection:Sliding windows



- Slide a window across the image and evaluate a detection model at each location
  - Thousands of windows to evaluate:efficiency and low false positive rates are essential
  - Difficult to extend to a large range of scales, aspect ratios

Approaches to detection:Object proposals



- Generate and evaluate a few hundred region proposals
  - Proposal mechanism can take advantage of low-level perceptual organization cues
  - Proposal mechanism can be category-specific or cateogry-independent, hand-crafted or trained
  - Classifier can be slower but more powerful

Selective search for detection

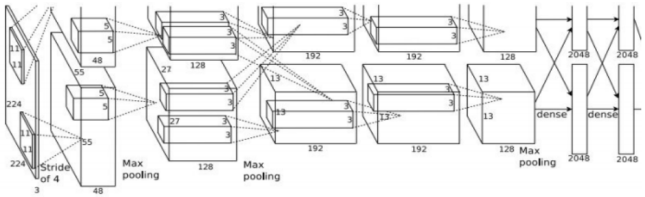
- Use Hierarchical segmentation: start with small superpixels and merge based on diverse cues

Approaches to detection

- Before 2010, dominated by sliding windows
- 2010-2013: proposal-driven
- Deep learning approaches started as proposal-driven, but have evolved back toward sliding windows
- Most recently, ”global” method are becoming more common

CNN as feature extractor

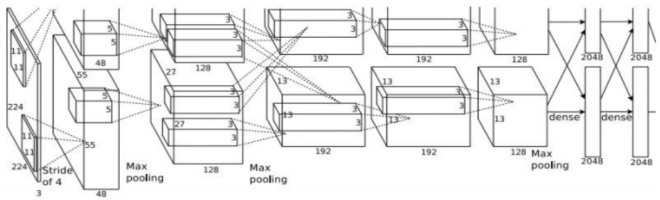
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? NO
Background? YES

CNN as feature extractor

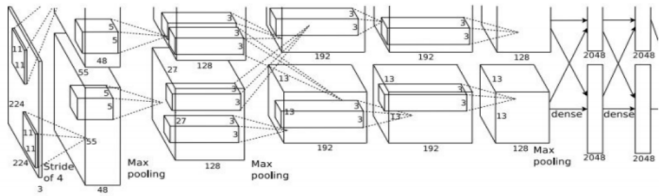
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

CNN as feature extractor

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? YES
Background? NO

CNN as feature extractor

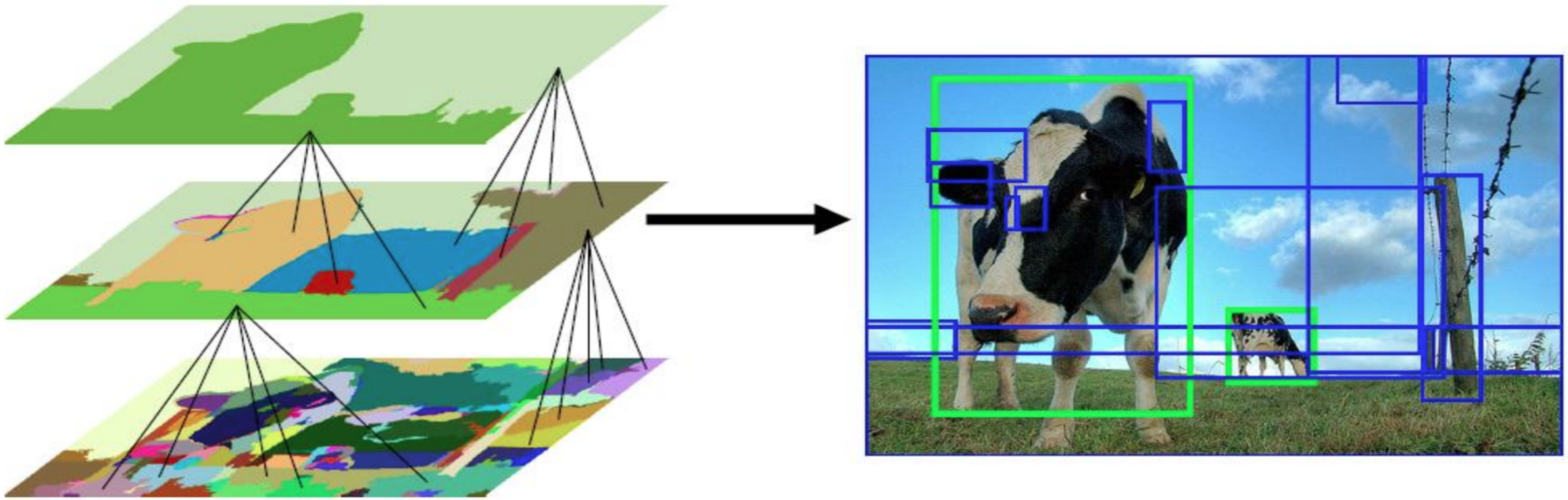
- What could be the problems?
  - Suppose we have a 600$\times$600 image, if sliding window size is 20$\times$20, then have (600-20+1) $\times$(600-20+1)= 330,200 windows

CNN as feature extractor

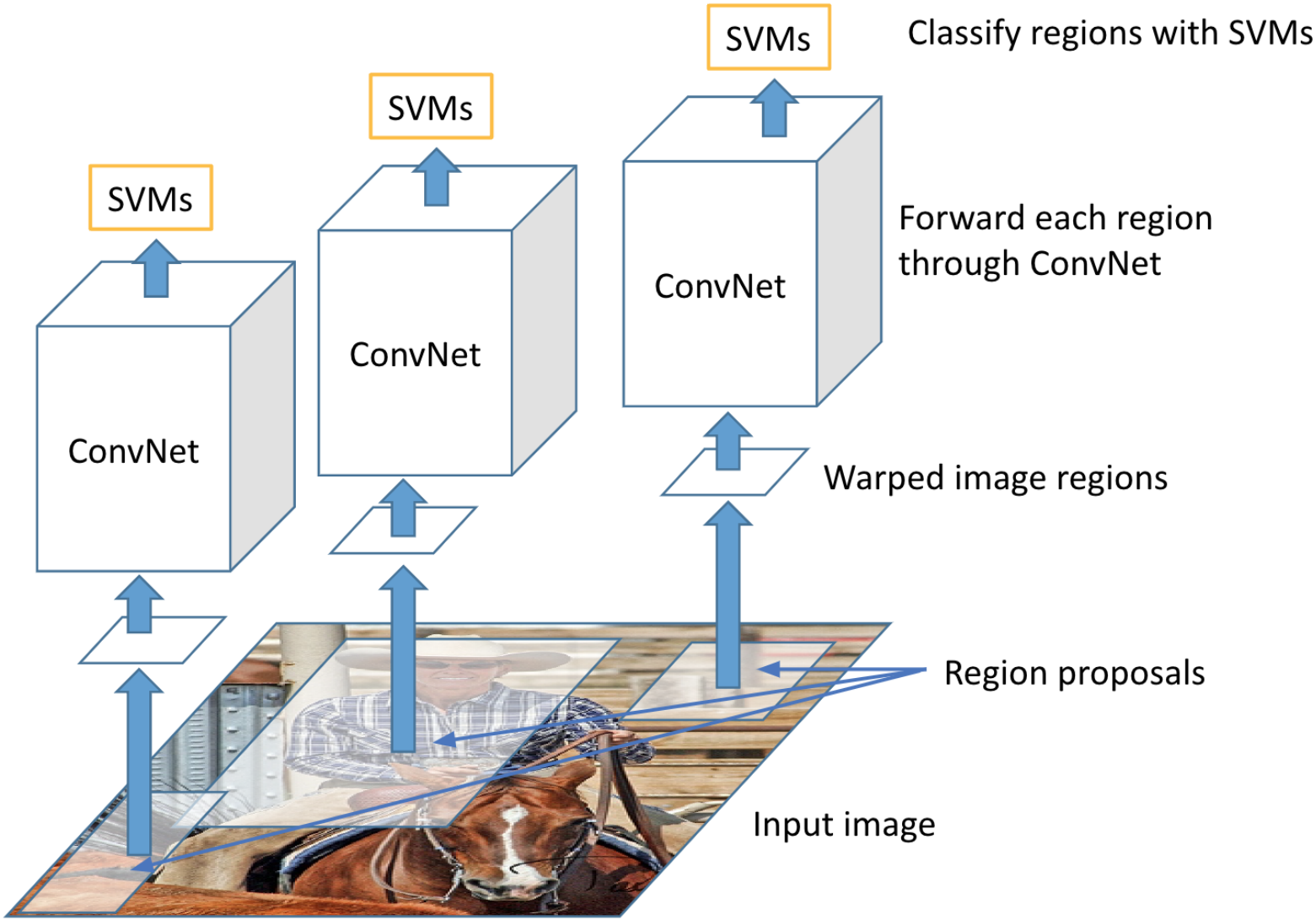
- What could be the problems?
  - Suppose we have a 600$\times$600 image, if sliding window size is 20$\times$20, then have (600-20+1) $\times$(600-20+1)= 330,200 windows
  - sometimes we want to have more accurate results-¿ multi-scale detection
    - Resize image
    - Multi-scale sliding window
- For each image, we need to do the forward pass in the CNN for 330,000 times. $->$ slow!!!

Region Proposal

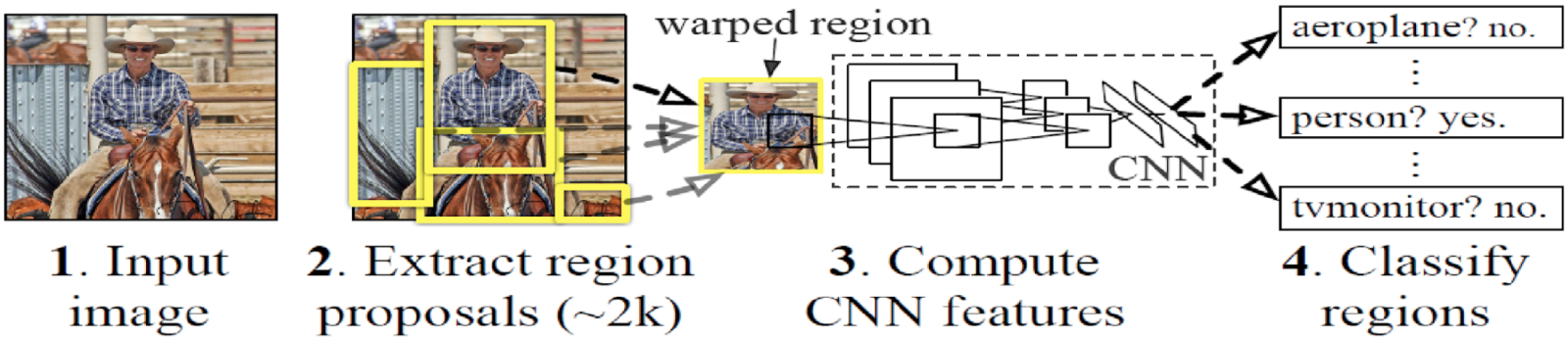
- Solution
  - Use some fast algorithms to filter out some regions first, only feed the potential region (region proposals) into CNN
  - E.g. Selective Search



R-CNN:Region Proposal +CNN features

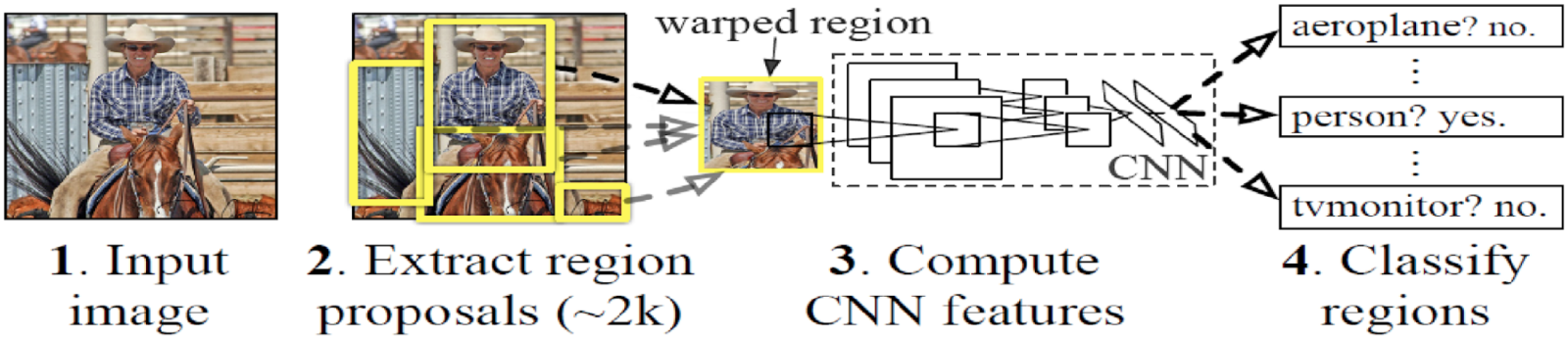


R-CNN(Girshick et al. CVPR 2014)



- Replace sliding windows with ”selective search” region proposals (Uijilings et al. IJCV 2013)
- Extract rectangles around regions and resize to 227×227
- Extract features with fine-tuned CNN (that was initialized with network trained on ImageNet before training)
- Classify last layer of network features with SVM, refine bounding box localization (bbox regression) simultaneously

R-CNN(Girshick et al. CVPR 2014)



- **Regions:** 2000 Selective Search proposals
- **Network:** AlexNet pre-trained on ImageNet (1000 classes), fine-turned on PASCAL (21 classes)
- **Final detector**: warp proposal regions, extract fc1 network activation (4096 dimensions), classify with linear SVM
- **Bounding box regressions** to refine box locations
- **Performance:**mAP of **53.7%** on PASCAL 2010 (vs. **35.1 %** for selective search and textbf33.4 % for Deformable Part Models

R-CNN pros and cons

- Pros:
  - Much more accurate than previous approaches!
  - Any deep architecture can immediately be ”plugged in”
- Cons:
  - Not a single end -to-end system
    - Fine-tune network with softmax classifier (log loss)
    - Train post-hoc linear SVMs (hinge loss)
    - Train post-hoc bounding -box regressions (least squares)
  - Training was slow (84h), took up a lot of storage
    - 2000 CNN passes per image
  - Inference (detection) was slow (47s/image with VGG16

Bounding Box Regression

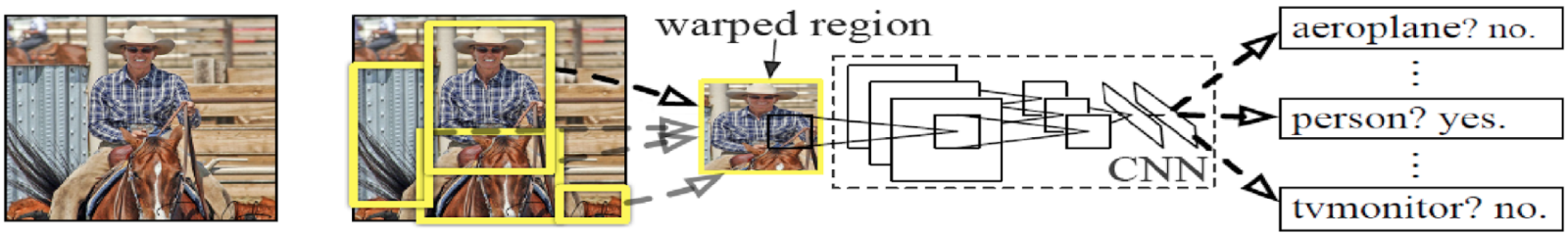
- Intuition
  - If you observe part of the object, according to the seen examples, you should be able to refine the localization
  - E.g. given the red box below, since you've seen many airplanes, you know this is not a good localization, you will adjust it to the green one

Bounding Box Regression

- Intuition
  - If you observe part of the object, according to the seen examples, you should be able to refine the localization
  - E.g. given the red box below, since you've seen many airplanes, you know this is not a good localization, you will adjust it to the green one

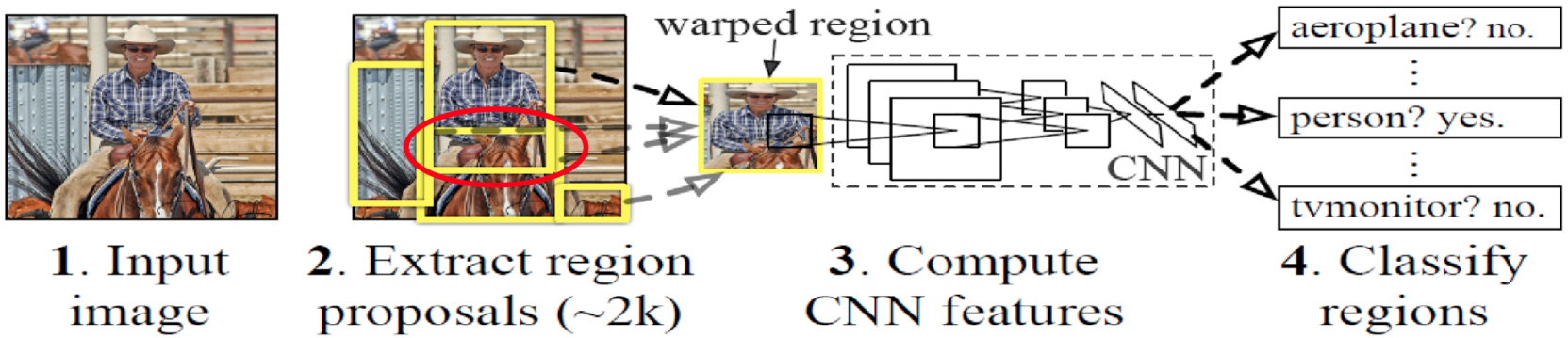
R-CNN(Girshick et al. CVPR 2014)

- What could be the problems?



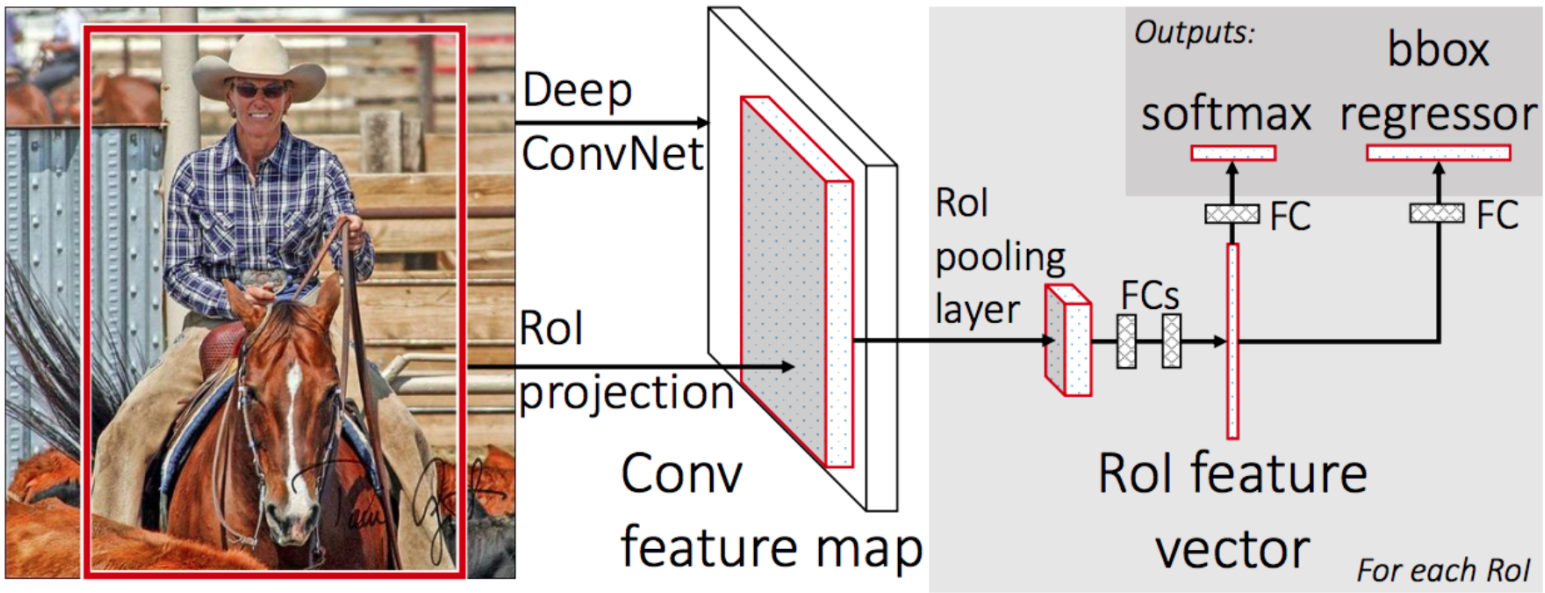
R-CNN(Girshick et al. CVPR 2014)

- What could be the problems?
  - Repetitive computation! For overlapping regions, we feed it multiple times into CNN



Fast R-CNN (Girshick ICCV 2015)

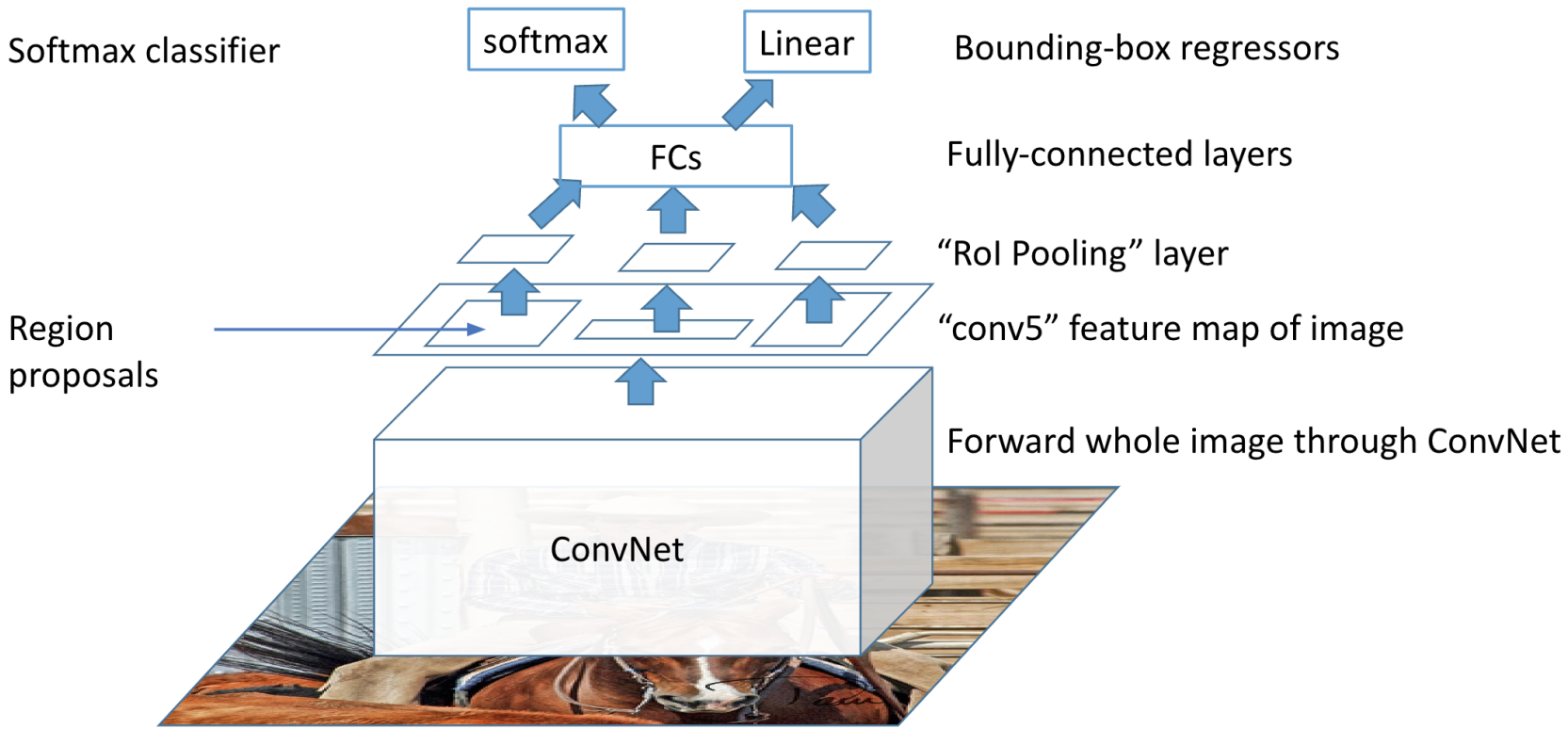
- Solution
  - Why not feed the whole image into cNN only once!. Then crop features instead of image itself



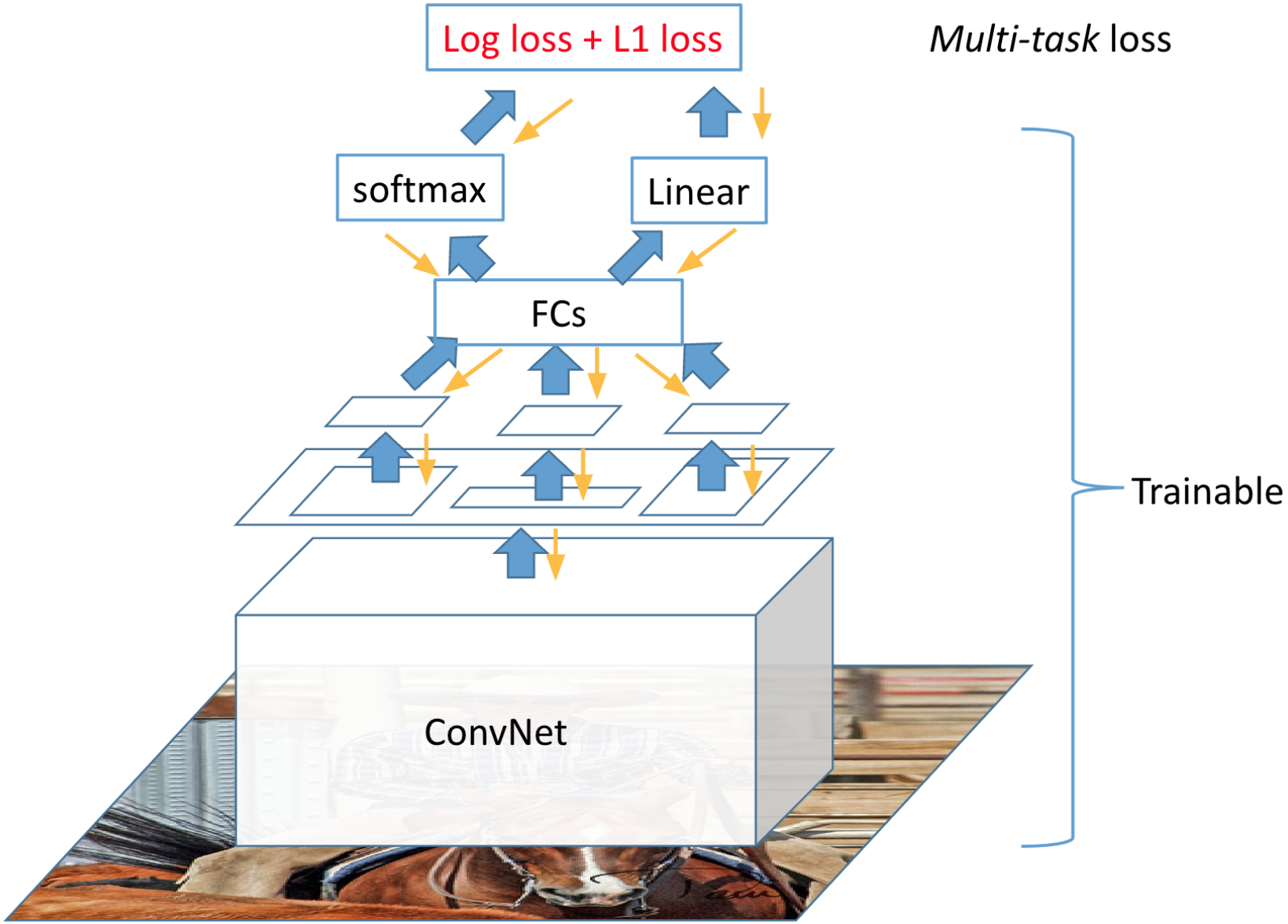
- For each RoI, network predicts probabilities for C+1 classes (class 0 is background) and four bounding box offests for C classes

Fast R-CNN (Girshick ICCV 2015)

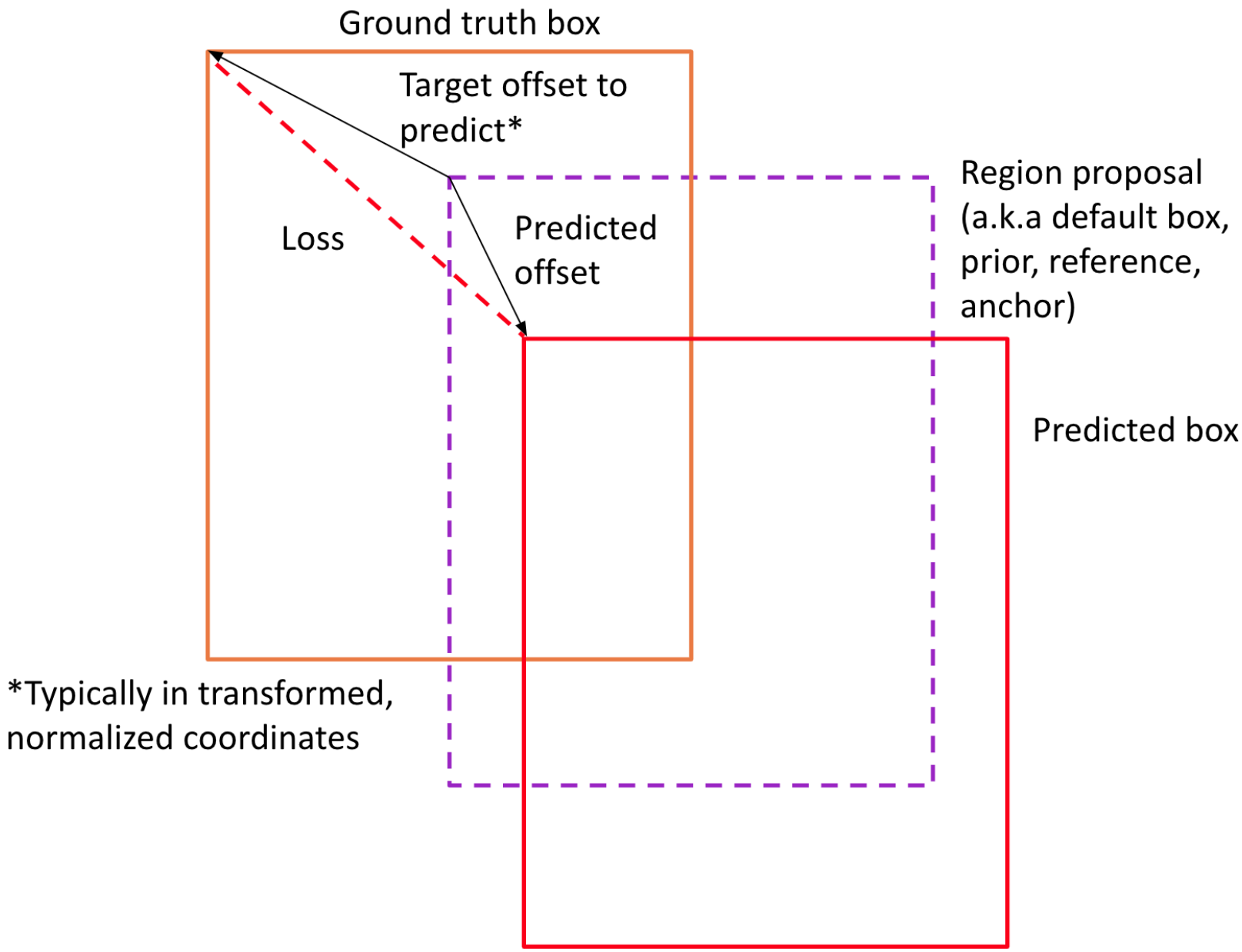
Rather than using post-hoc bounding-box regressions, bounding-box regression is implemented as an additional liner layer in the network



Fast R-CNN (Girshick ICCV 2015)



Bounding box regression



Fast R-CNN results

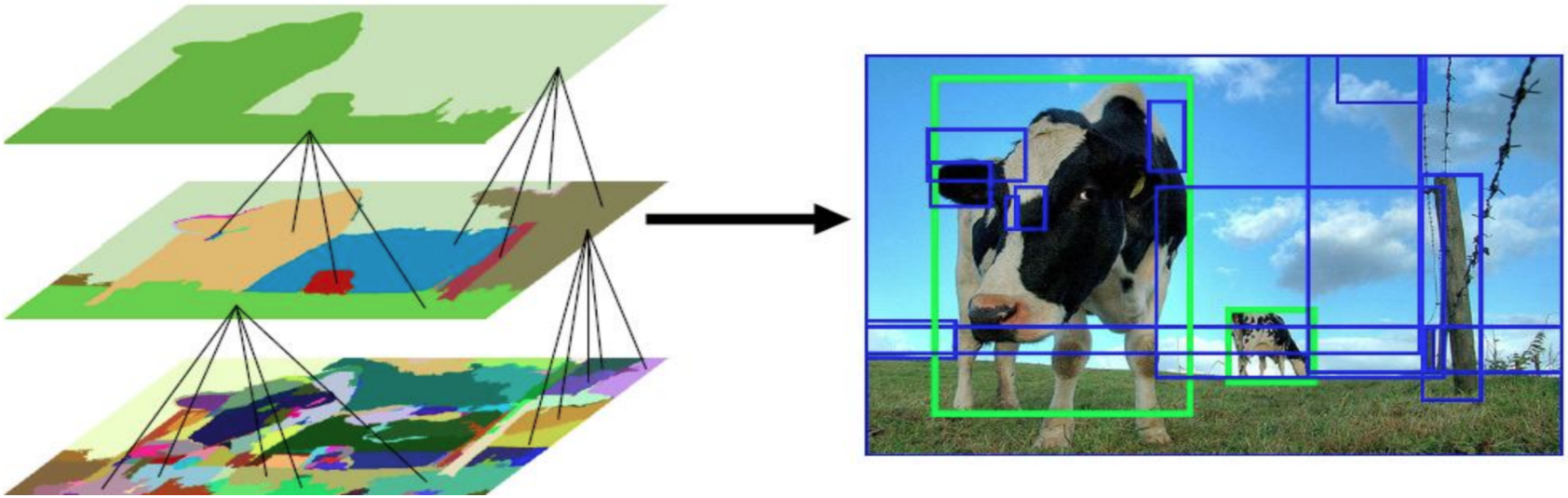
	Fast R-CNN	R-CNN
Train time (h)	**9.5**	84
- Speedup	**8.8x**	
Test time / image	**0.32s**	47.0s
- Test speedup	**146x**	
mAP	**66.9%**	66.0%

Fast R-CNN (Girshick ICCV 2015)

- What could be the problems?

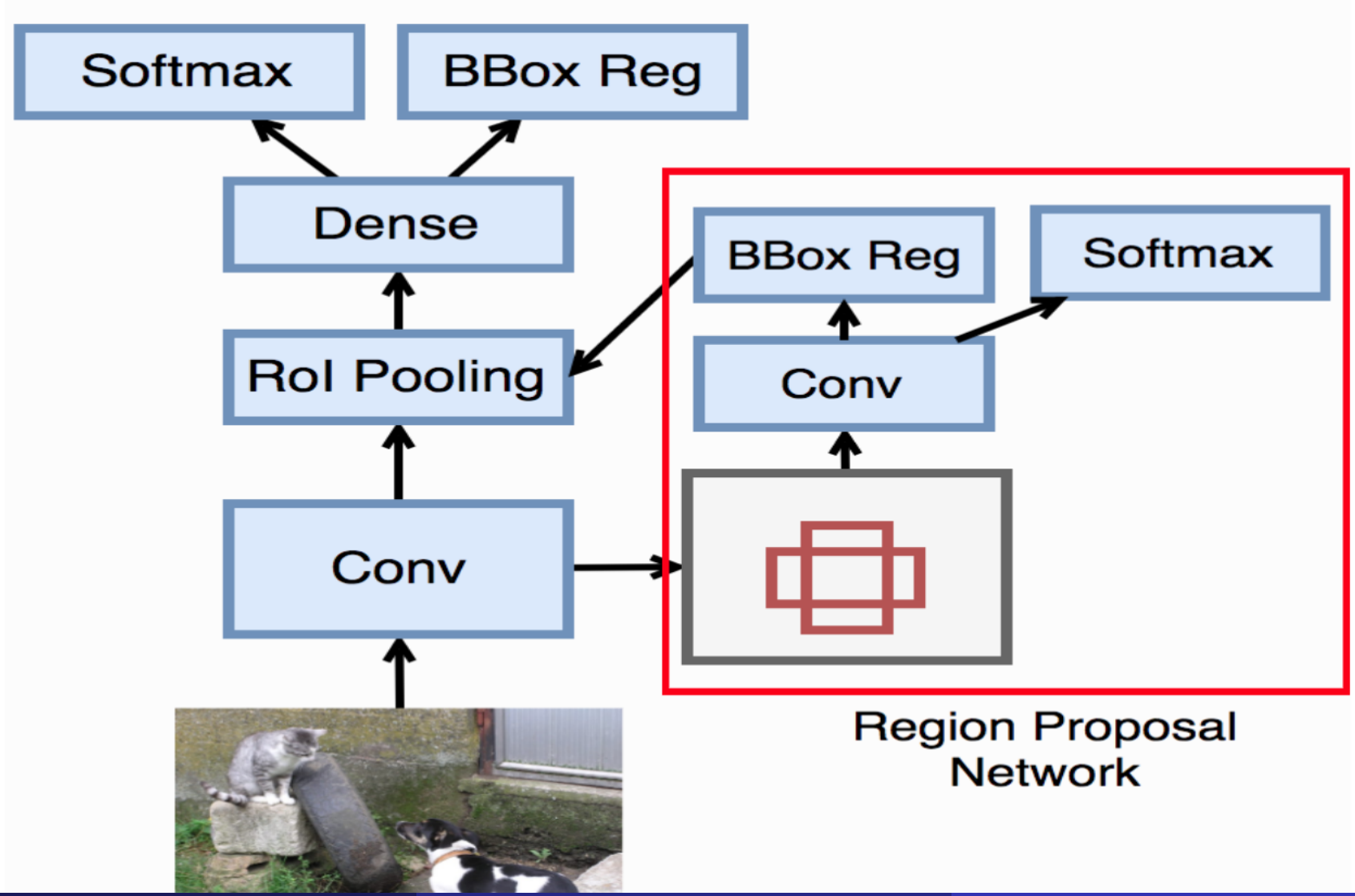
Fast R-CNN (Girshick ICCV 2015)

- What could be the problems?
  - Why we need the region proposal pre-processing step? That's not "deep learning" at all. Not cool!

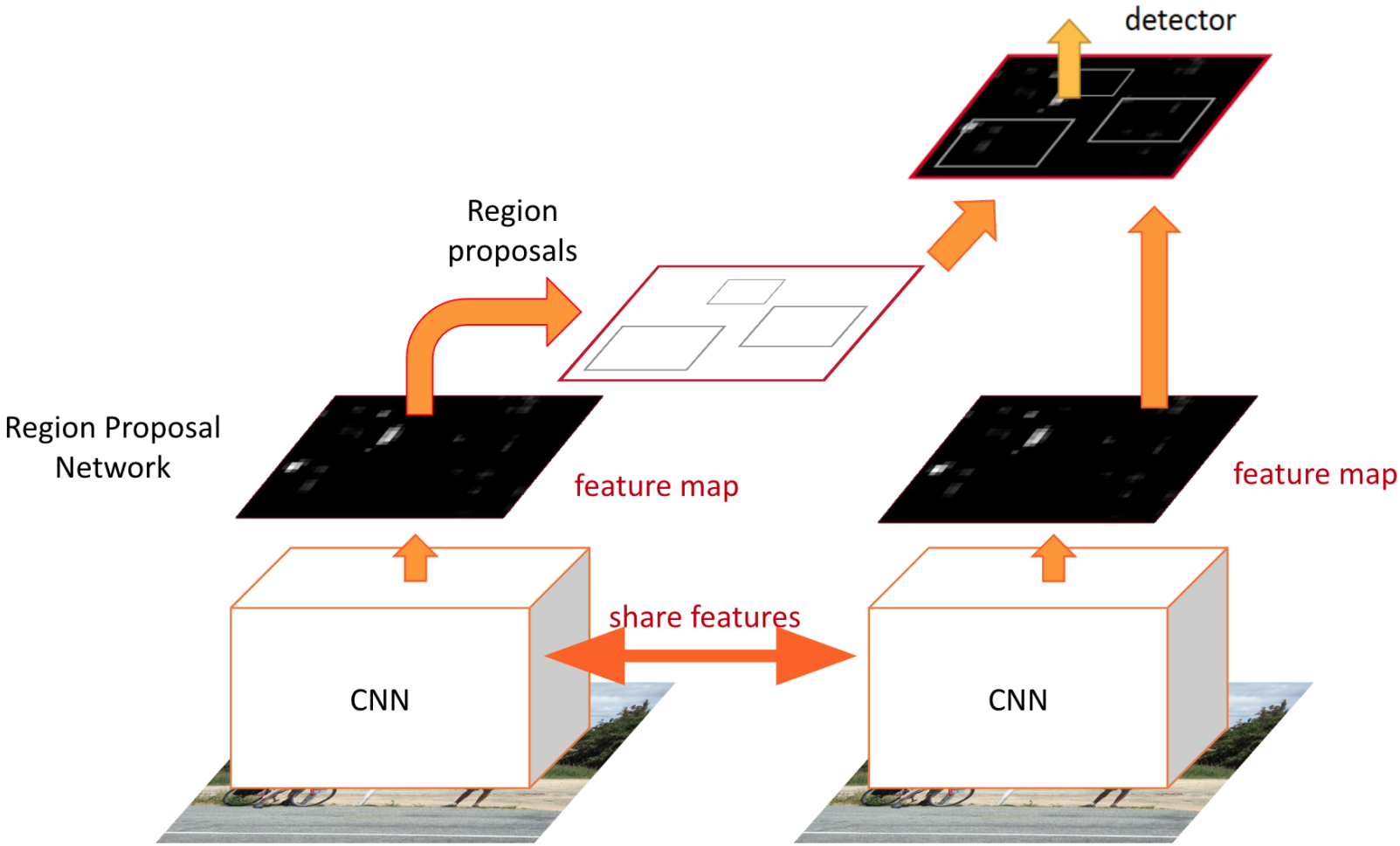


Faster R-CNN (Ren et al. NIPS 2015

- Solution
  - Why not generate region proposals using CNN??

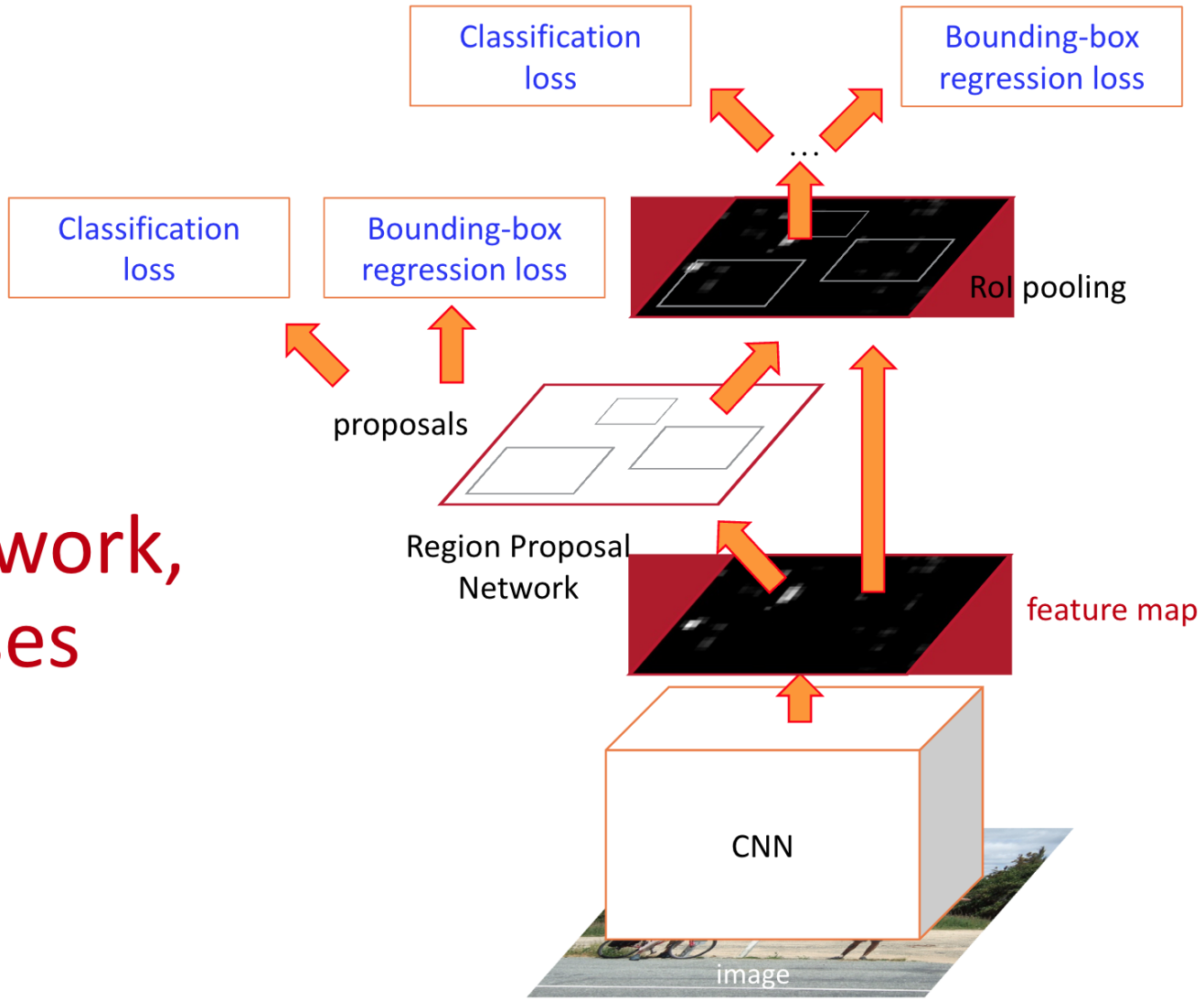


Faster R-CNN (Ren et al. NIPS 2015



Faster R-CNN (Ren et al. NIPS 2015

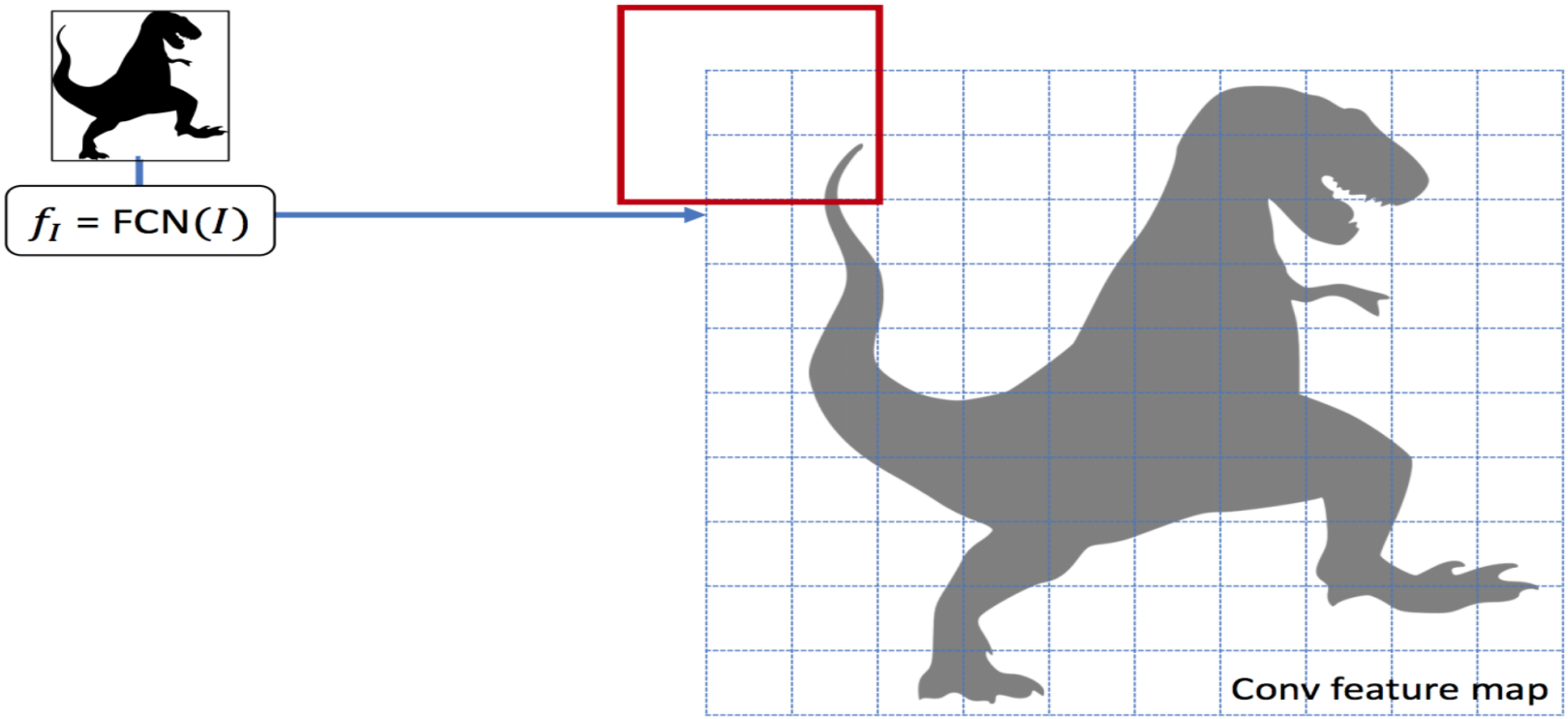
One network,
four losses



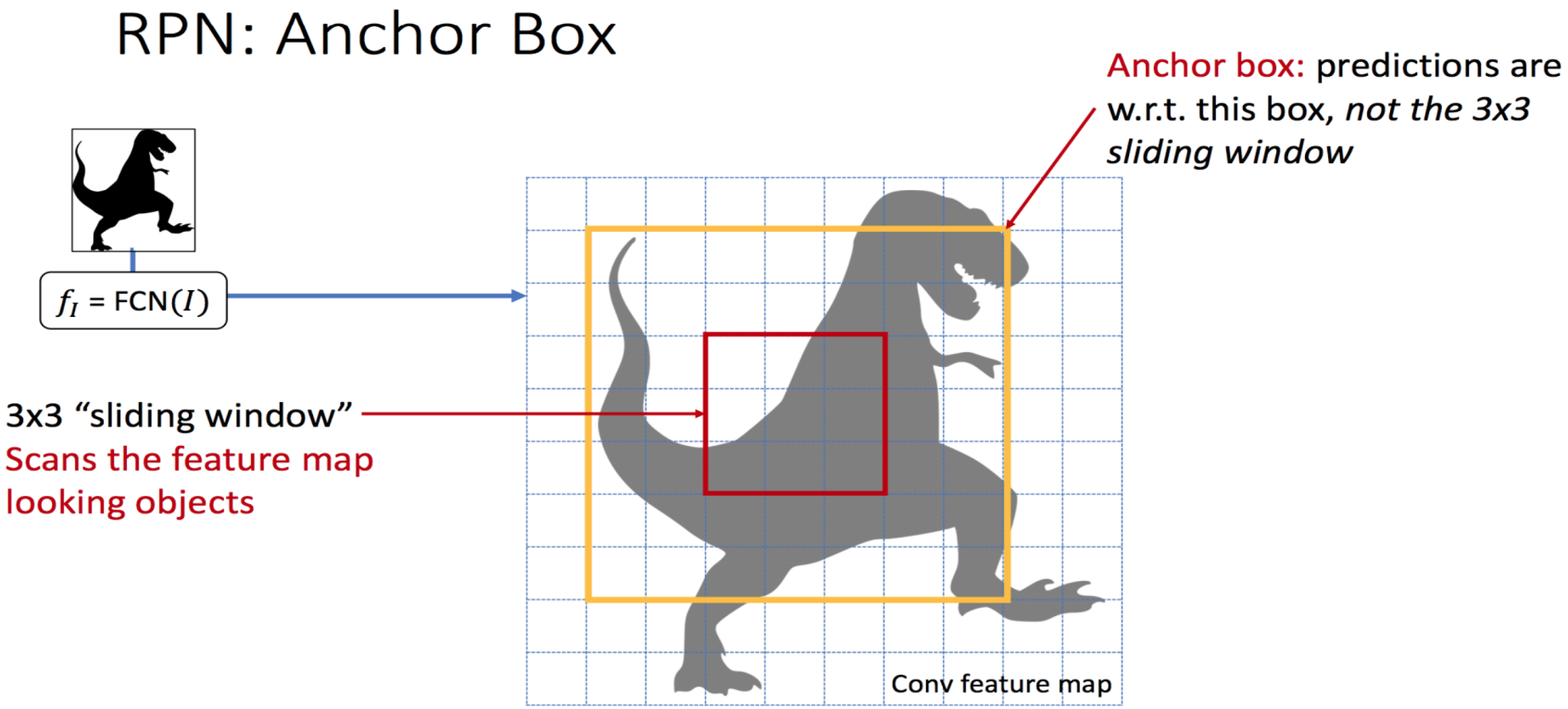
Faster R-CNN (Ren et al. NIPS 2015

	Fast R-CNN	R-CNN
Train time (h)	**9.5**	84
- Speedup	**8.8x**	
Test time / image	**0.32s**	47.0s
- Test speedup	**146x**	
mAP	**66.9%**	66.0%

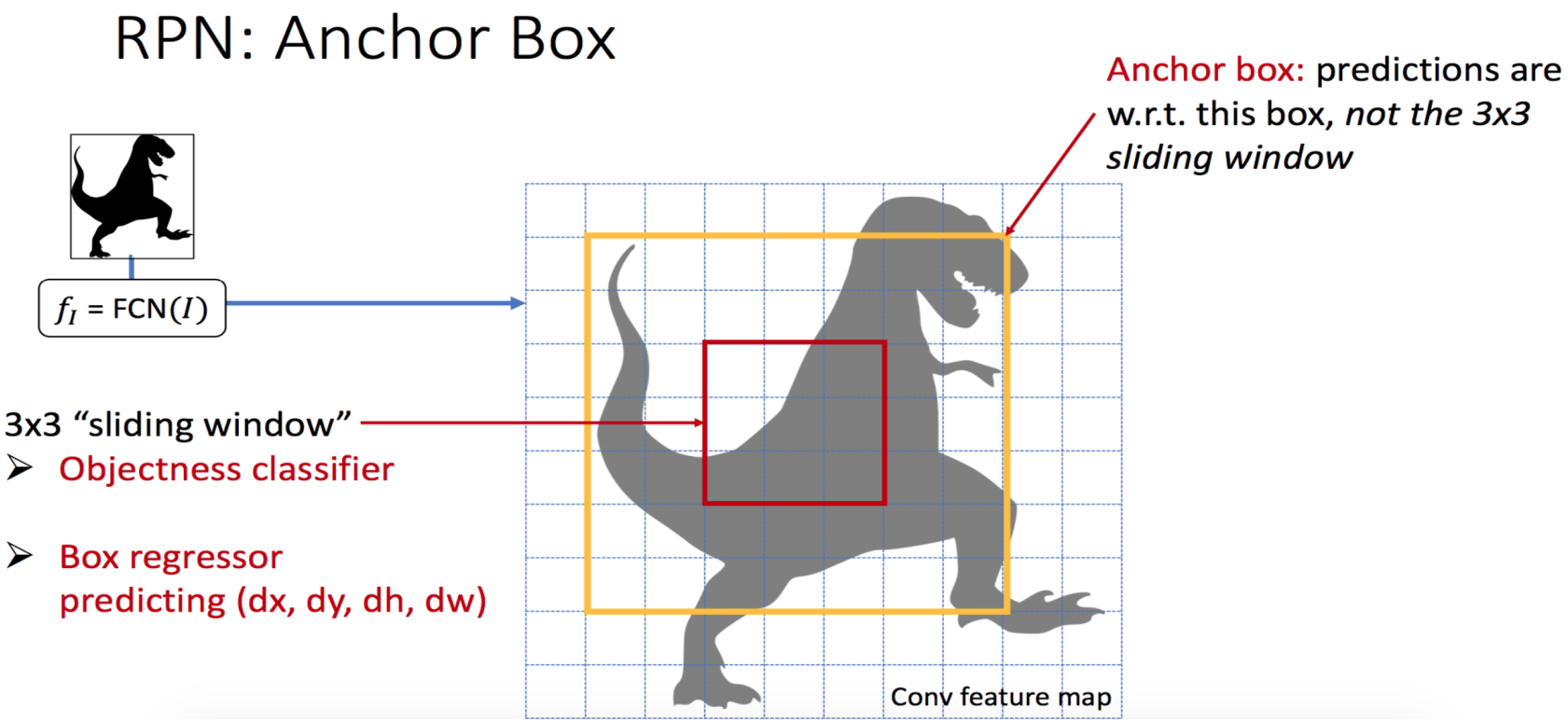
Faster R-CNN (Ren et al. NIPS 2015



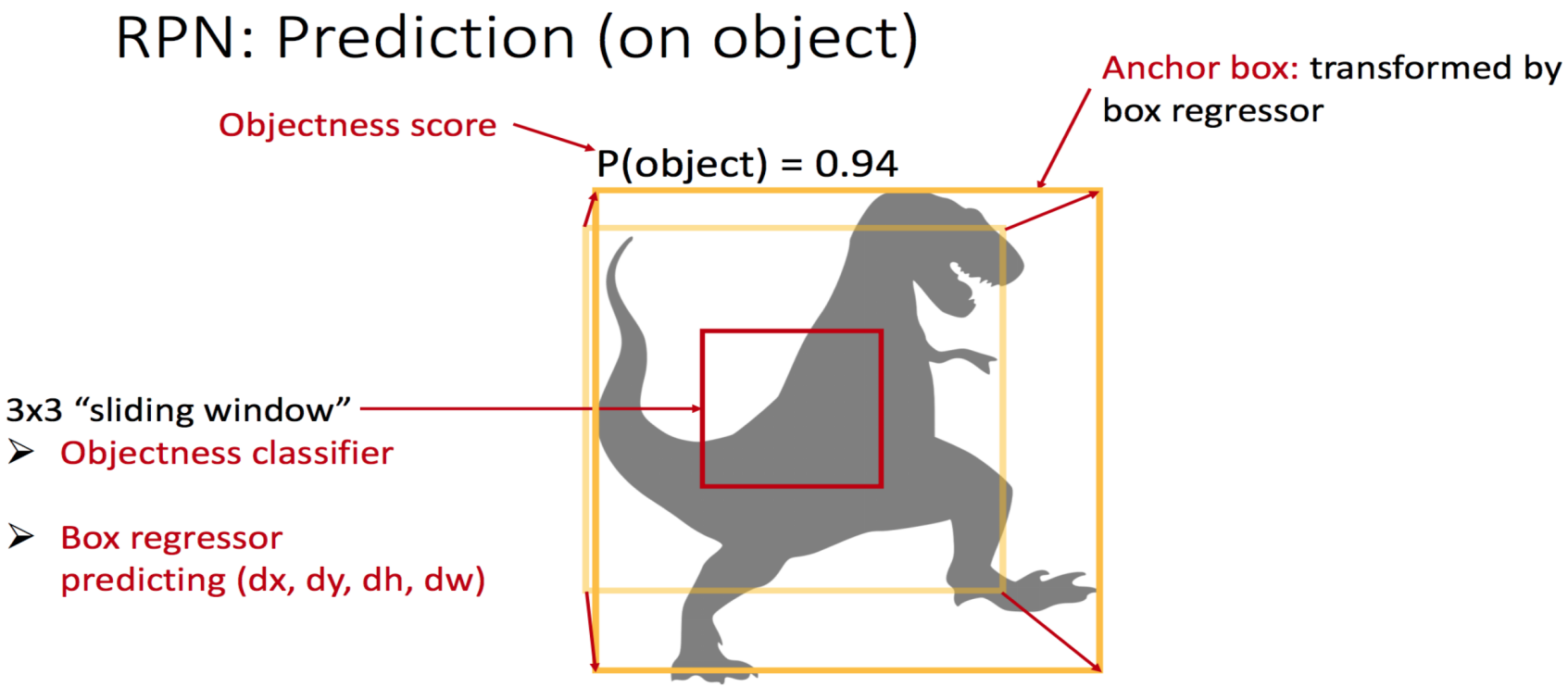
Faster R-CNN (Ren et al. NIPS 2015



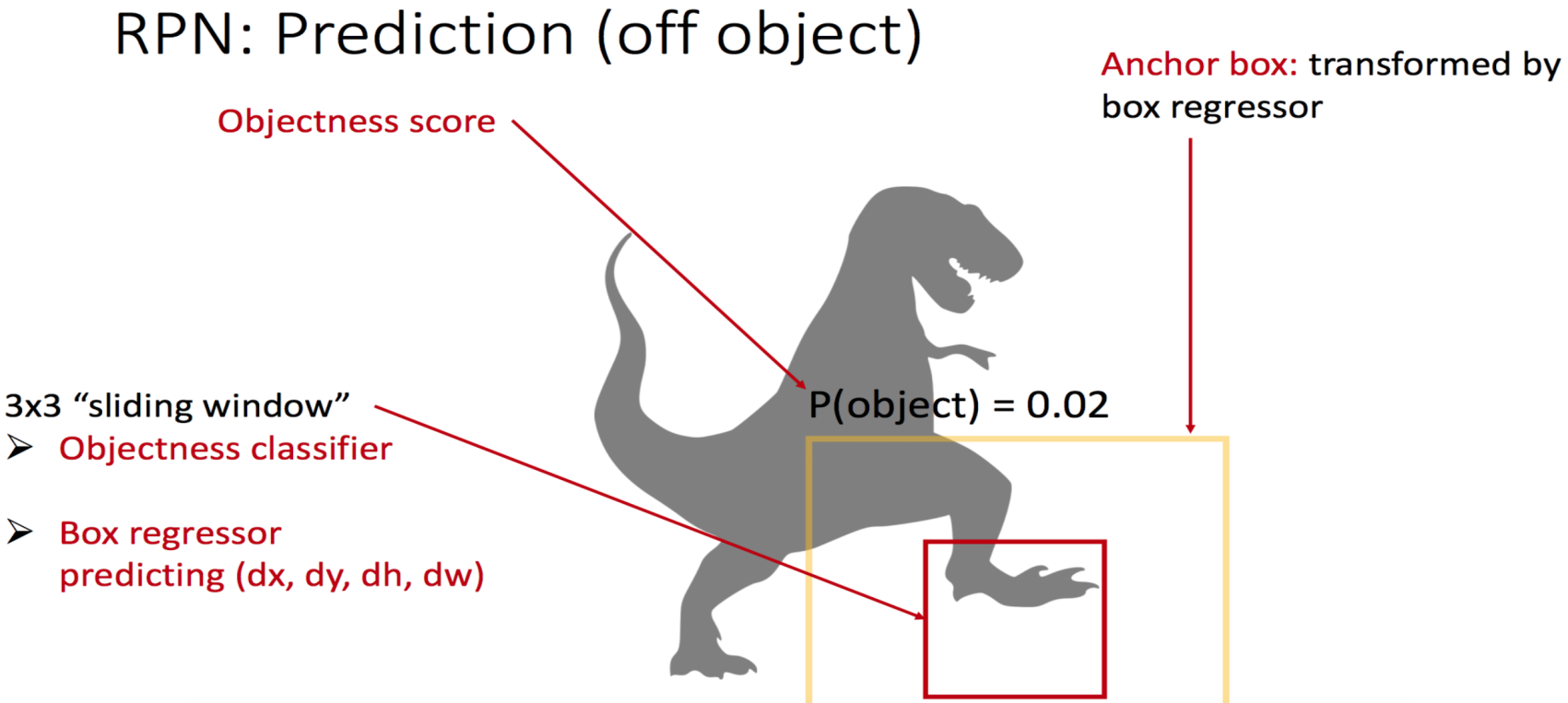
Faster R-CNN (Ren et al. NIPS 2015



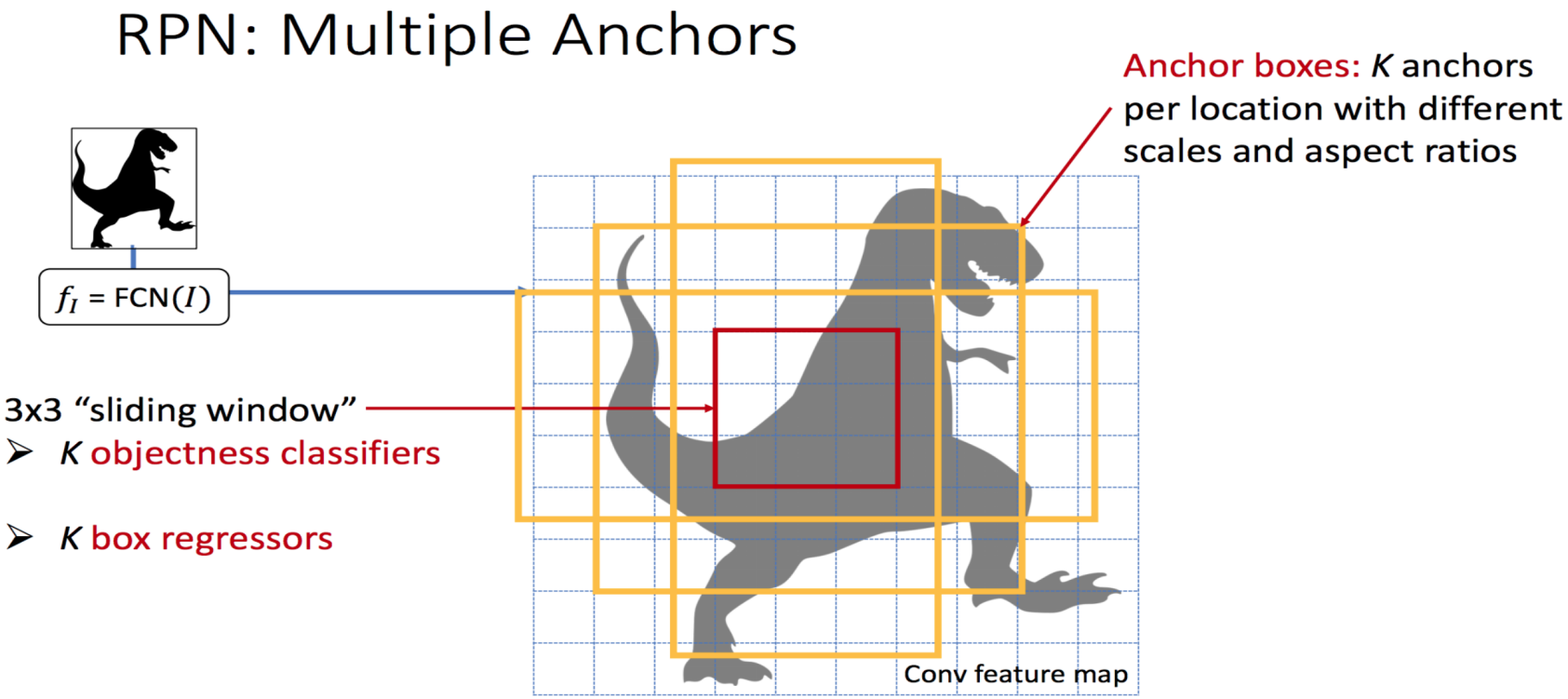
Faster R-CNN (Ren et al. NIPS 2015



Faster R-CNN (Ren et al. NIPS 2015

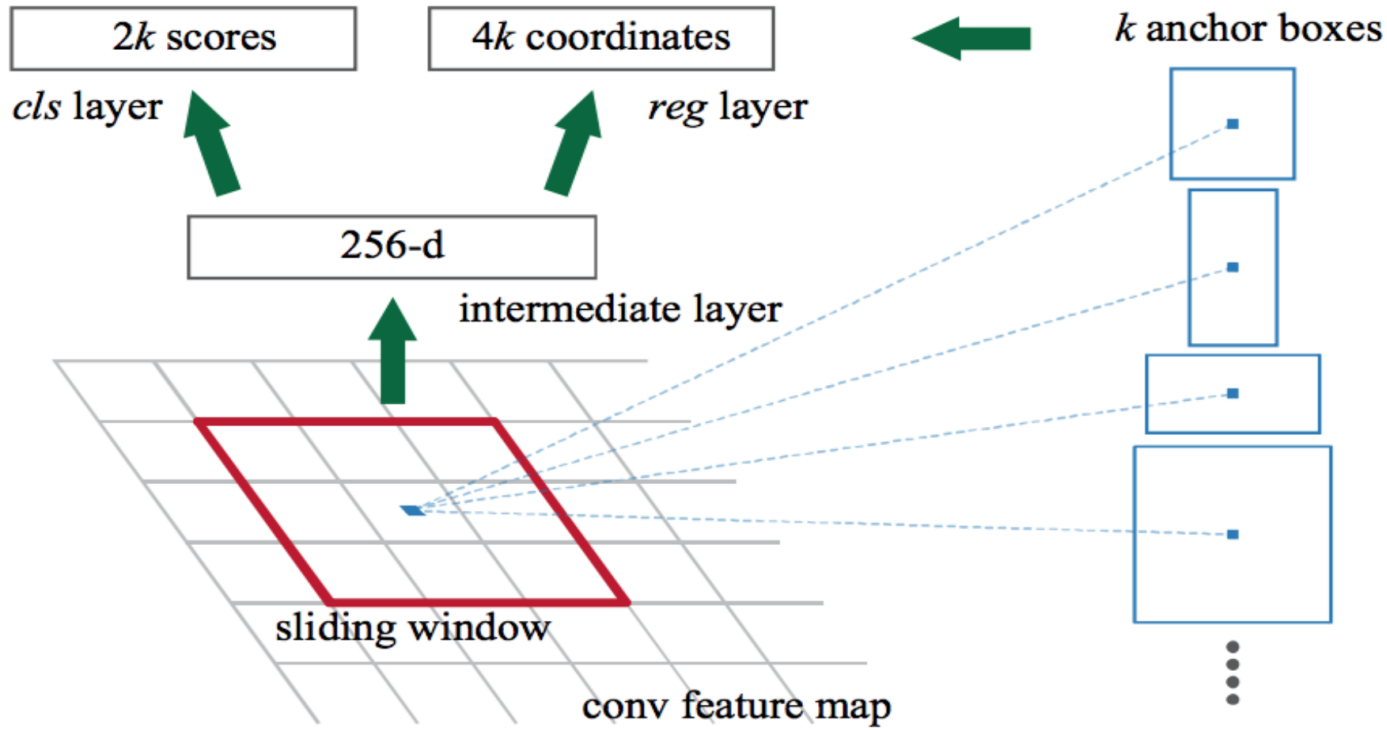


Faster R-CNN (Ren et al. NIPS 2015



Faster R-CNN (Ren et al. NIPS 2015

Region proposal network



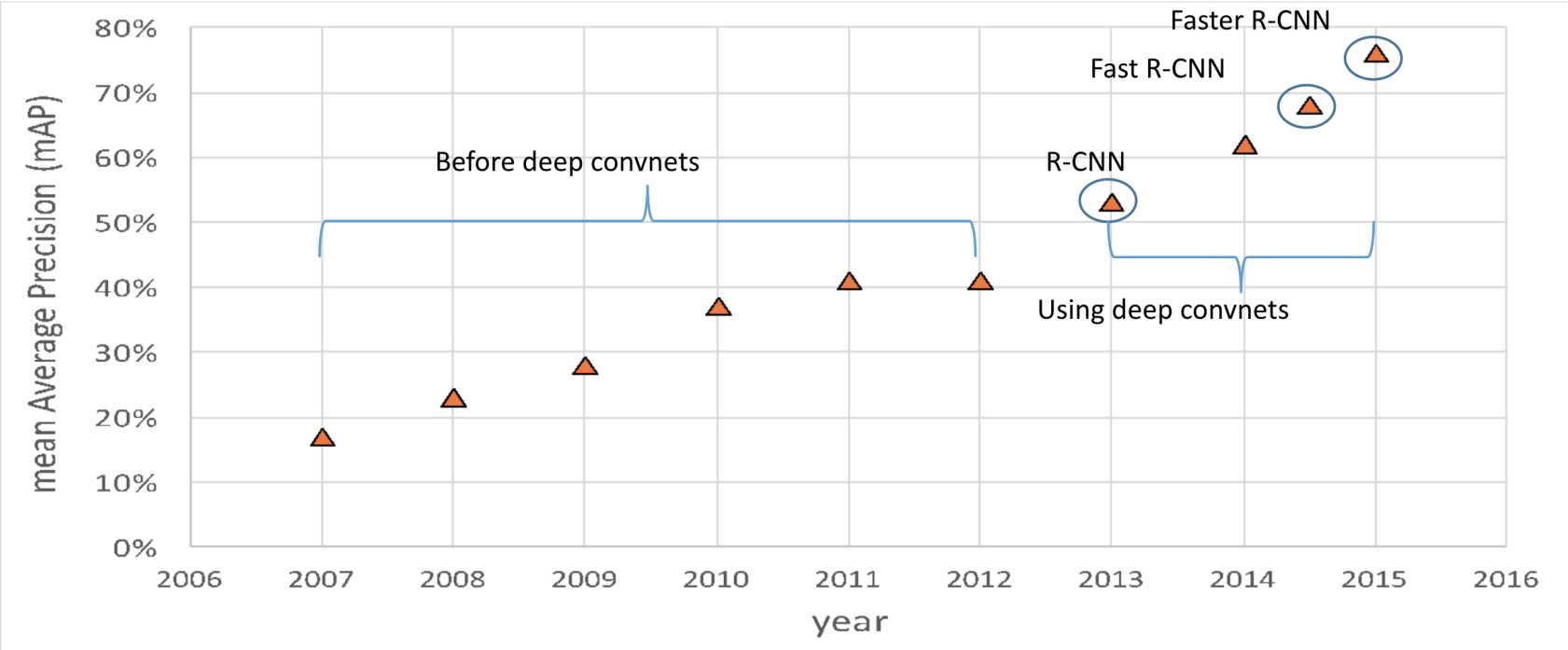
Faster R-CNN (Ren et al. NIPS 2015

Faster R-CNN results

system	time	07 data	07+12 data
R-CNN	~50s	66.0	-
Fast R-CNN	~2s	66.9	70.0
Faster R-CNN	198ms	**69.9**	**73.2**

detection mAP on PASCAL VOC 2007, with VGG-16 pre-trained on ImageNet

Faster R-CNN (Ren et al. NIPS 2015



Faster R-CNN (Ren et al. NIPS 2015)

- What could be the problems

Faster R-CNN (Ren et al. NIPS 2015)

- What could be the problems
  - Two-stage detection pipeline is still too slow to apply on real-time images and videos

References

The End

Questions? Comments?