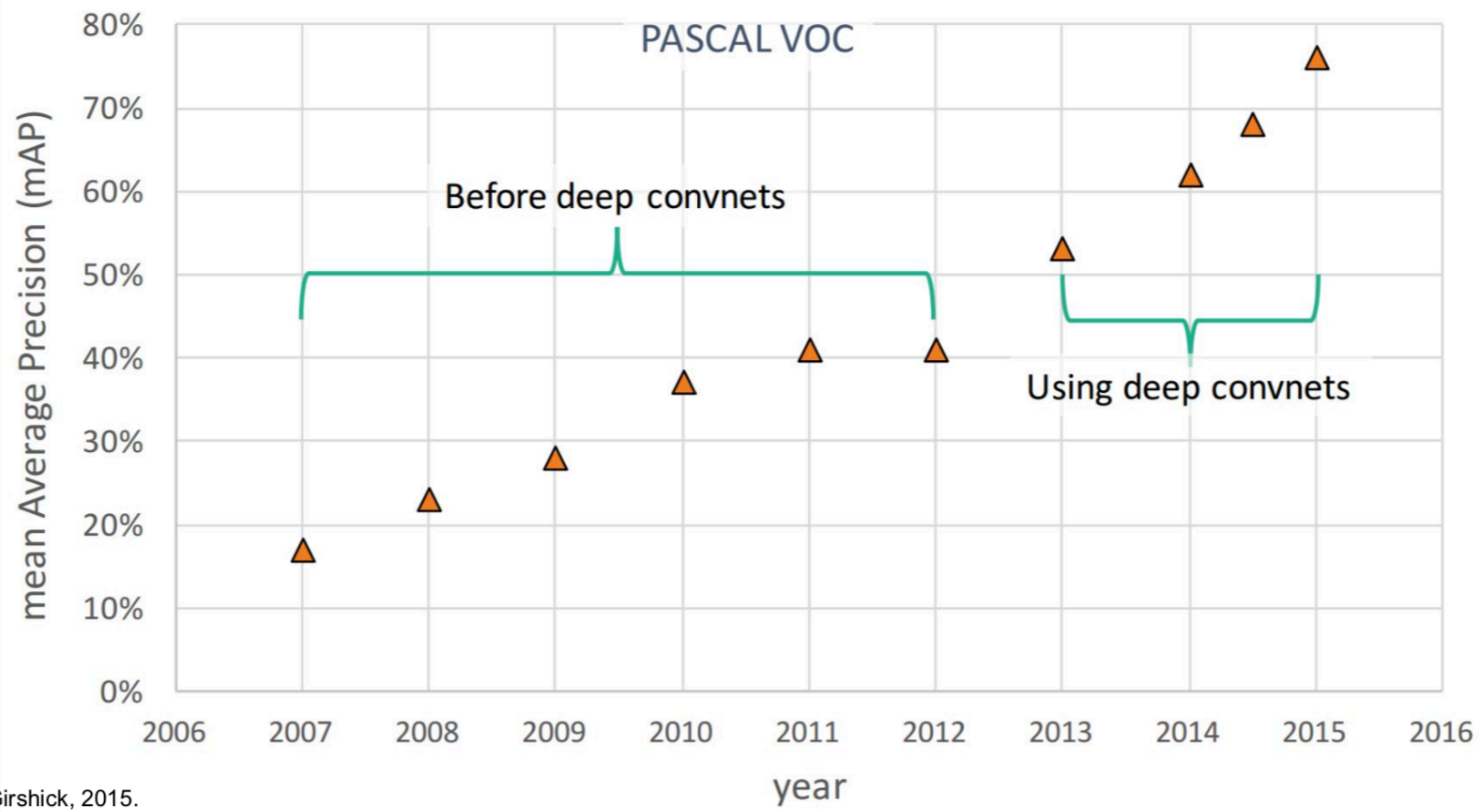
Object Detection

Zexin Yan
exxon.yan@smartmore.com

What is object detection?



What is object detection?

Classification: What is it?

This image is CC0 public domain

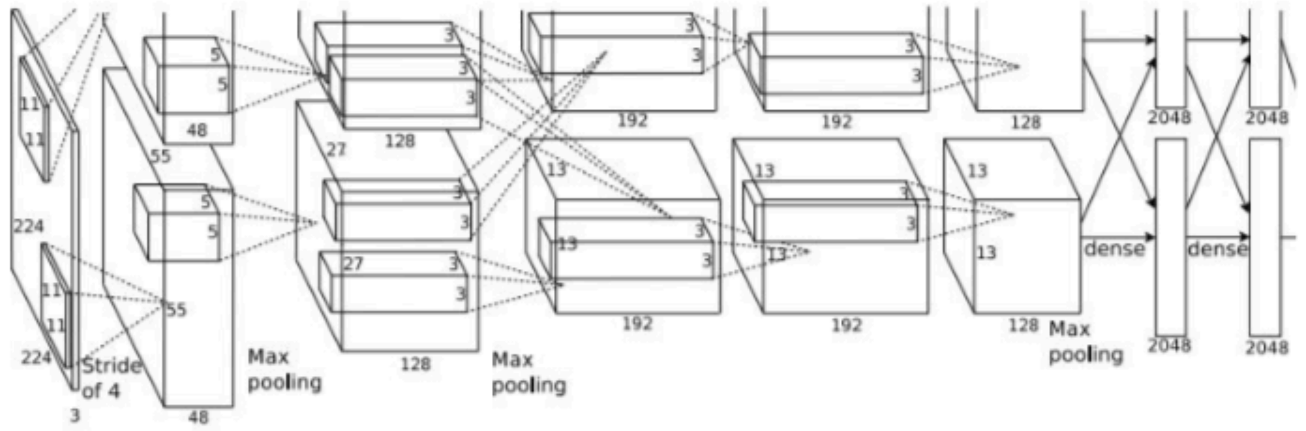


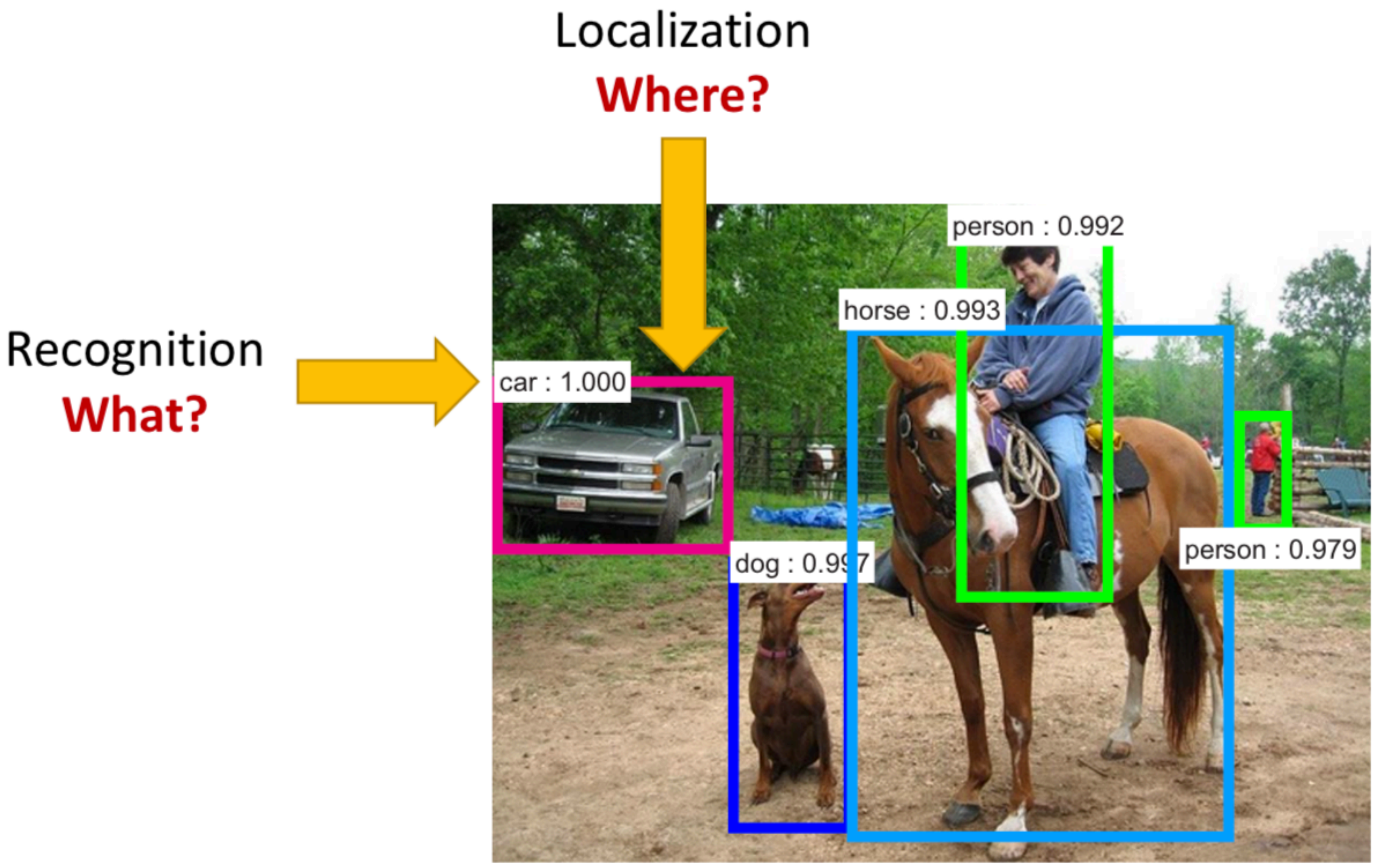
Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

Vector: 4096

Fully-Connected: 4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Object Detection: What and where?

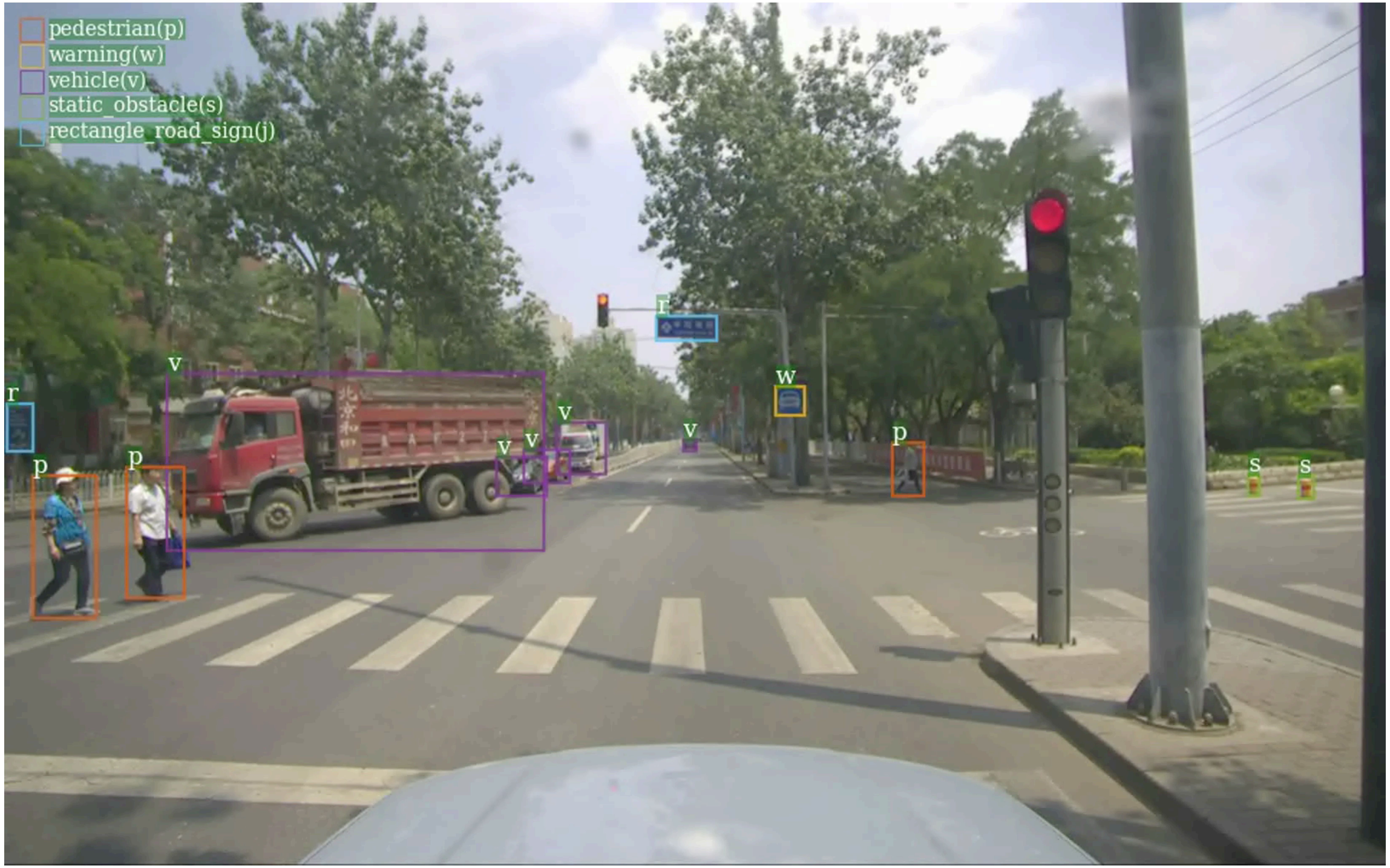


The application of object detection?

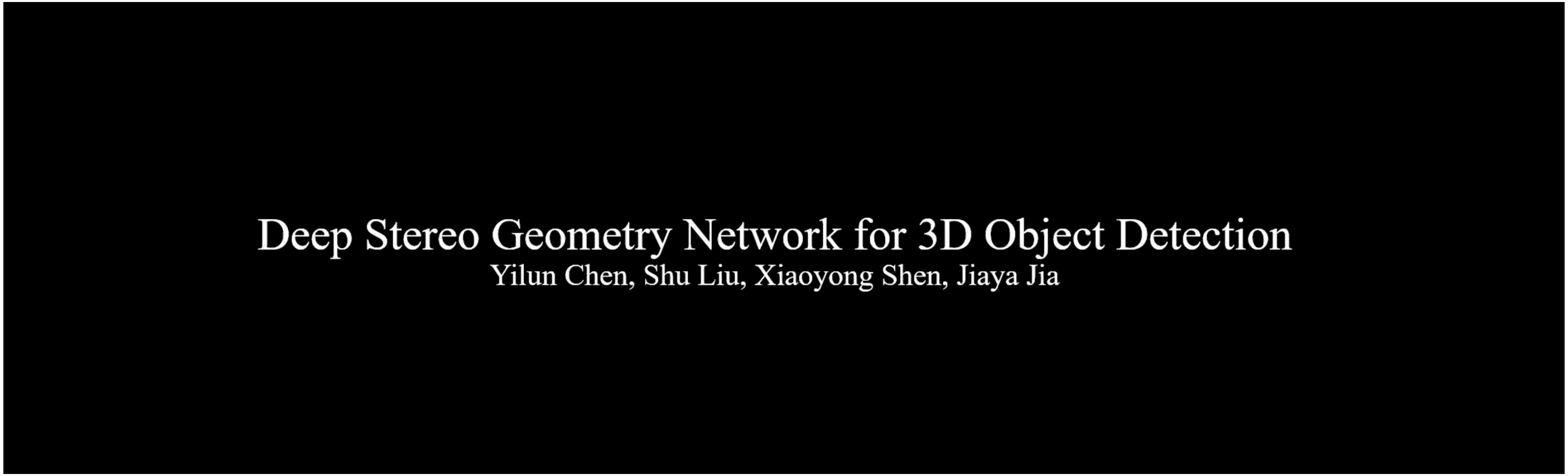
Face Detection

The application of object detection?

Autonomous driving

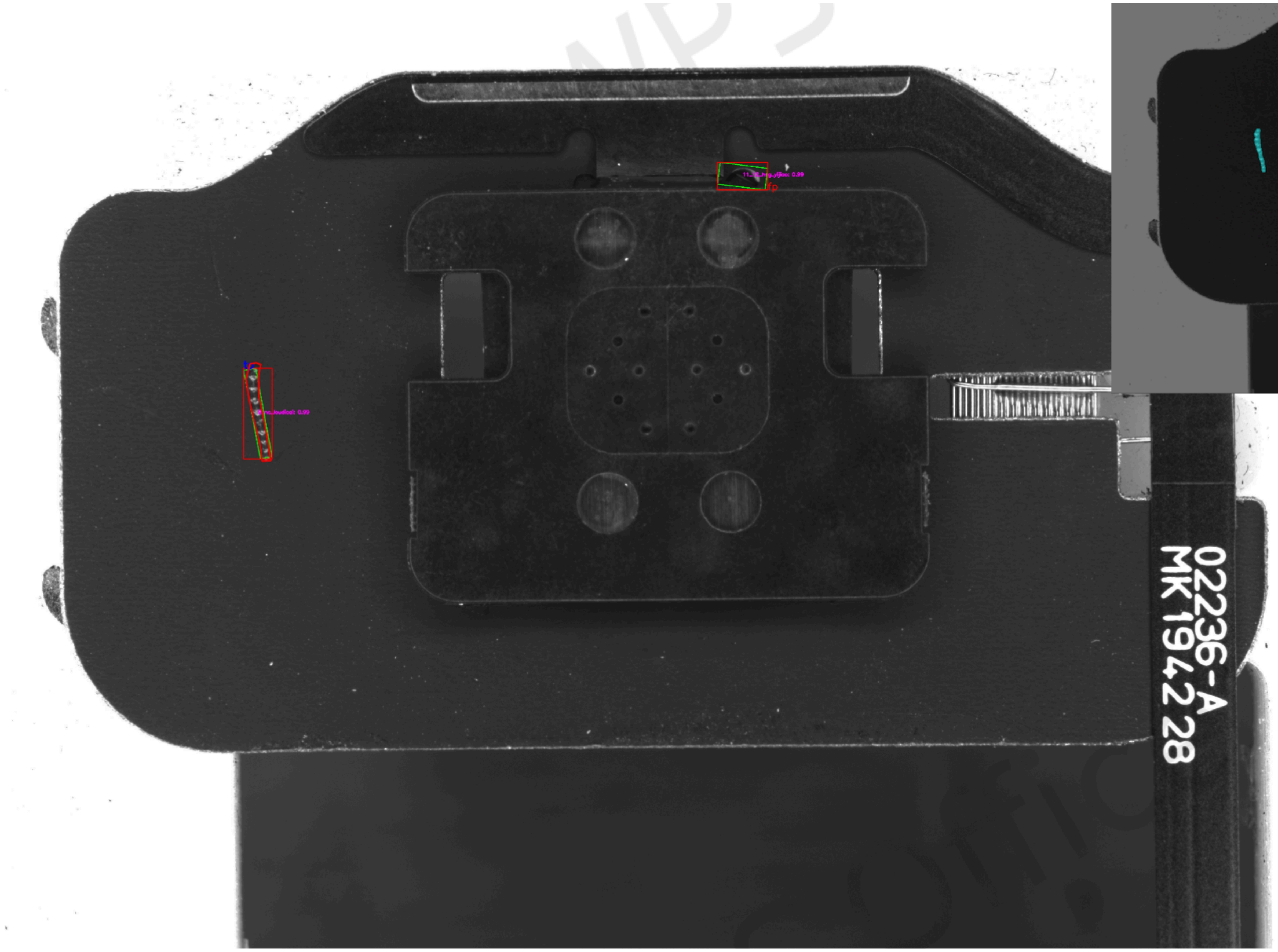


The application of object detection?



The application of object detection?

Defect Detection



Metrics

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

TP = True positive

TN = True negative

FP = False positive

FN = False negative

Metrics

Ground truth

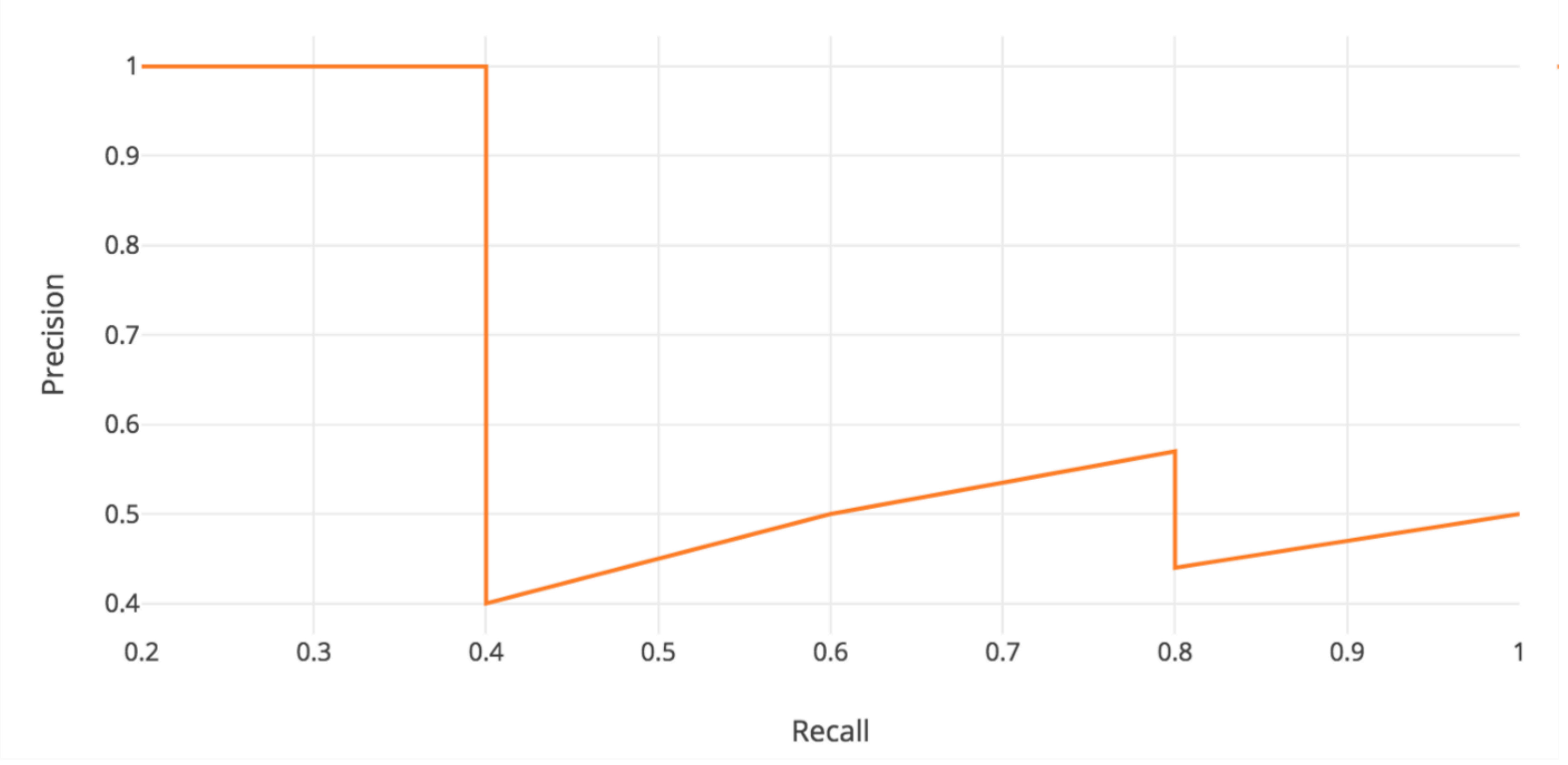
Prediction

$$IoU = \frac{\text{area of overlap}}{\text{area of union}}$$

Metrics

Rank by confidence

Rank	Correct?	Precision	Recall
1	True	1.0 ↑	0.2 ↑
2	True	1.0 –	0.4 ↑
3	False	0.67 ↓	0.4 –
4	False	0.5 ↓	0.4 –
5	False	0.4 ↓	0.4 –
6	True	0.5 ↑	0.6 ↑
7	True	0.57 ↑	0.8 ↑



Precision-recall curve

$$\mathrm{AP} = \int_0^1 p(r)dr$$

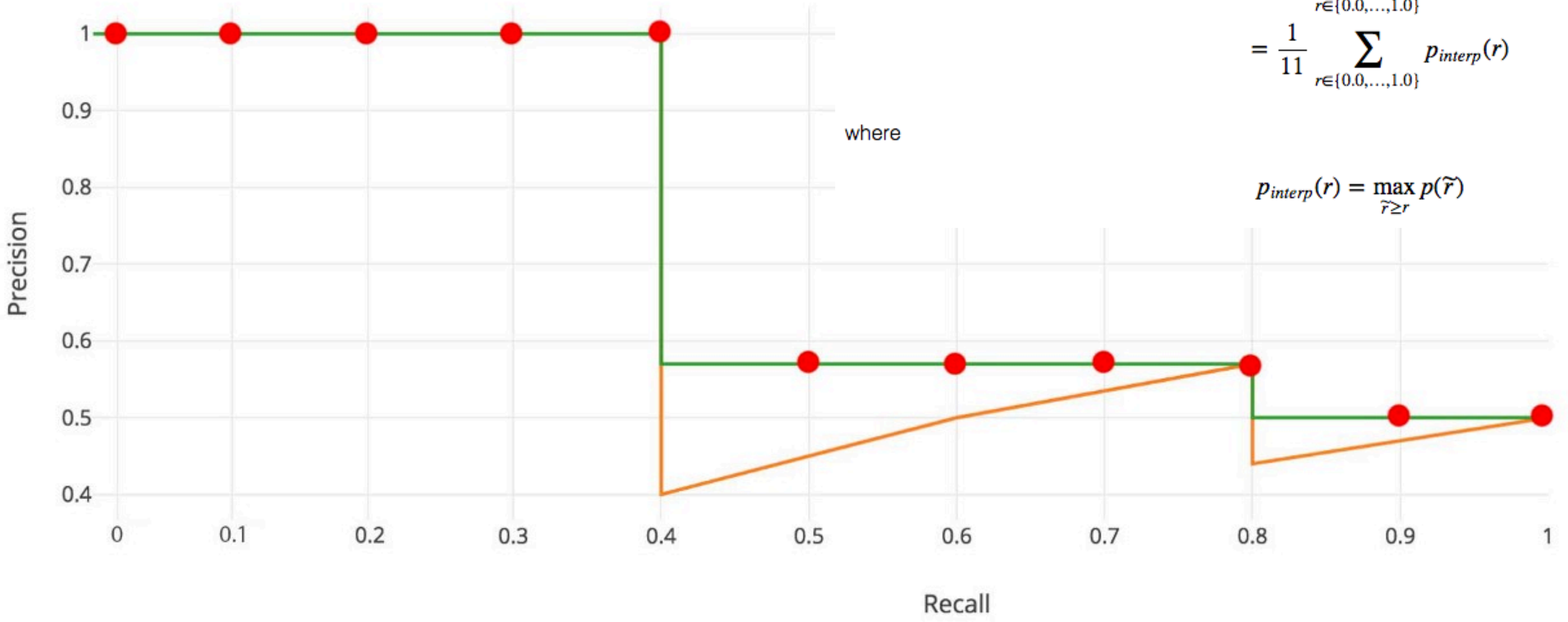
Metrics

Interpolated AP

$$AP = \frac{1}{11} \sum_{r \in \{0.0, \ldots, 1.0\}} AP_r$$
$$= \frac{1}{11} \sum_{r \in \{0.0, \ldots, 1.0\}} p_{interp}(r)$$

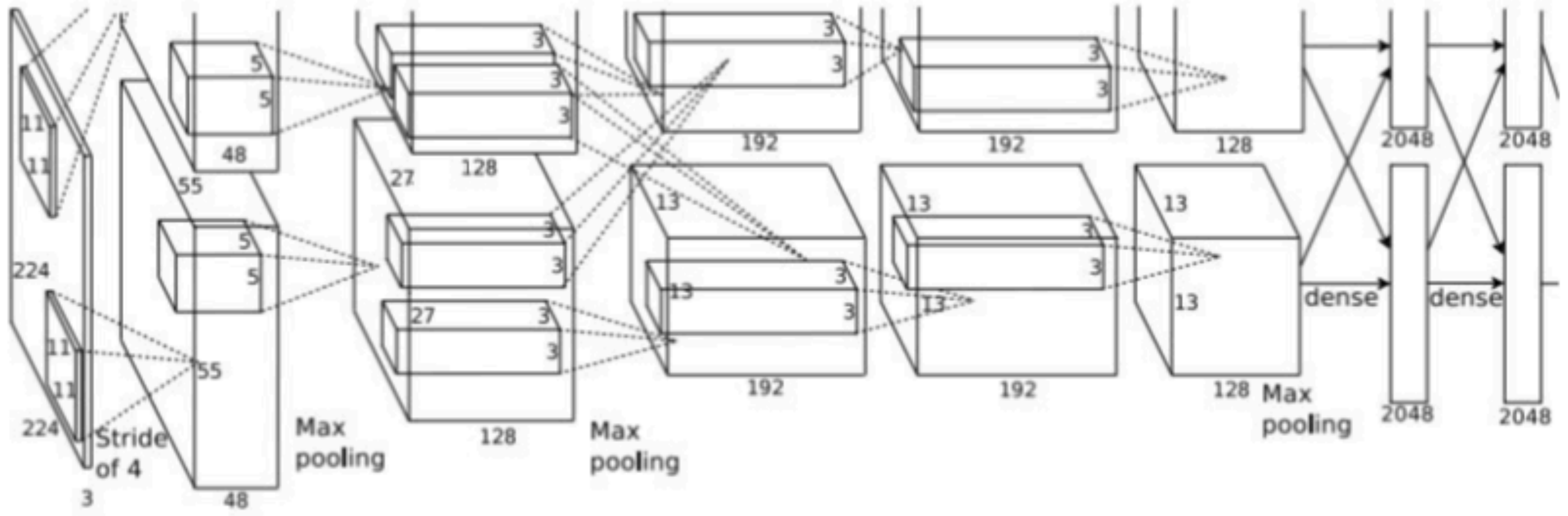
where

$$p_{interp}(r) = \max_{\tilde{r} \geq r} p(\tilde{r})$$



Object Detection as Classification: Sliding Window

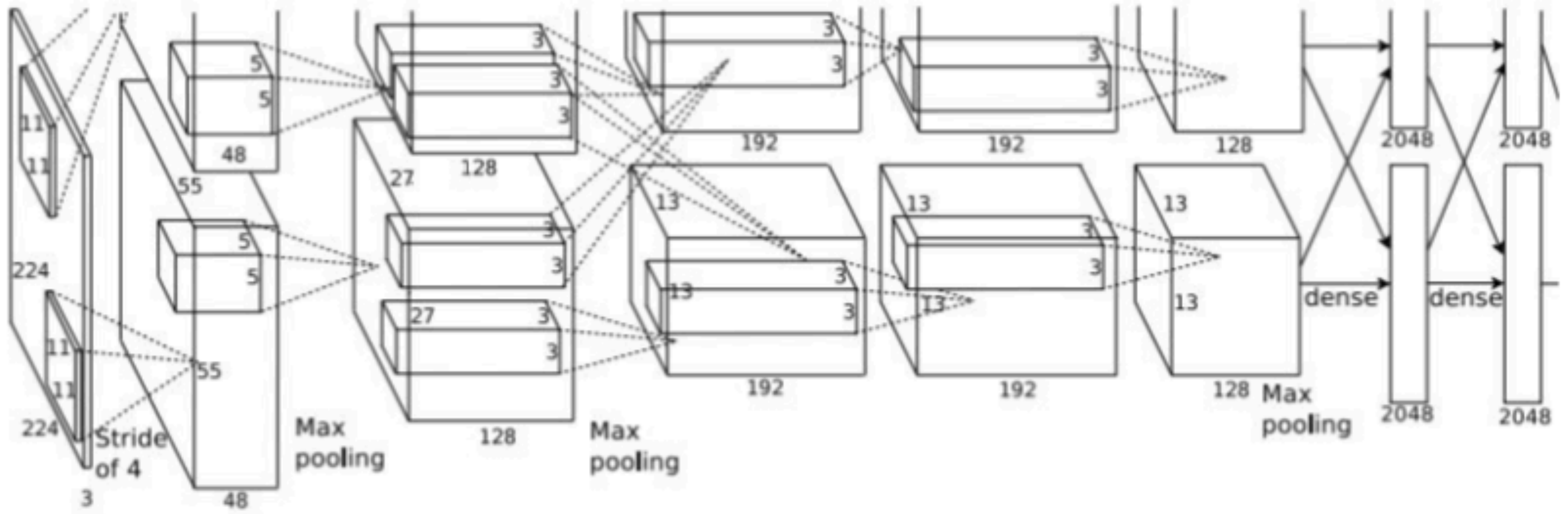
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? NO
Background? YES

Object Detection as Classification: Sliding Window

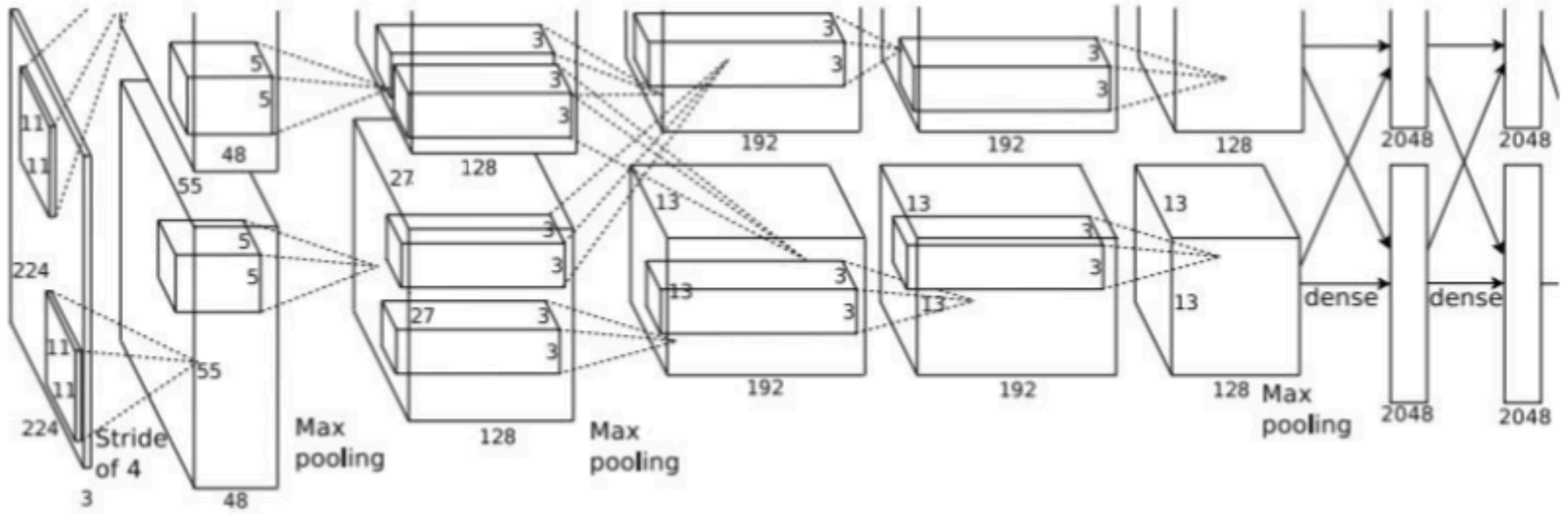
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

Object Detection as Classification: Sliding Window

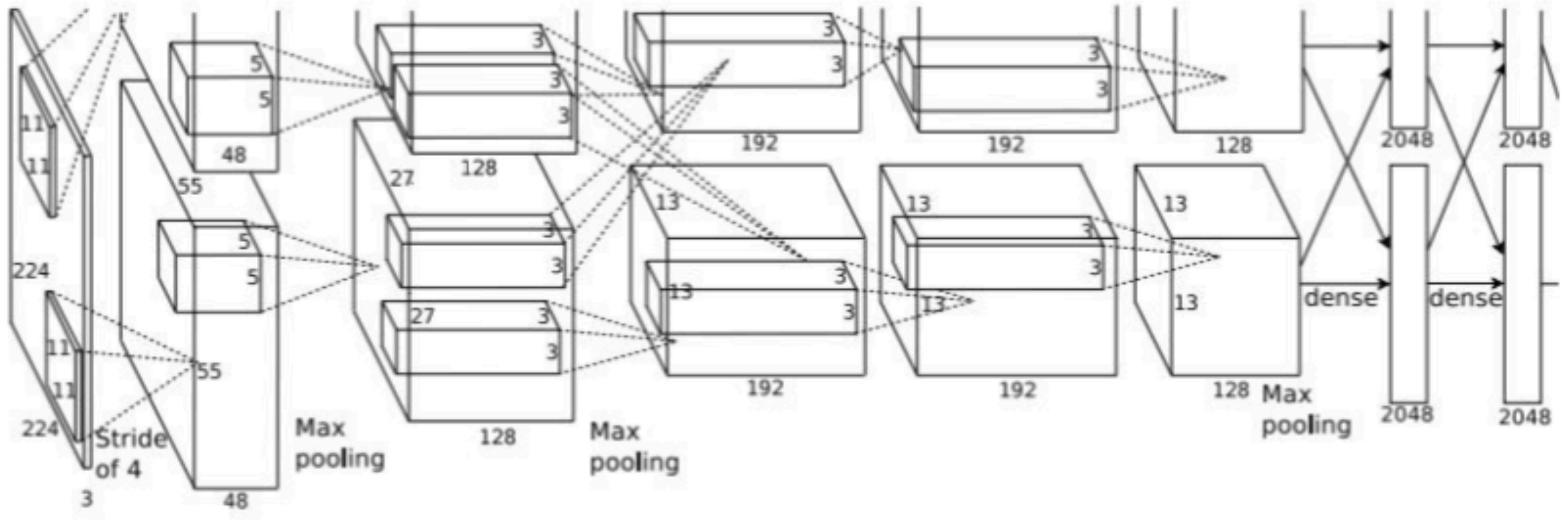
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

Object Detection as Classification: Sliding Window

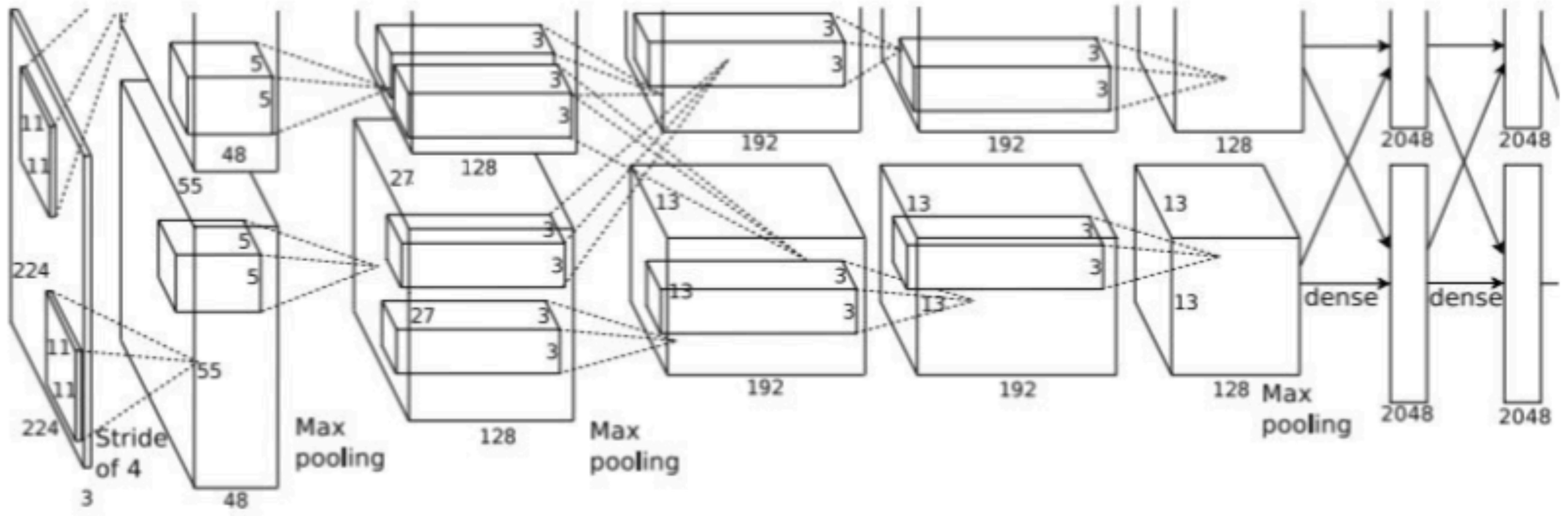
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? YES
Background? NO

Object Detection as Classification: Sliding Window

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

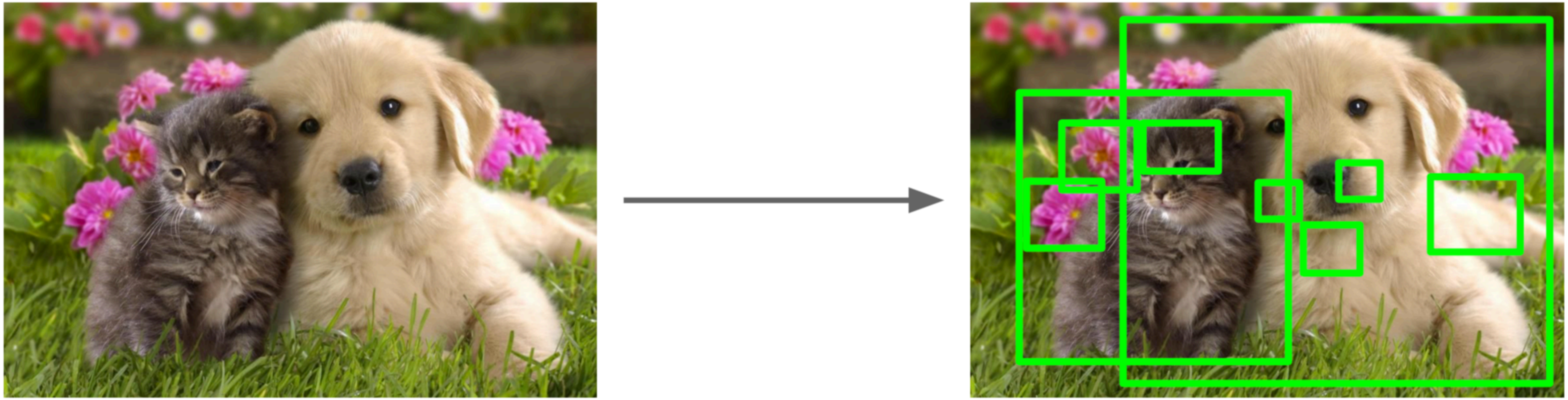


Dog? NO
Cat? YES
Background? NO

Problem: Need to apply CNN to huge number of locations and scales, very computationally expensive!

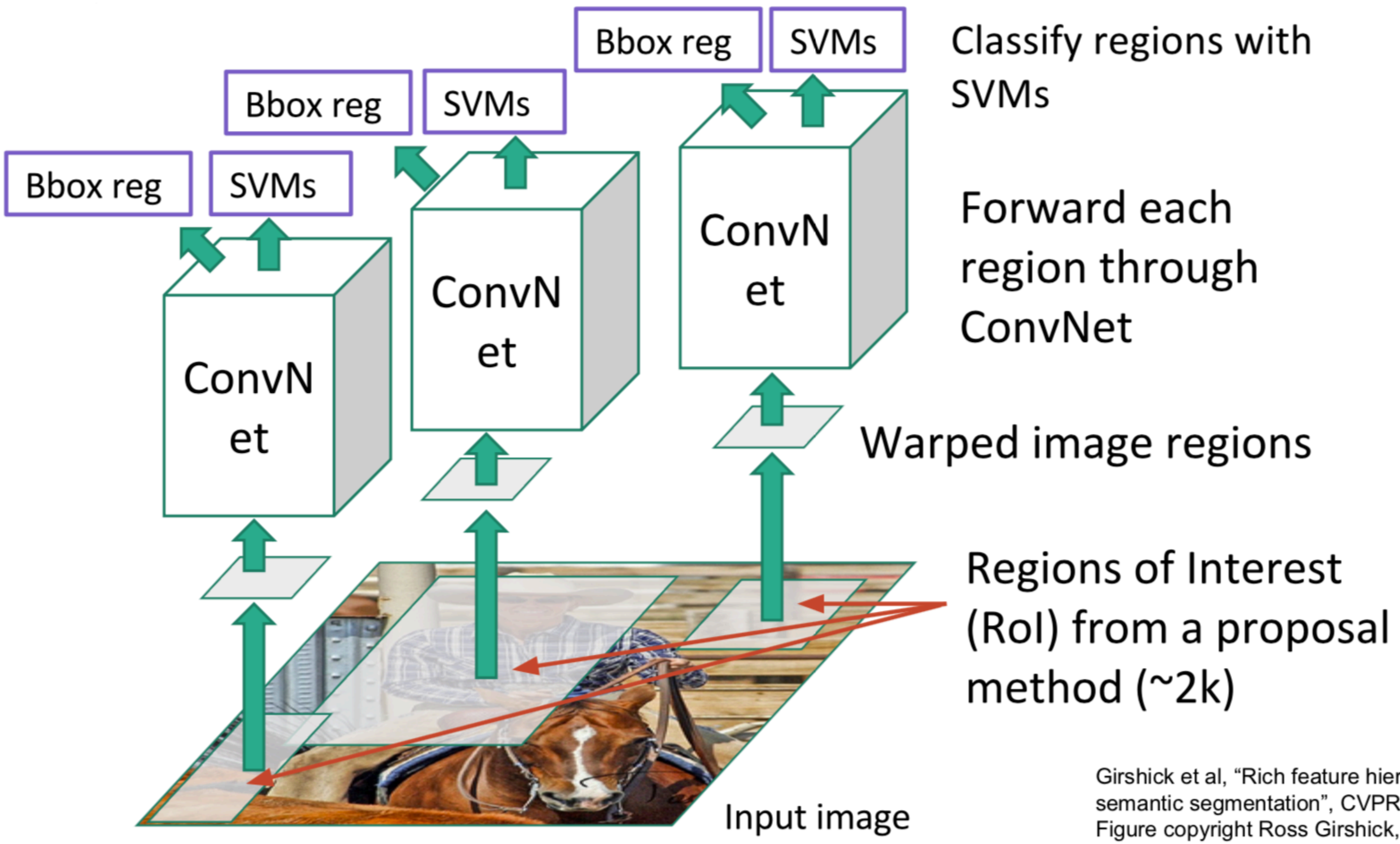
Region Proposals

- Find “blobby” image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 1000 region proposals in a few seconds on CPU



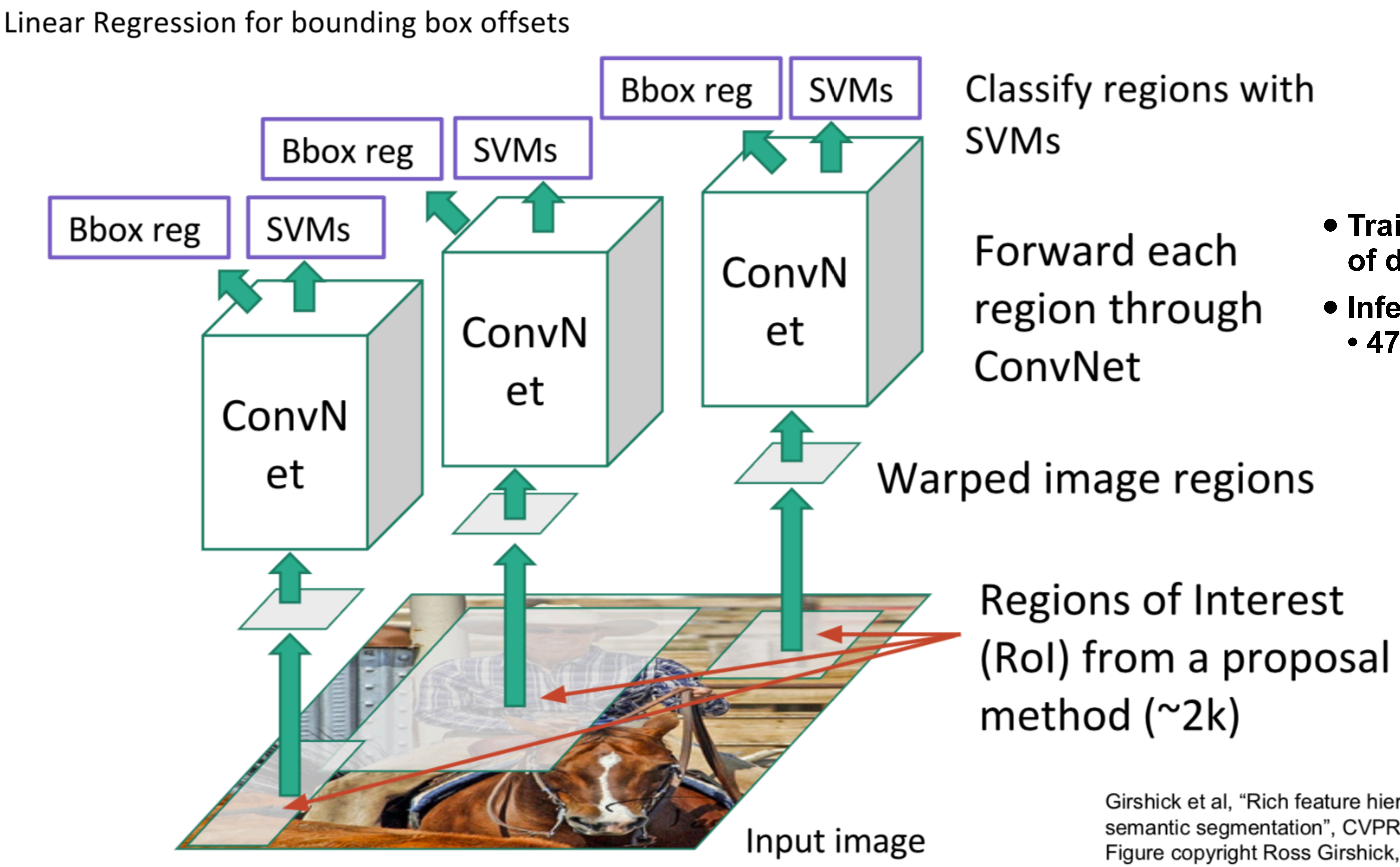
R-CNN

Linear Regression for bounding box offsets



Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

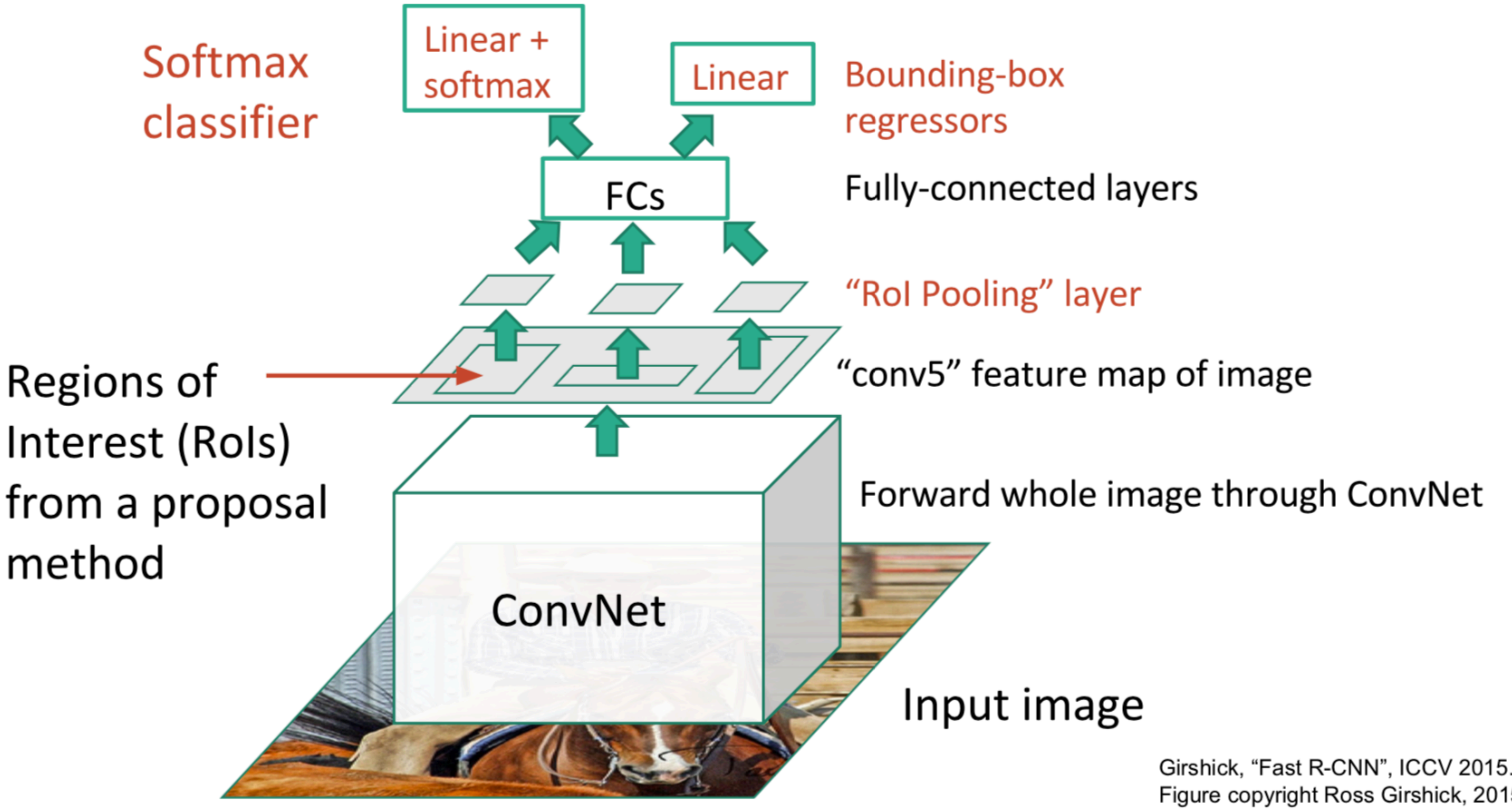
R-CNN



- **Training is slow (84h), takes a lot of disk space**
- **Inference (detection) is slow**
  - **47s / image with VGG16**

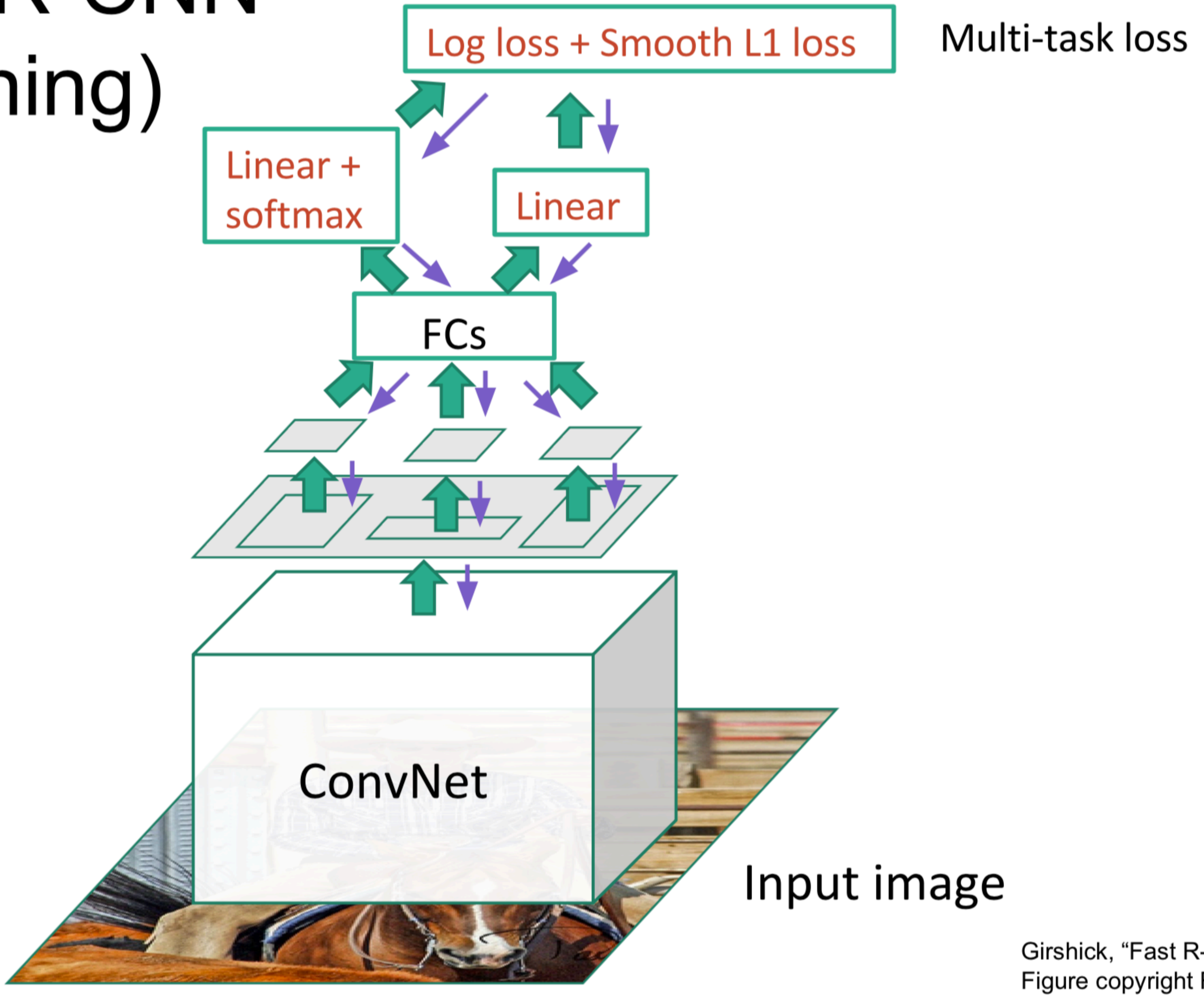
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

Fast R-CNN



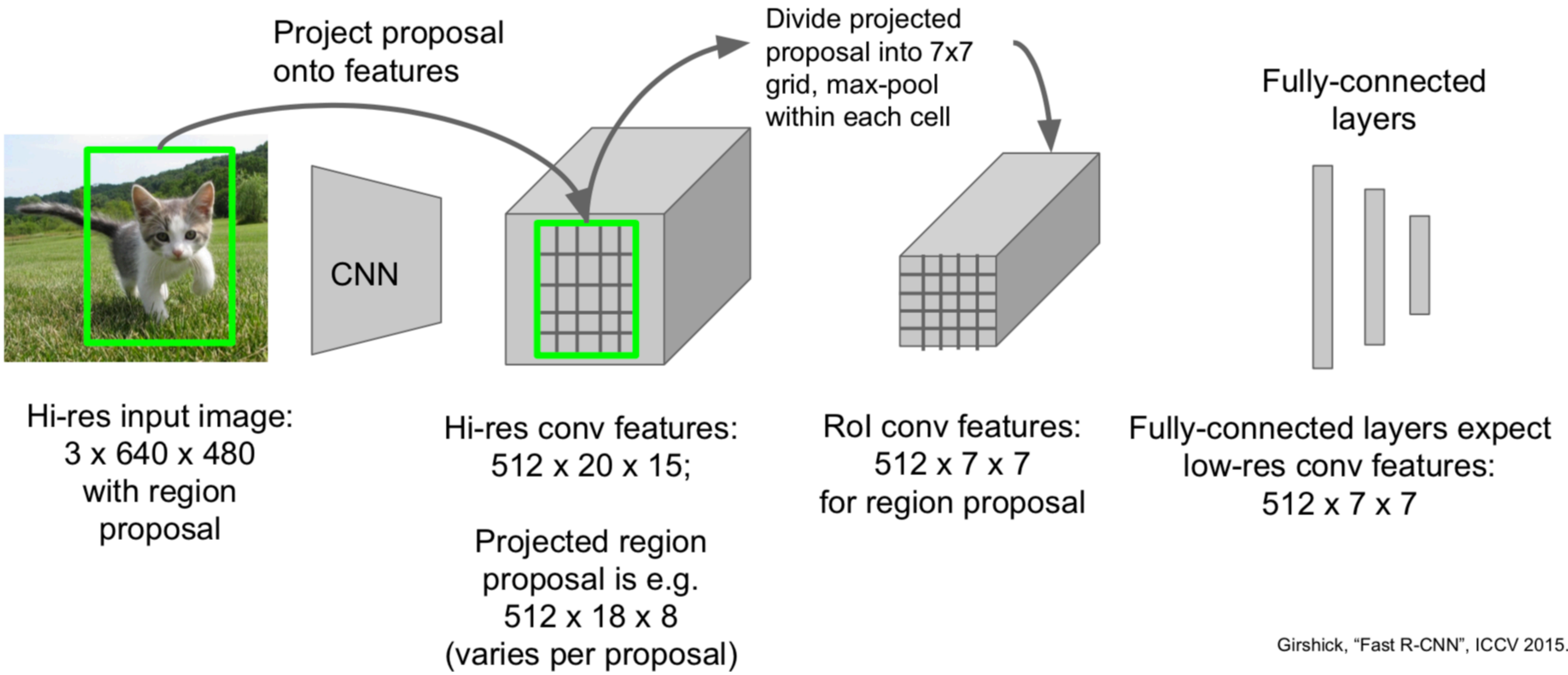
Girshick, "Fast R-CNN", ICCV 2015.

Fast R-CNN (Training)



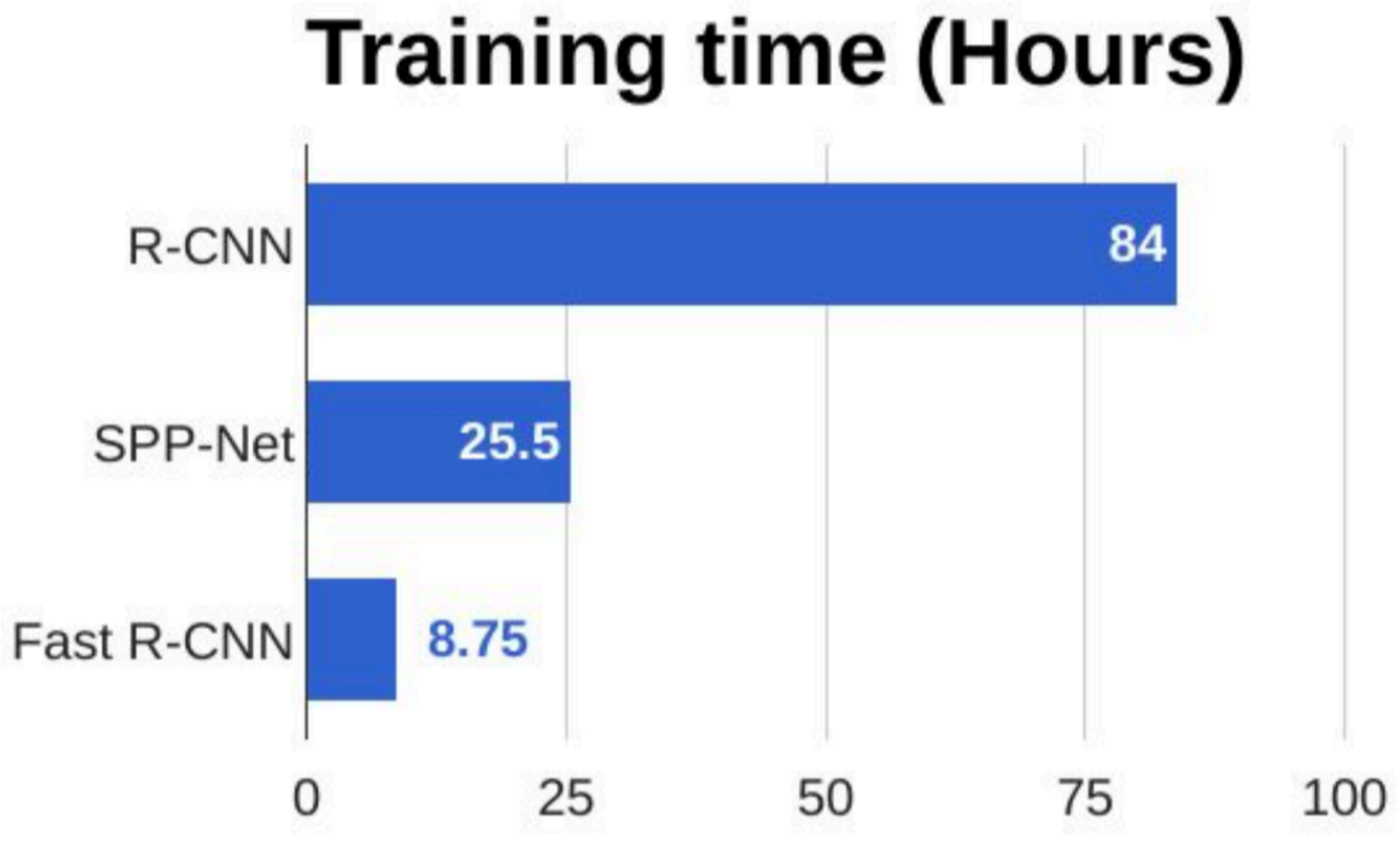
Girshick, "Fast R-CNN", ICCV 2015.

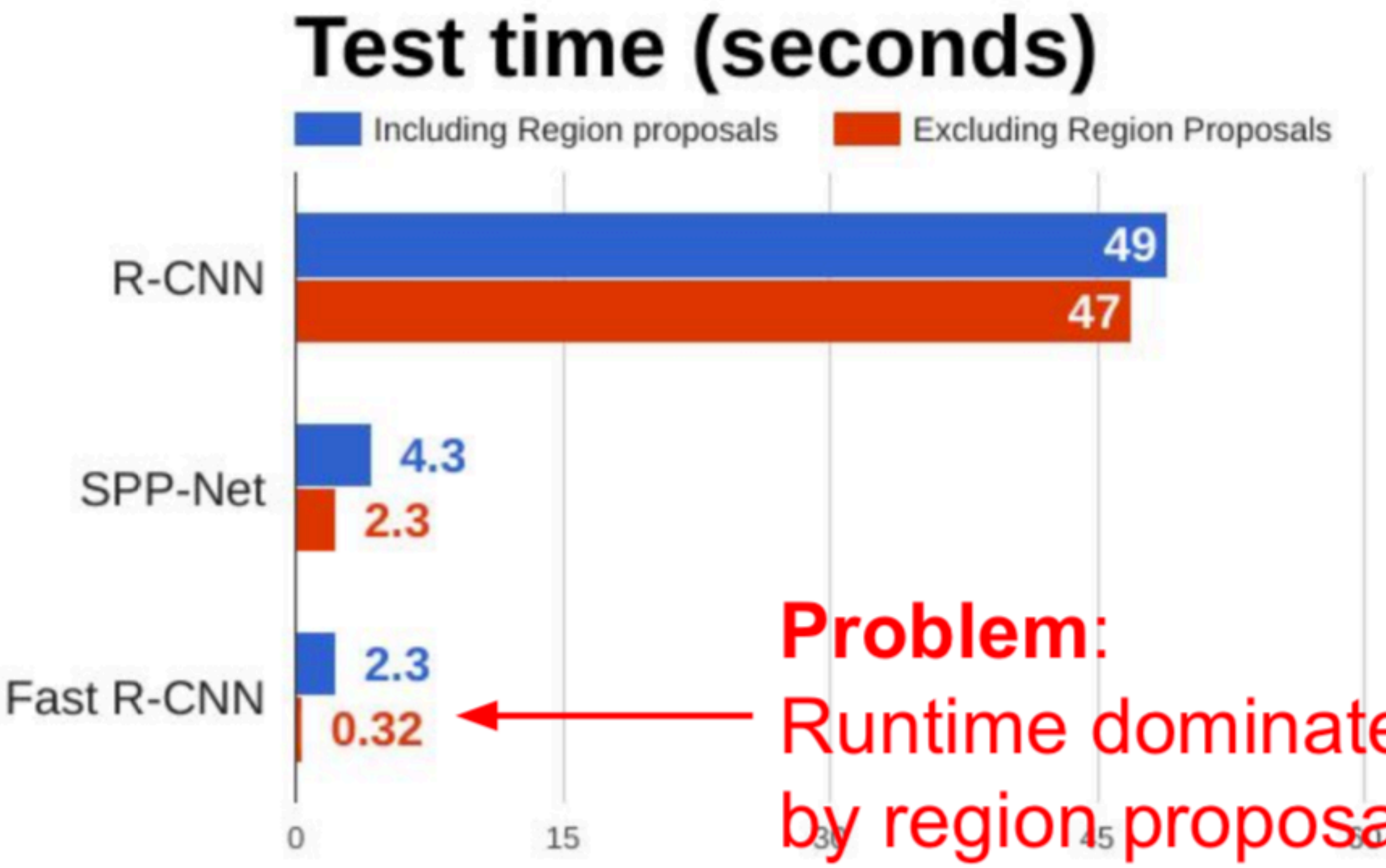
Fast R-CNN RoI Pooling



Girshick, “Fast R-CNN”, ICCV 2015.

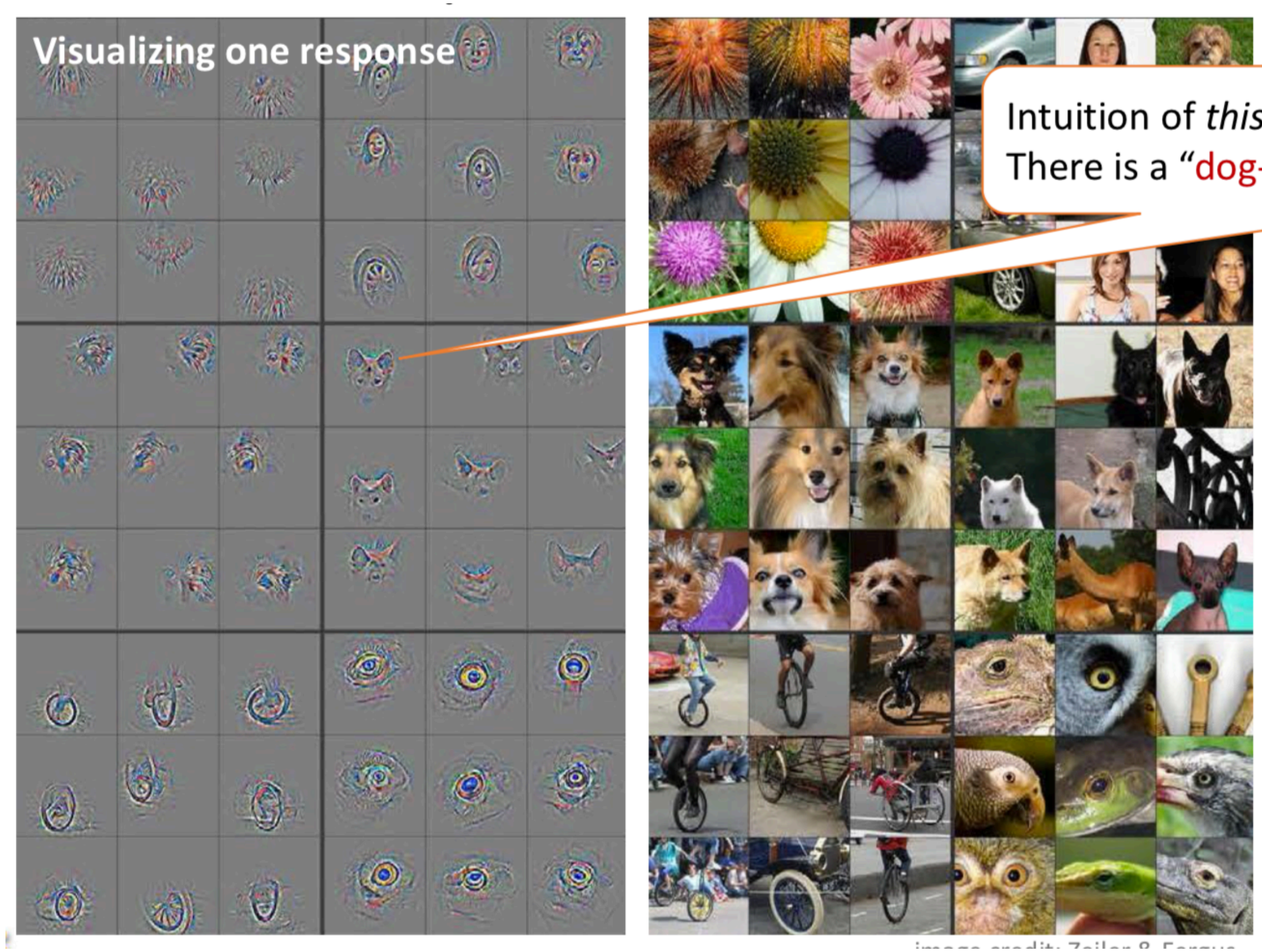
Fast R-CNN





Problem: Runtime dominated by region proposals!

Feature Maps = features and their locations



Intuition of *this visualization*:
There is a "dog-head" shape at this position.

- Location of a feature:
explicitly represents *where* it is.

- Responses of a feature:
encode *what* it is, and implicitly encode finer position information –

finer position information is encoded in the channel dimensions (e.g., bbox regression from responses at one pixel as in RPN)

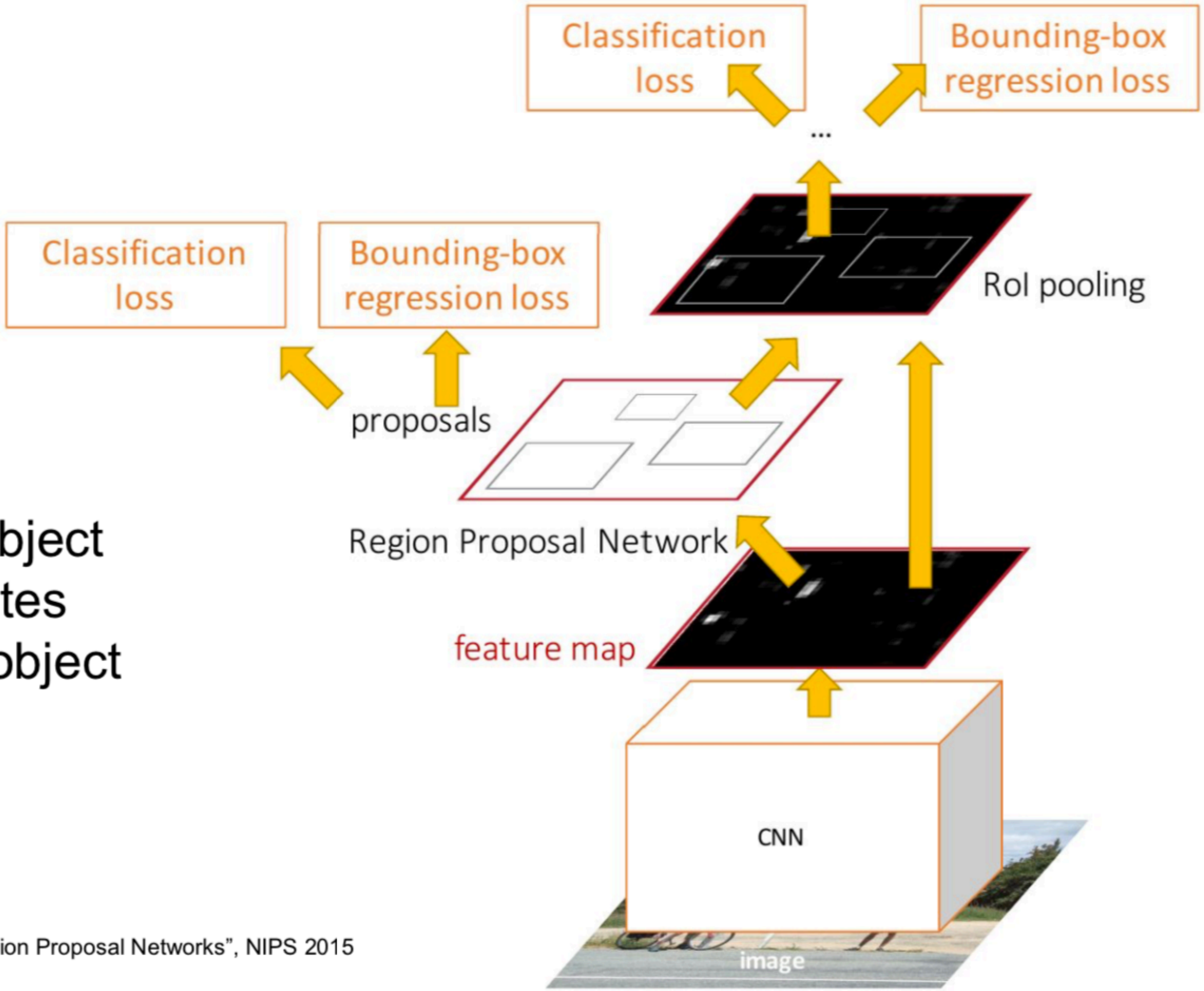
Faster R-CNN:

Make CNN do proposals!

Insert **Region Proposal Network (RPN)** to predict proposals from features

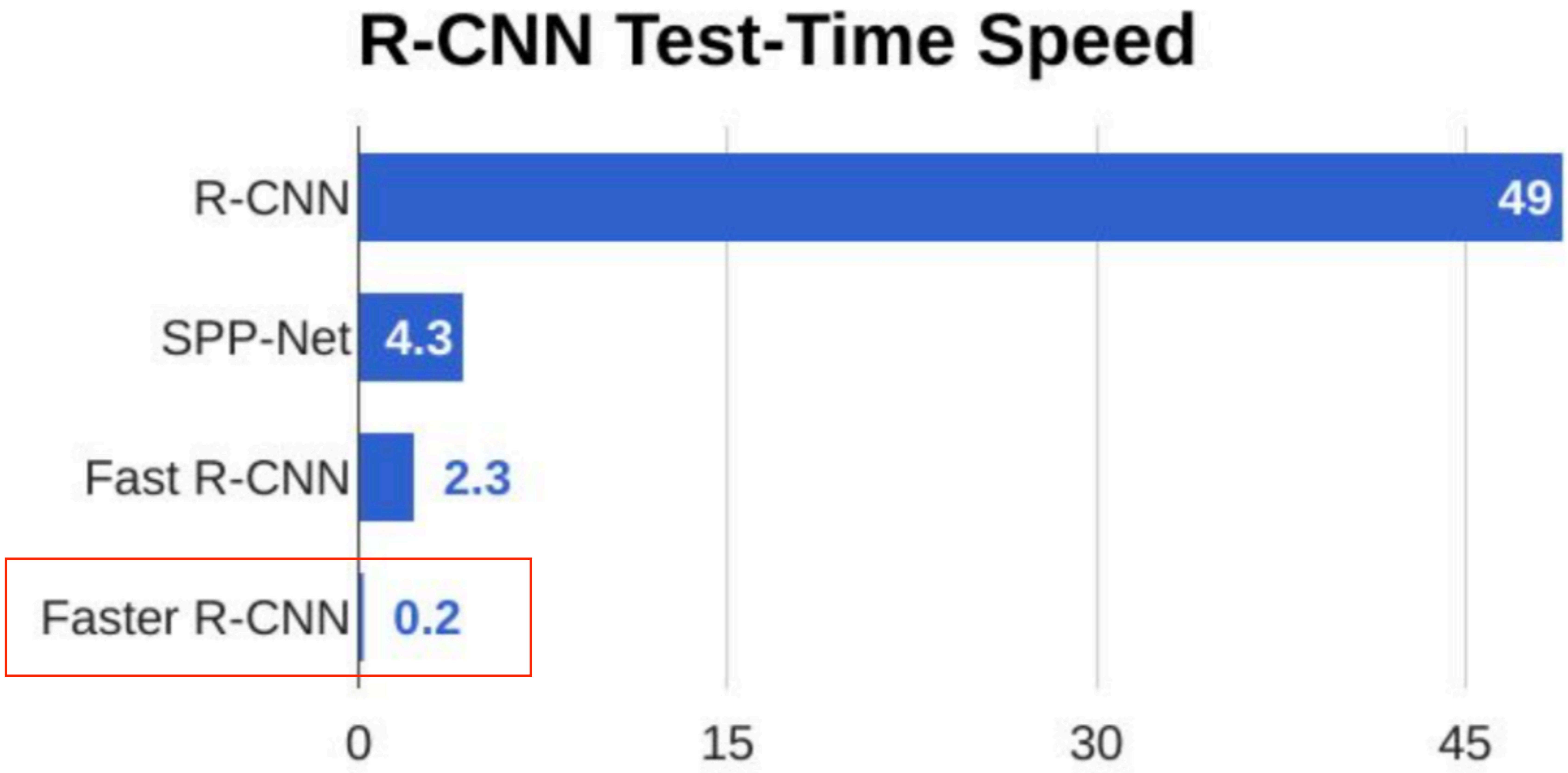
Jointly train with 4 losses:

1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates



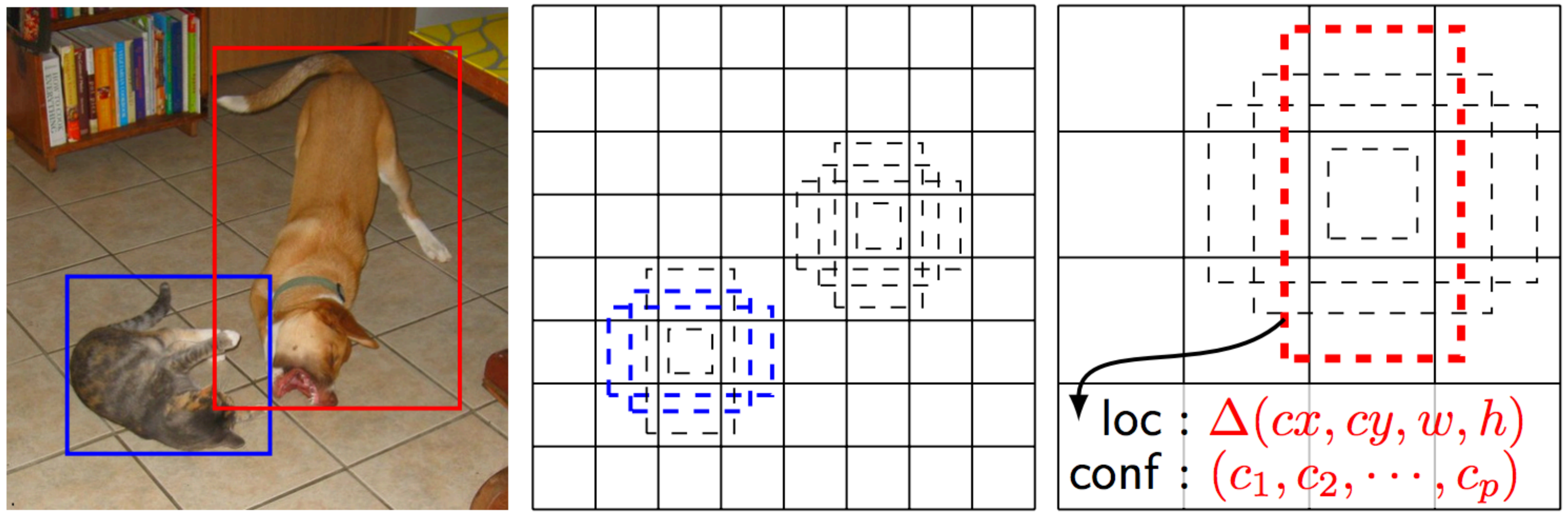
Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Faster R-CNN

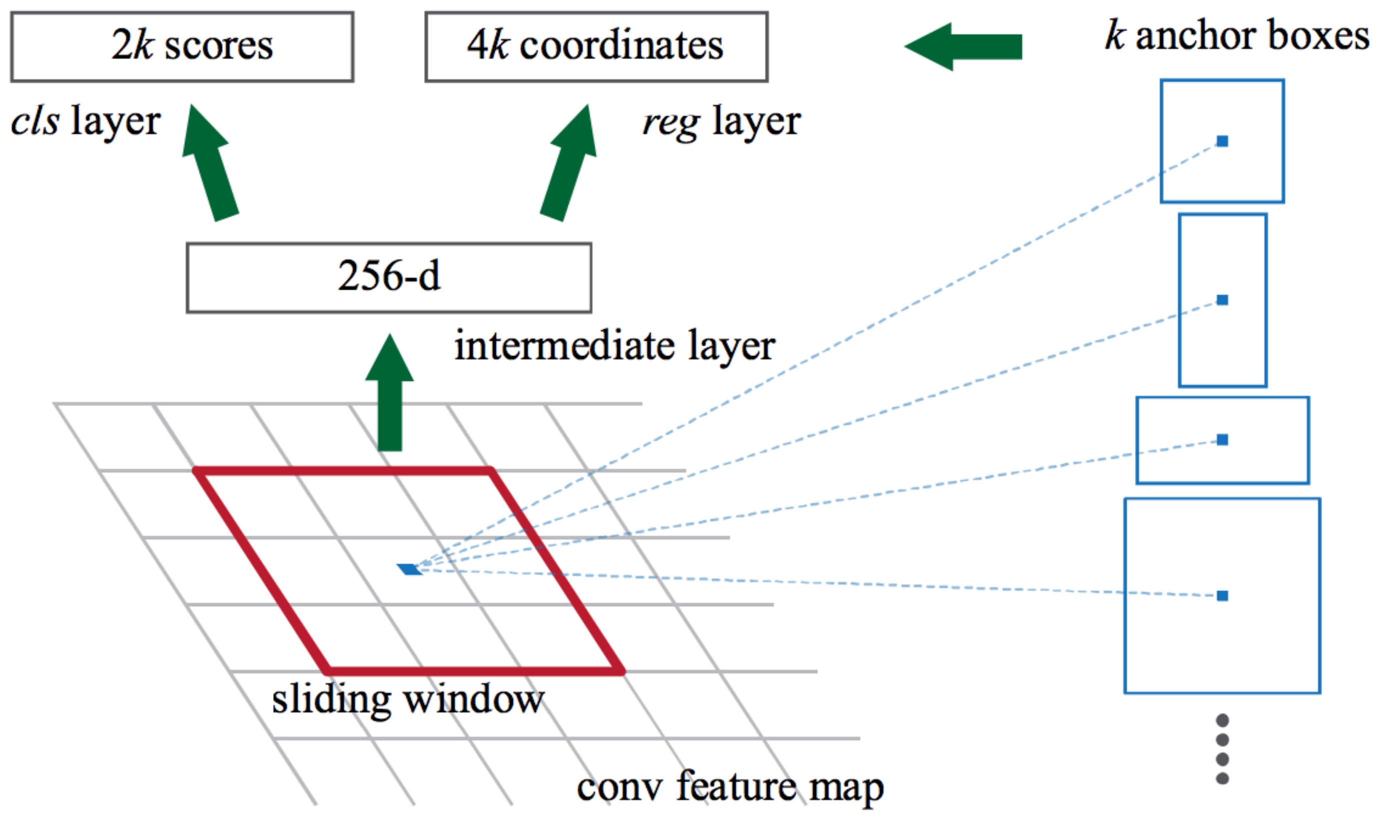

R-CNN Test-Time Speed
R-CNN
SPP-Net
Fast R-CNN
Faster R-CNN
49
4.3
2.3
0.2
0
15
30
45

Anchor as References

- Anchors: pre-defined reference boxes
  - Box regression is with reference to anchors:
    - regressing an anchor box to a ground-truth box
  - Object probability is with reference to anchors, e.g.:
    - anchors as positive samples: if IoU > 0.7 or IoU is max
    - anchors as negative samples: if IoU < 0.3

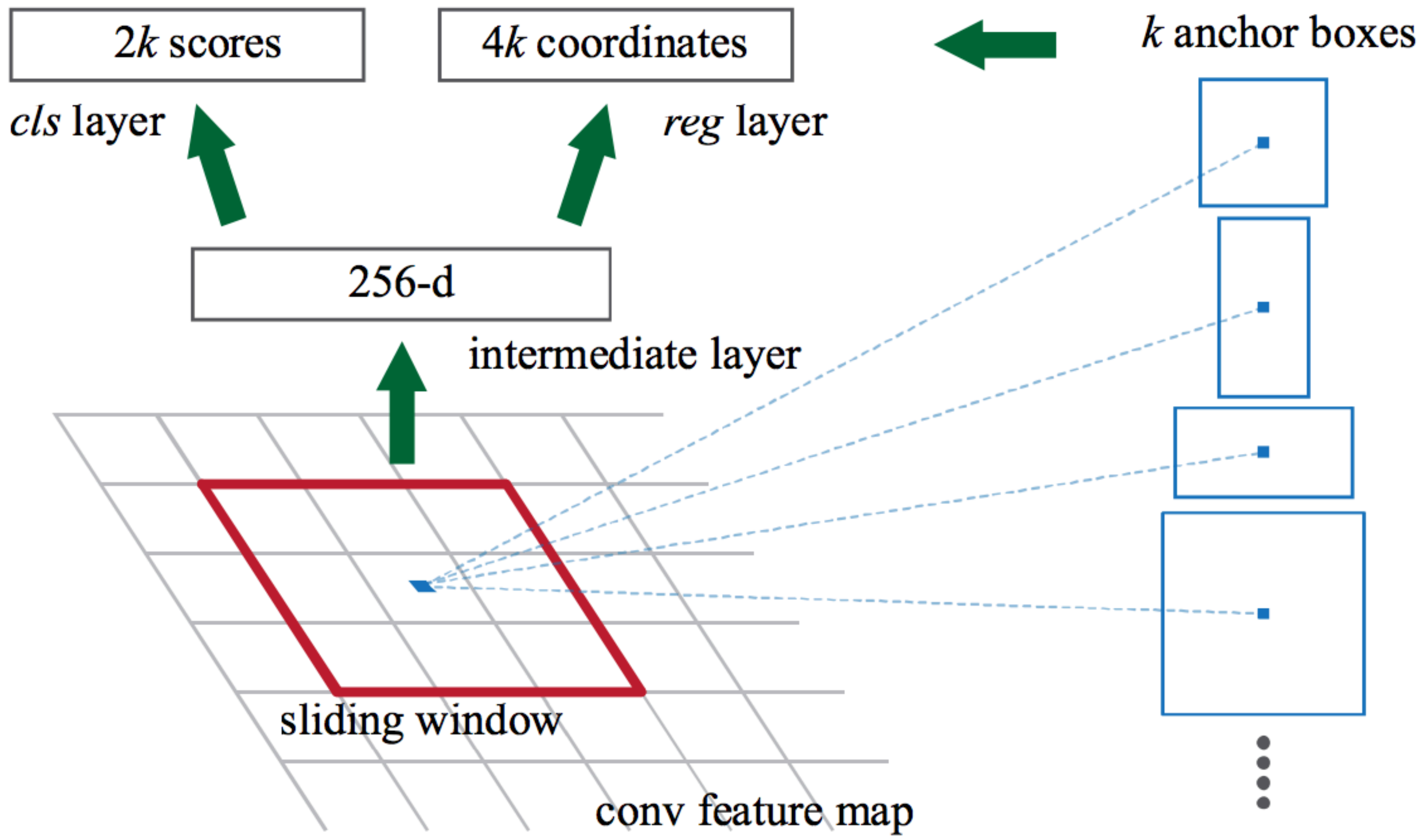


(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map



Anchor as References

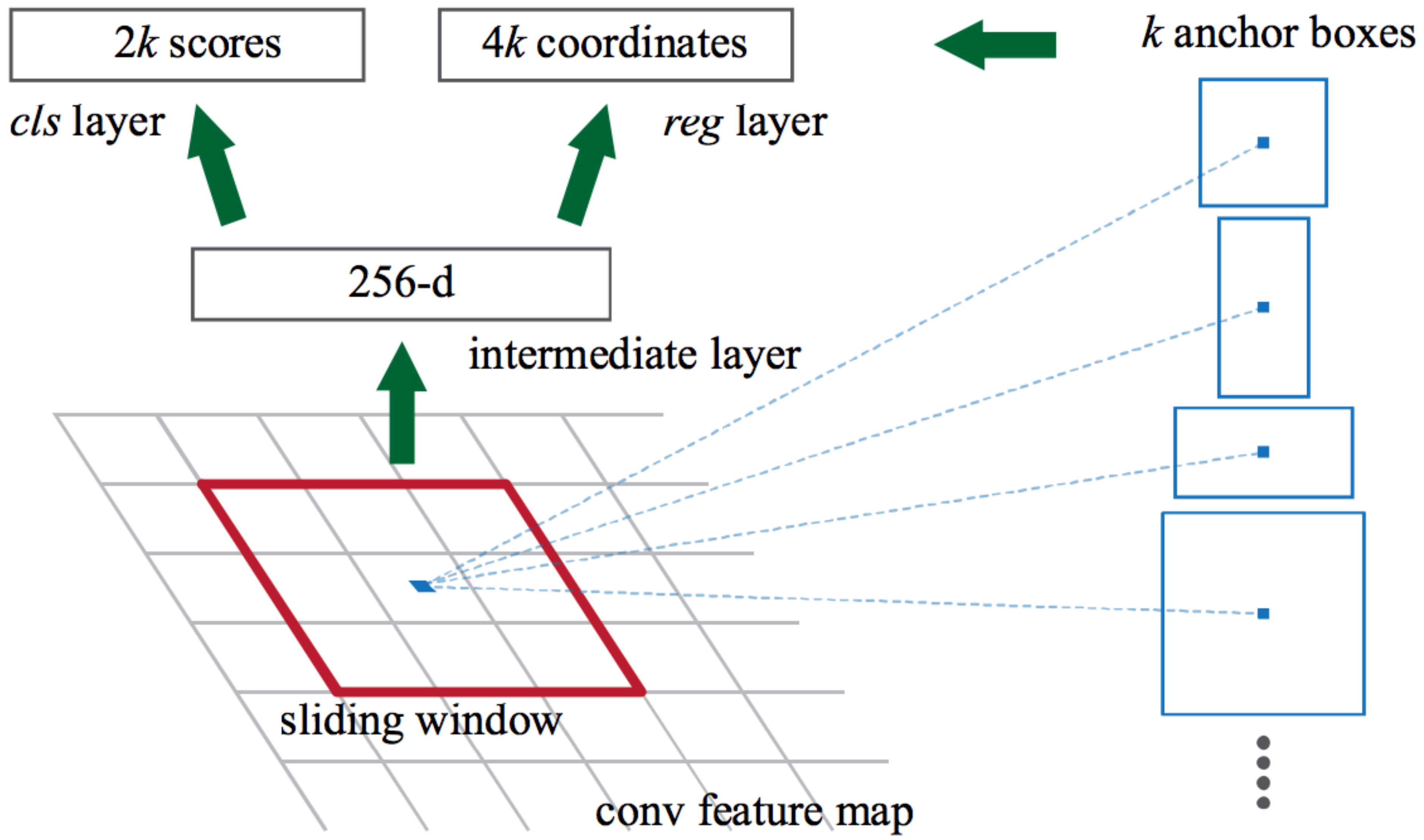
- Translation-invariant anchors:
  - the same set of anchors are used at each sliding position
  - the same prediction functions (with reference to the sliding window) are used
  - a translated object will have a translated prediction



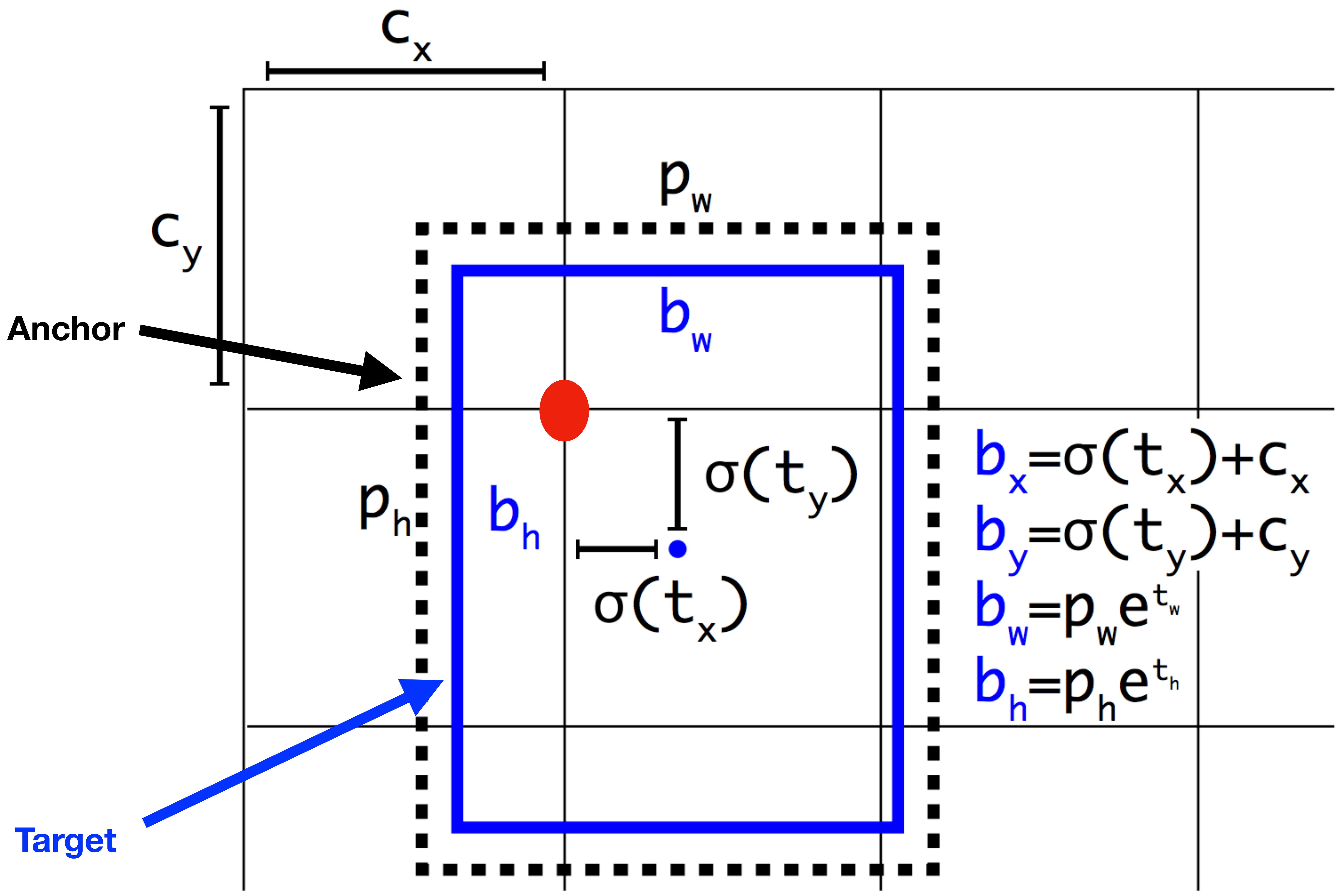
Anchor as References

- Multi-scale/size anchors:
  - multiple anchors are used at each position:

    e.g., 3 scales (128^2, 256^2, 512^2) and 3 aspect ratios (2:1, 1:1, 1:2) yield 9 anchors
  - each anchor has its own prediction function
  - single-scale features, multi-scale predictions



Anchor as References

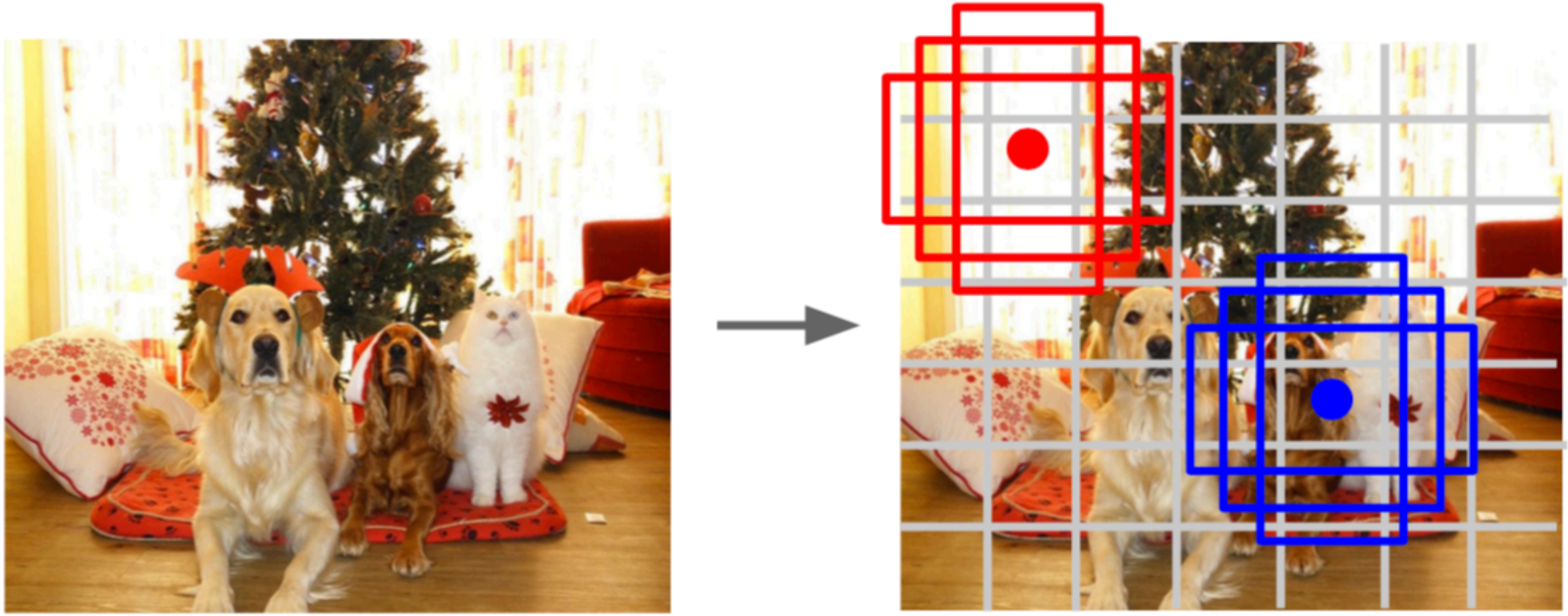


A feature map.

Anchor-Based Two-Stage Detector

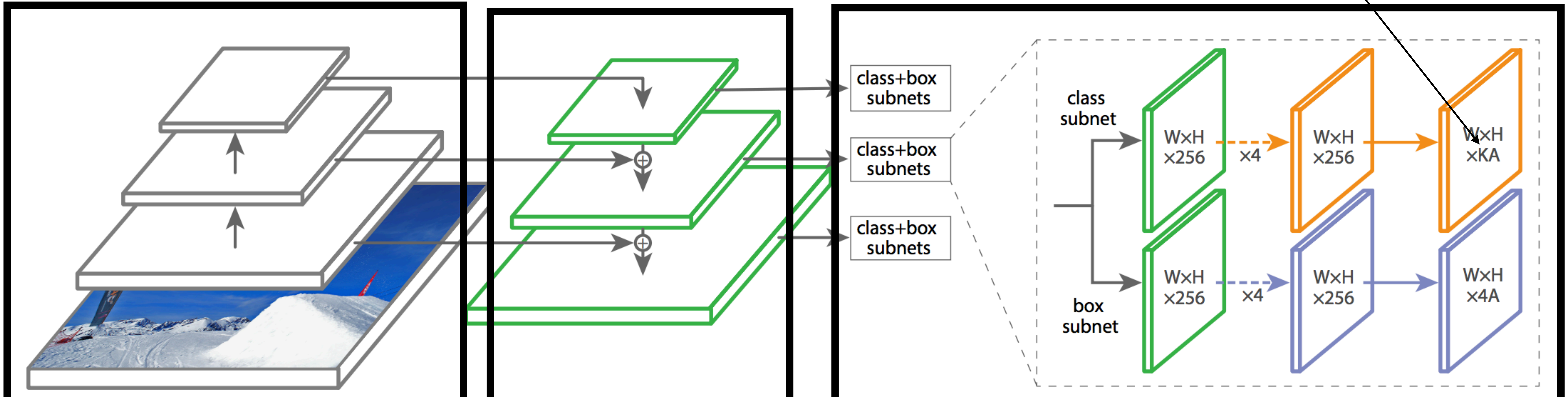
- Rich feature hierarchies for accurate object detection and semantic segmentation
  - https://arxiv.org/abs/1311.2524
- Fast R-CNN
  - https://arxiv.org/abs/1504.08083
- Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks
  - https://arxiv.org/pdf/1506.01497.pdf
- Mask RCNN
  - https://arxiv.org/pdf/1703.06870.pdf

Anchor-Based One-Stage Detector



Anchor-Based One-Stage Detector

K: The number of classes
A: The number of anchors



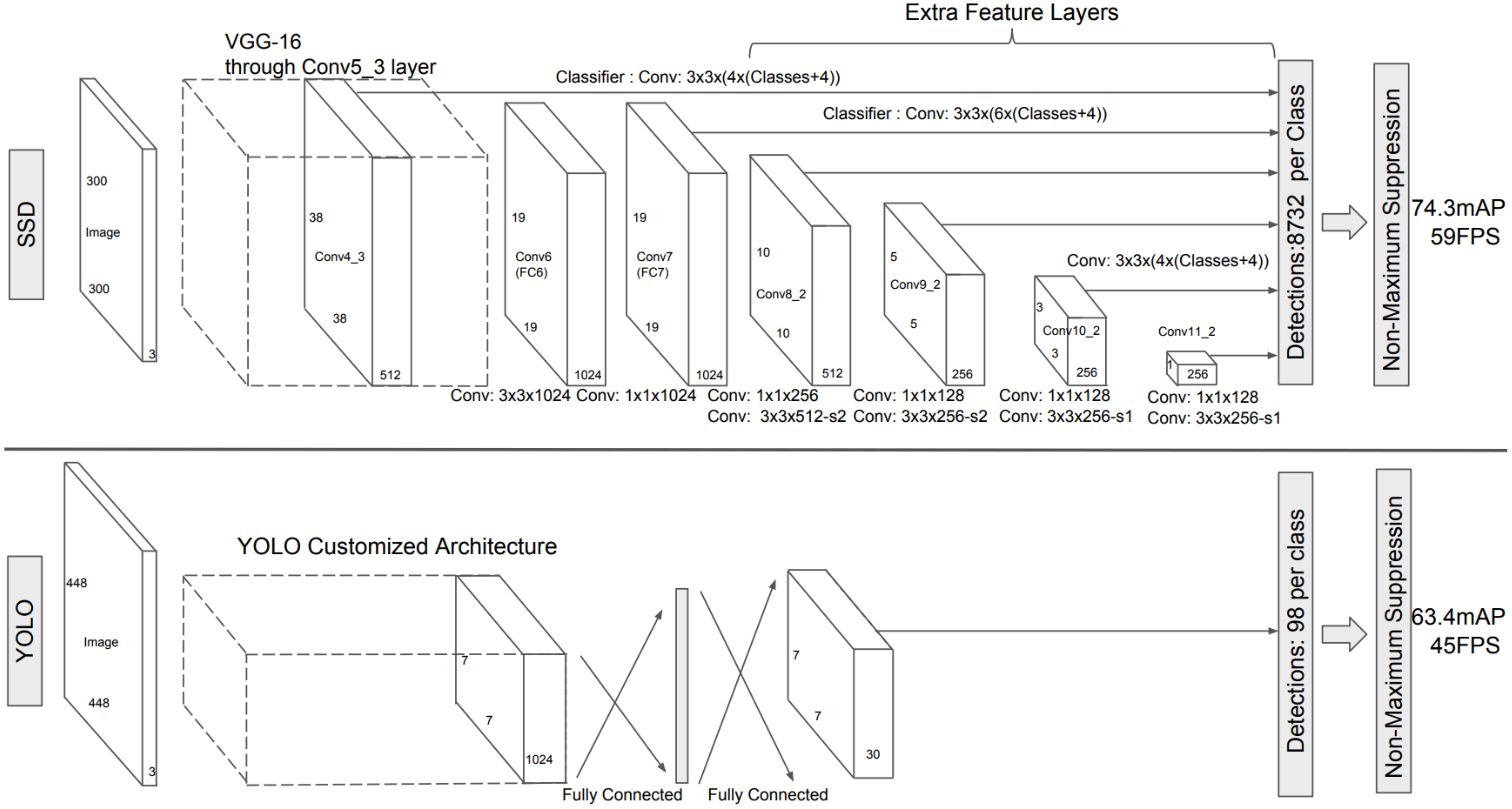
(a) ResNet | (b) feature pyramid net | (c) class subnet (top) | (d) box subnet (bottom)

Backbone: **Feature Extractor**

Neck: **Feature Enhancer**

Head: **Classification and Regression for each anchor**

Anchor-Based One-Stage Detector



Anchor-Based One-Stage Detector

- SSD: Single Shot MultiBox Detector
  - https://arxiv.org/abs/1512.02325
- You Only Look Once: Unified, Real-Time Object Detection
  - https://arxiv.org/abs/1506.02640
- YOLO9000: Better, Faster, Stronger
  - https://arxiv.org/abs/1612.08242
- Focal Loss for Dense Object Detection
  - https://arxiv.org/pdf/1708.02002.pdf

Two-Stage VS One-Stage Detector

- Two-Stage detector ≈ One-Stage detector + Refine Head
- There are some training details in two-stage detector which makes the two-stage detector may perform worse than one-stage detector in some scenario. I will leave this for you to think and discuss after reading the papers.

Anchor-Free One-Stage Detector

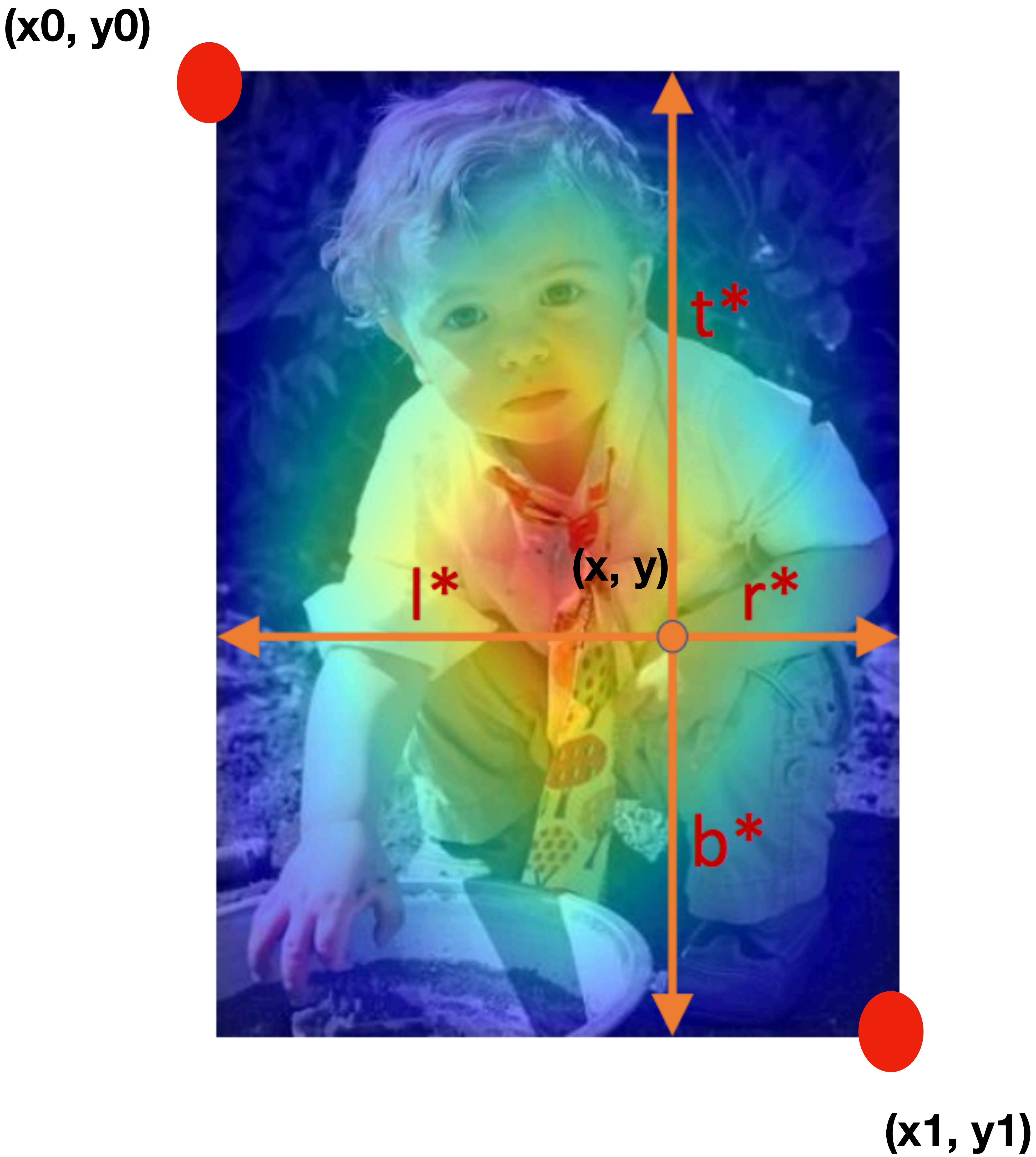
- Anchor is an important but sensitive hyper-parameters for detection performance.
  - What happens if there is a dog which is much larger than all of the anchors in a detection algorithm?
  - **It will fail to detect!**

Anchor-Free One-Stage Detector

- Point as references to avoid all hyper-parameters related to anchor boxes.

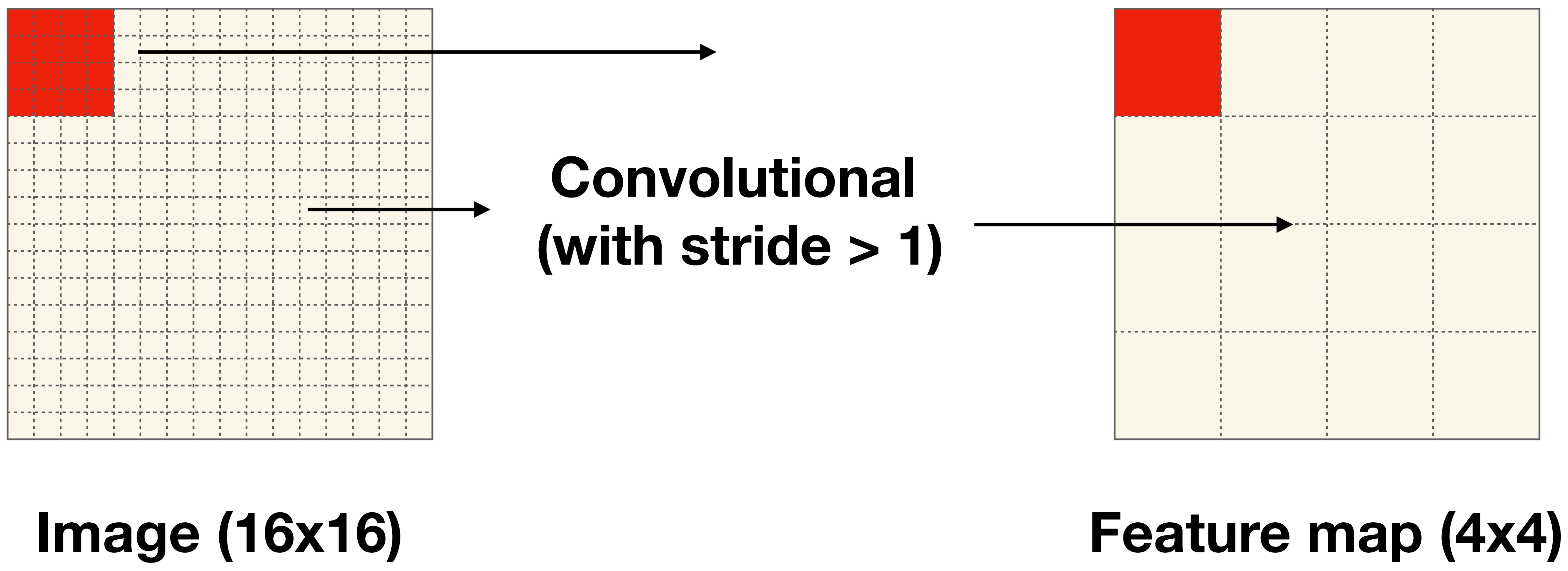
- Regression for each point in a feature map

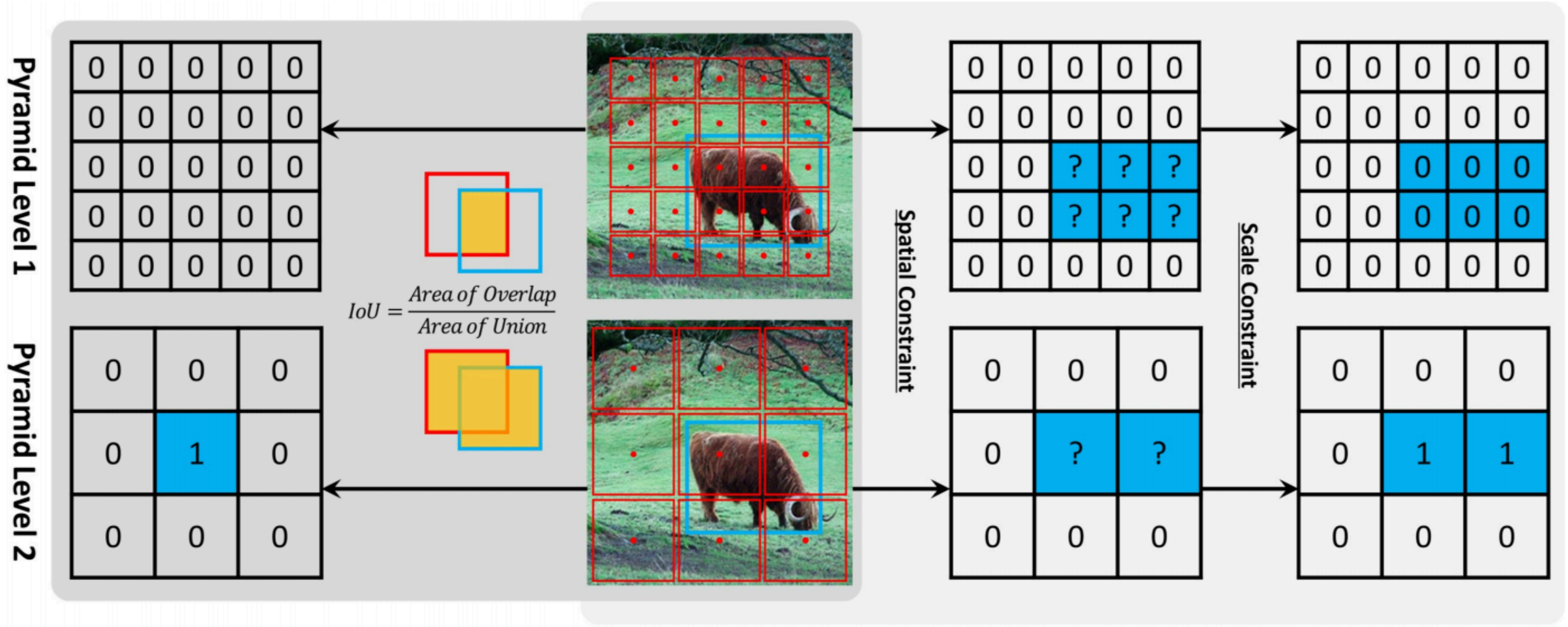
$$l^* = x - x_0^{(i)}, \;\; t^* = y - y_0^{(i)},$$
$$r^* = x_1^{(i)} - x, \;\; b^* = y_1^{(i)} - y.$$



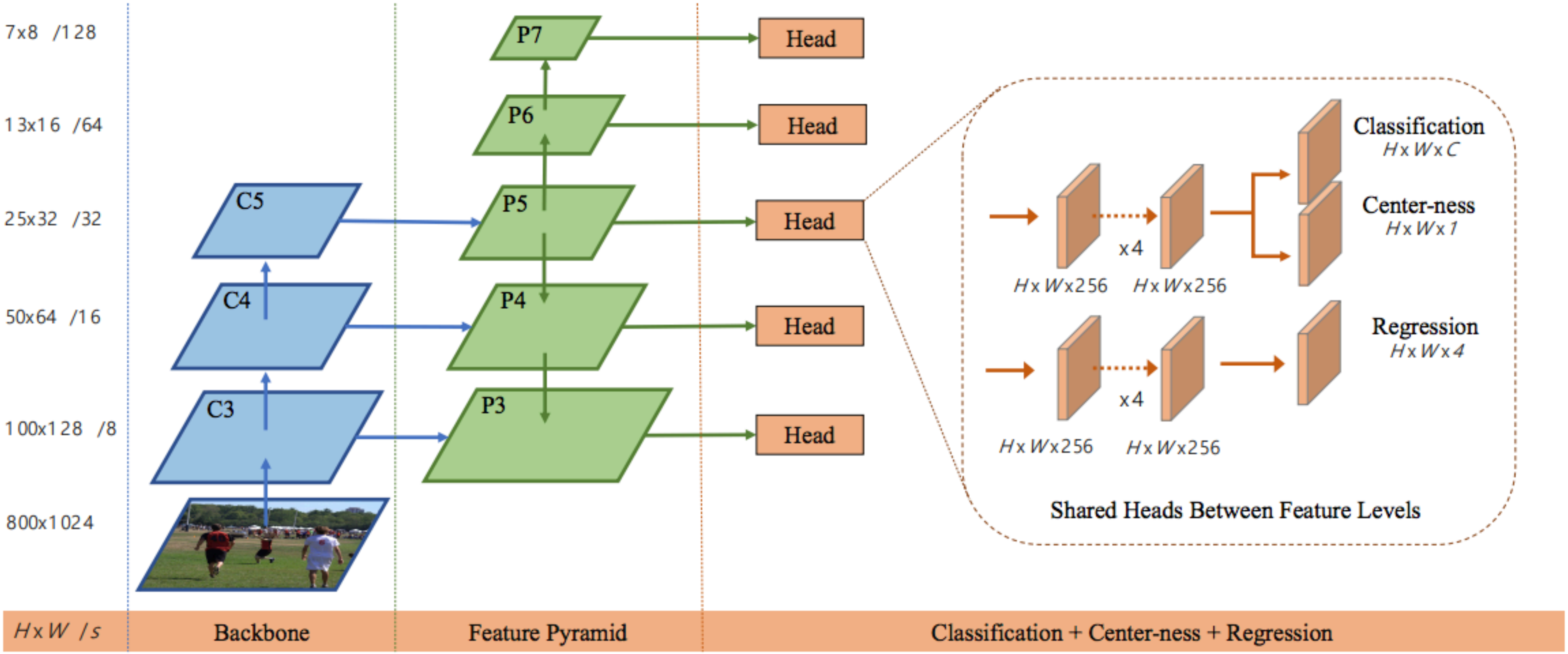
Anchor-Free One-Stage Detector

- Anchor-Based Detector will use IoU to assign the positive or negative to an anchor.
- Anchor-Free Detector has other strategy.
  - spatial constraint: center (x, y) is considered as a positive sample if it falls into any ground-truth bounding box.
  - scale constraint: Based on max(l*, r*, t*, b*) to assign ground-truth box to different head.





Anchor-Free One-Stage Detector

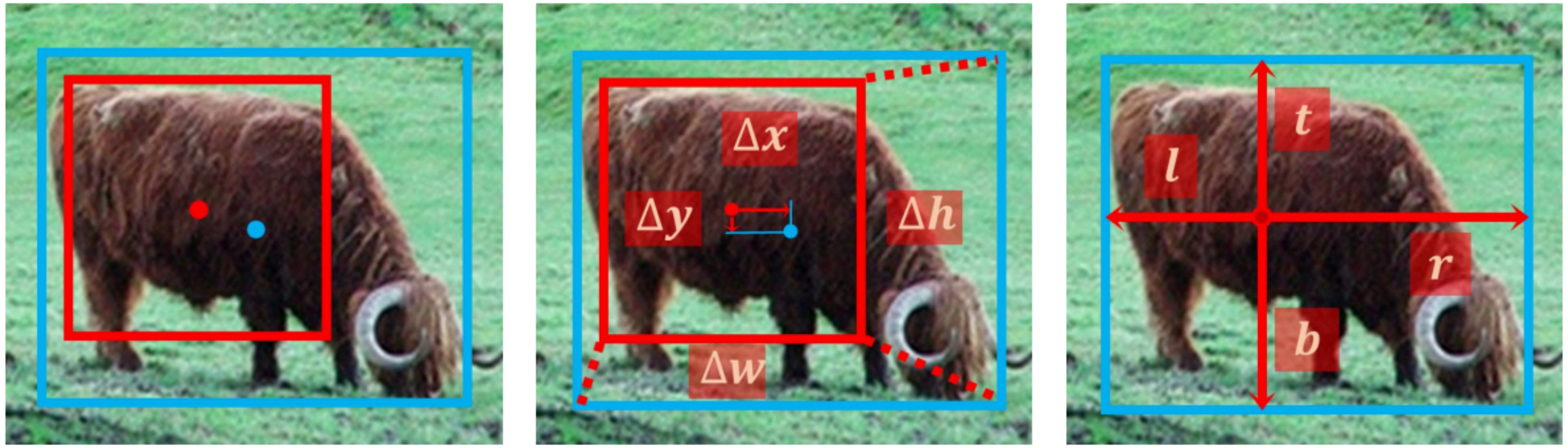


Anchor-Free One-Stage Detector

- FCOS: Fully Convolutional One-Stage Object Detection
  - https://arxiv.org/pdf/1904.01355.pdf
- CenterNet: Keypoint Triplets for Object Detection
  - https://arxiv.org/pdf/1904.08189.pdf
- FoveaBox: Beyond Anchor-based object Detector
  - https://arxiv.org/pdf/1904.03797v1.pdf

Anchor-Based VS Anchor-Free

- What is the reference
    - Anchor vs Points (or others)



(a) Positive sample (b) RetinaNet (c) FCOS

Anchor-Based VS Anchor-Free

- How to assign positive and negative
  - IoU(Anchor-based) vs Rules(Anchor-free)
  - It is nowadays an interesting research topic in object detection.
    - Bridging the Gap between Anchor-based and Anchor-free Detection via Adaptive Training Sample Selection. https://arxiv.org/abs/1912.02424
    - FreeAnchor: Learning to Match Anchors for Visual Object Detection. https://arxiv.org/abs/1909.02466

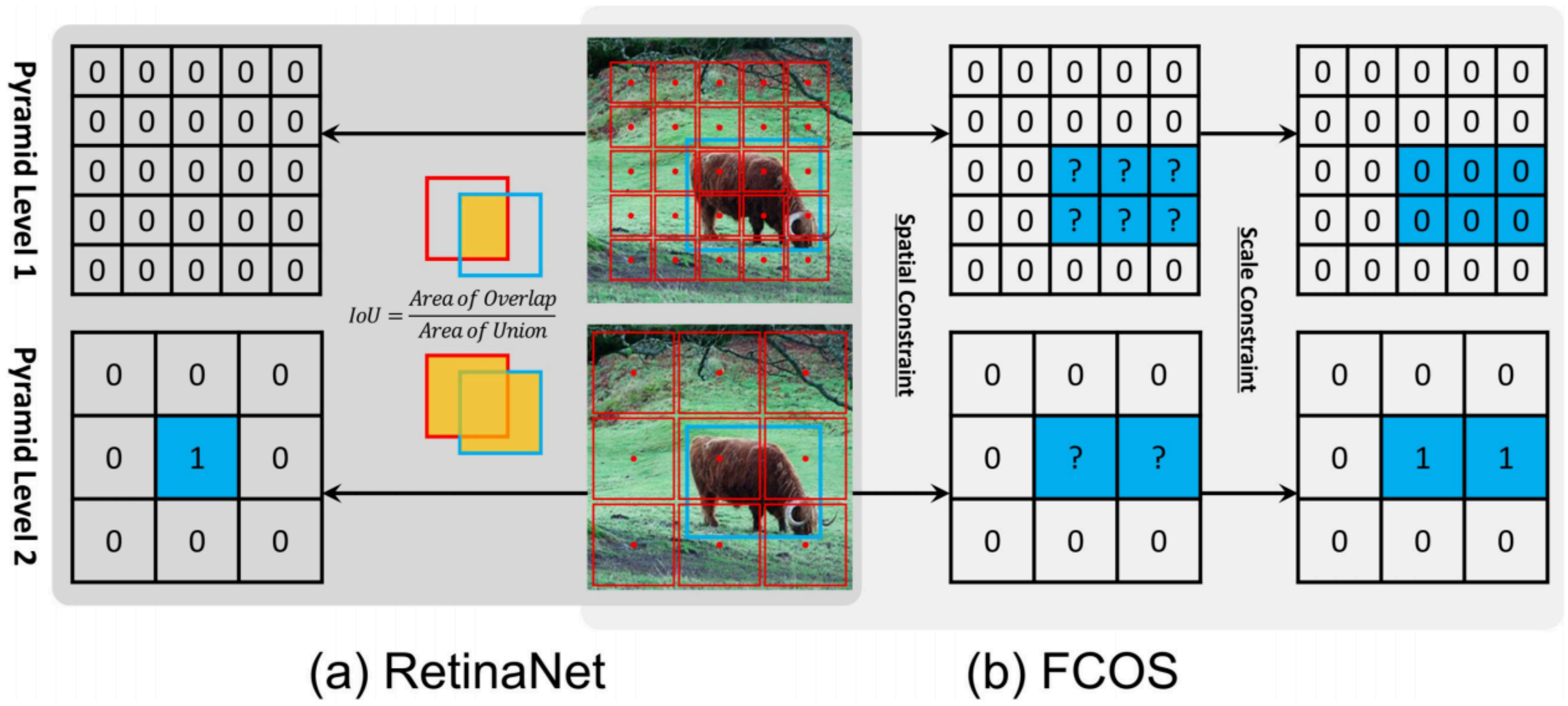


Figure 1: Definition of positives (1) and negatives (0). Blue box, red box and red point are ground-truth, anchor box and anchor point. (a) RetinaNet uses IoU to select positives (1) in spatial and scale dimension simultaneously. (b) FCOS first finds candidate positives (?) in spatial dimension, then selects final positives (1) in scale dimension.

Object Detection: lots of variable

- Base Network
  - VGG16
  - ResNet-(18/34/50/101)
  - Inception-(v1/v2/v3)
  - MobileNet-(v1/v2/v3)
  - ResNext
  - ….

- Architecture
  - two-stage
    - faster RCNN
    - Mask RCNN
  - one-stage
    - Yolo-(v1/v2/v3/v4)
    - SSD
    - RetinaNet
    - FCOS
  - ….

- Reference
  - anchor
  - anchor-free(points)
  - ….

- Takeaways
  - Two-stage is more accurate but slower
  - One-stage is faster but not as accurate

At last…

- Any question about object detection, please send me an email.
- If you are interested in computer vision, please feel free to apply for an internship or a full-time position in SmartMore.
- If you are interested, drop me an email at: exxon.yan@smartmore.com

Thanks!