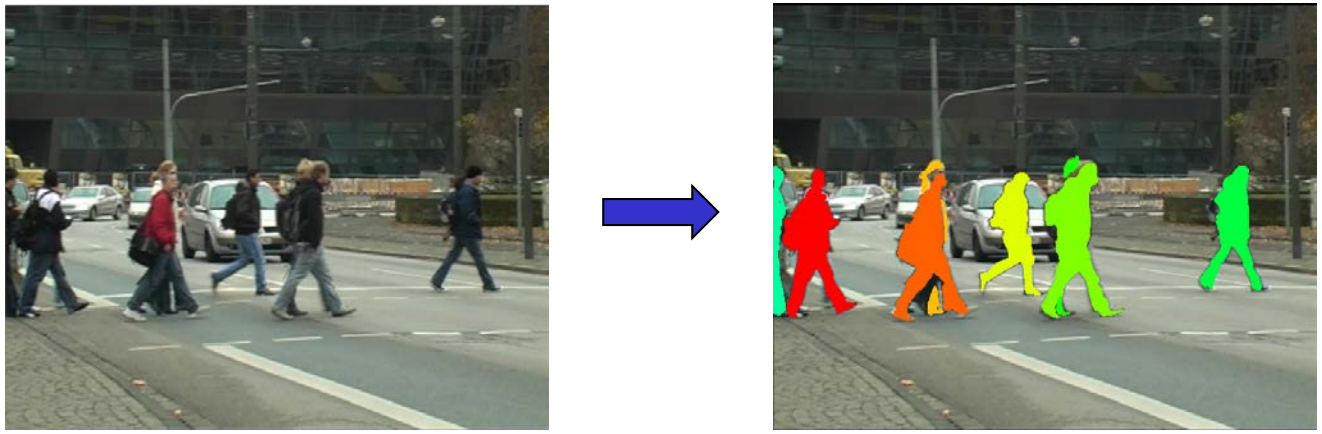
# Lecture 10: CNNs in Computer Vision – Part II

Xuming He
SIST, ShanghaiTech
Fall, 2019

# Outline



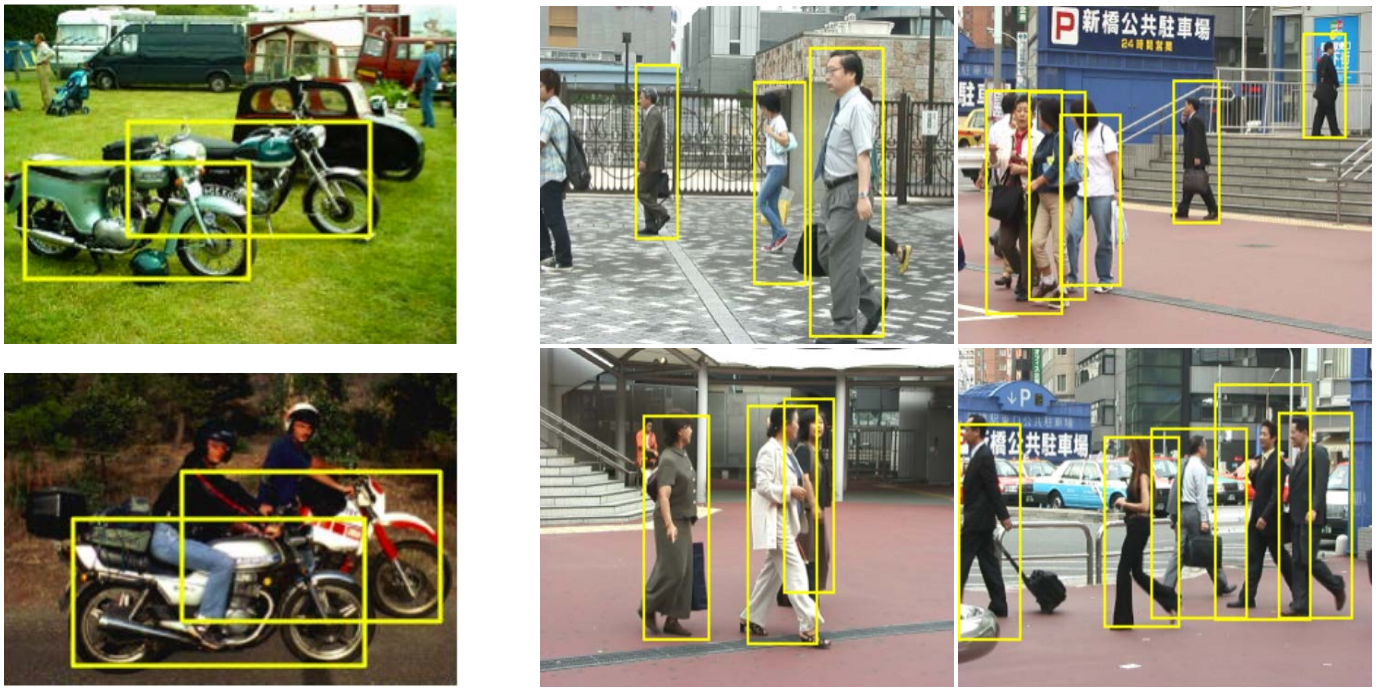
- Object Detection
- Semantic Instance Segmentation

*Acknowledgement: Feifei Li et al's cs231n notes*

# Object Detection

- Problem setup
  - Input: image, object class(es)
  - Output: object instance bounding box locations + object scores

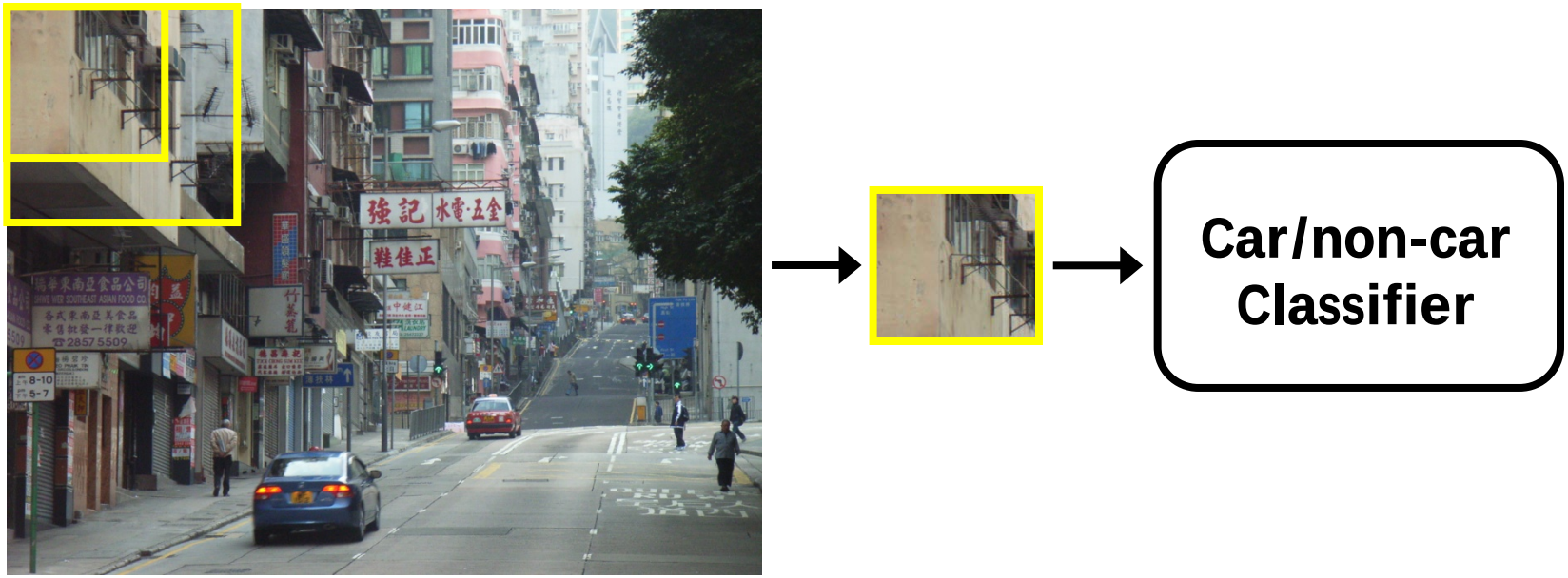
# Detection: Why it is so hard?




- Realistic scenes are crowded, cluttered, have overlapping objects.
- Object instances have large intra-class variance.

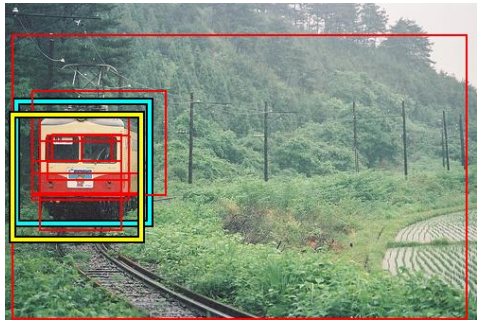
# Object Detection

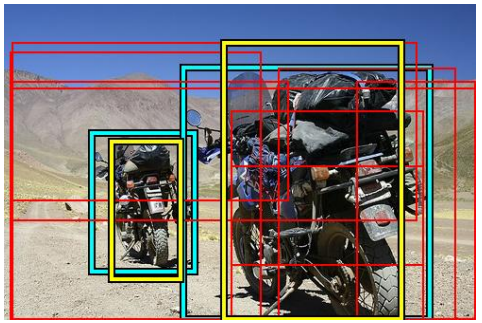
- Problem formulation
  - Object detection as a series of classification problems
  - Step1: Generate object candidate boxes
  - Step2: Scoring the object candidate with a classifier
- Example: sliding window method



# Object Detection

- CNN-based object detection
  - Reducing the search space by focusing on “object proposals” -- image regions that are likely to contain objects





  - Classifying each object proposals based on CNNs
  - Refining the object location afterwards
  - Typical method: Fast(er) R-CNN

# R-CNN

Input image

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

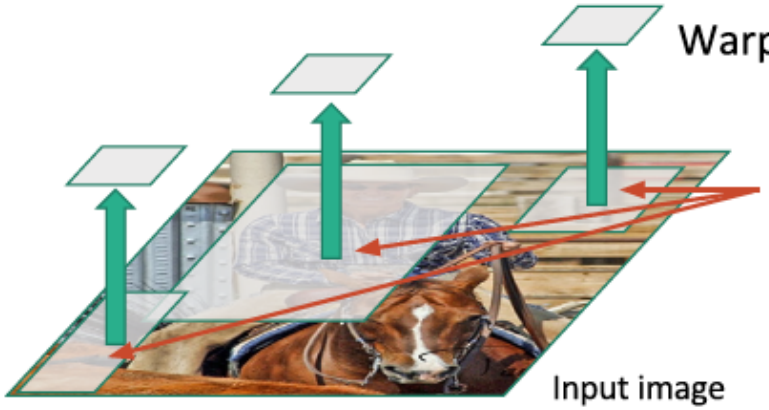
# R-CNN

Input image

Regions of Interest (RoI) from a proposal method (~2k)

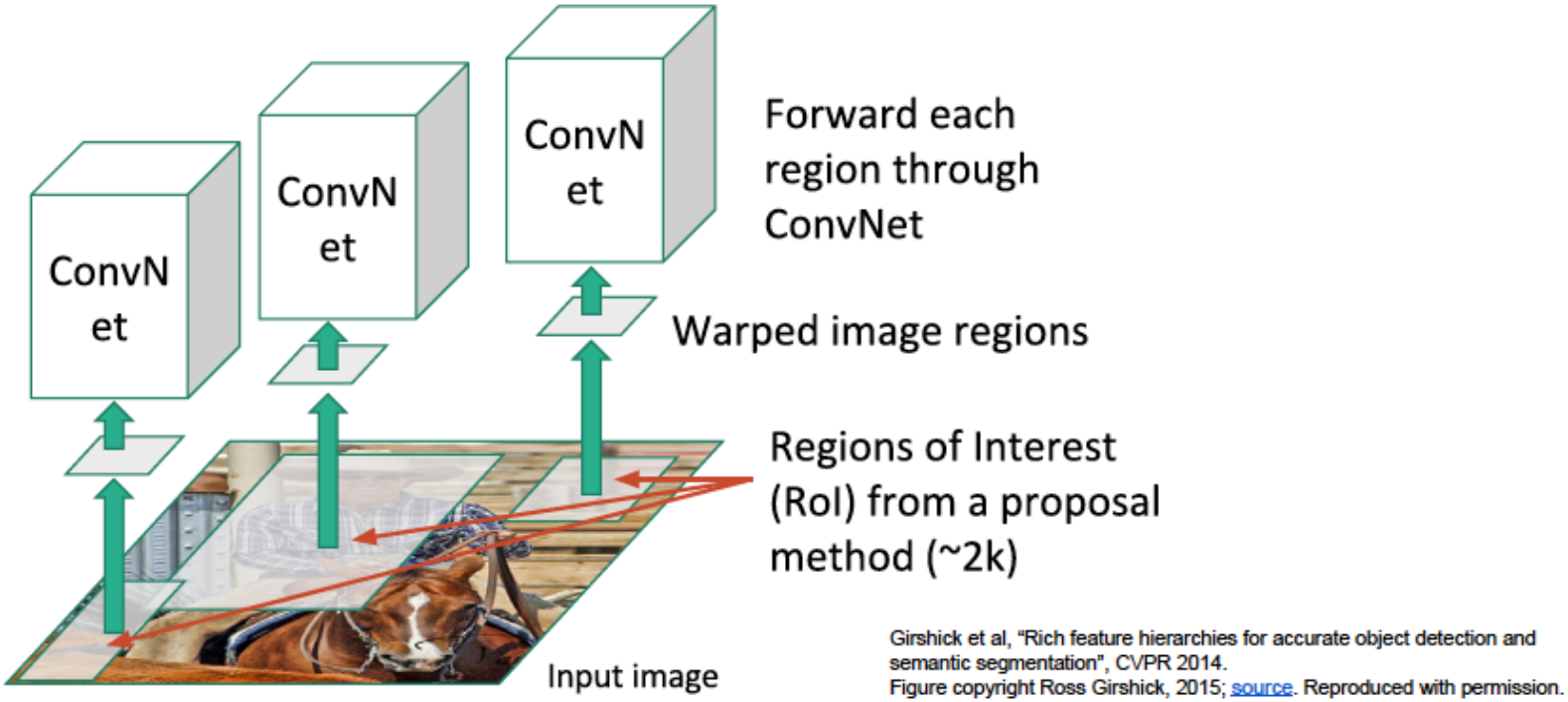
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

# R-CNN



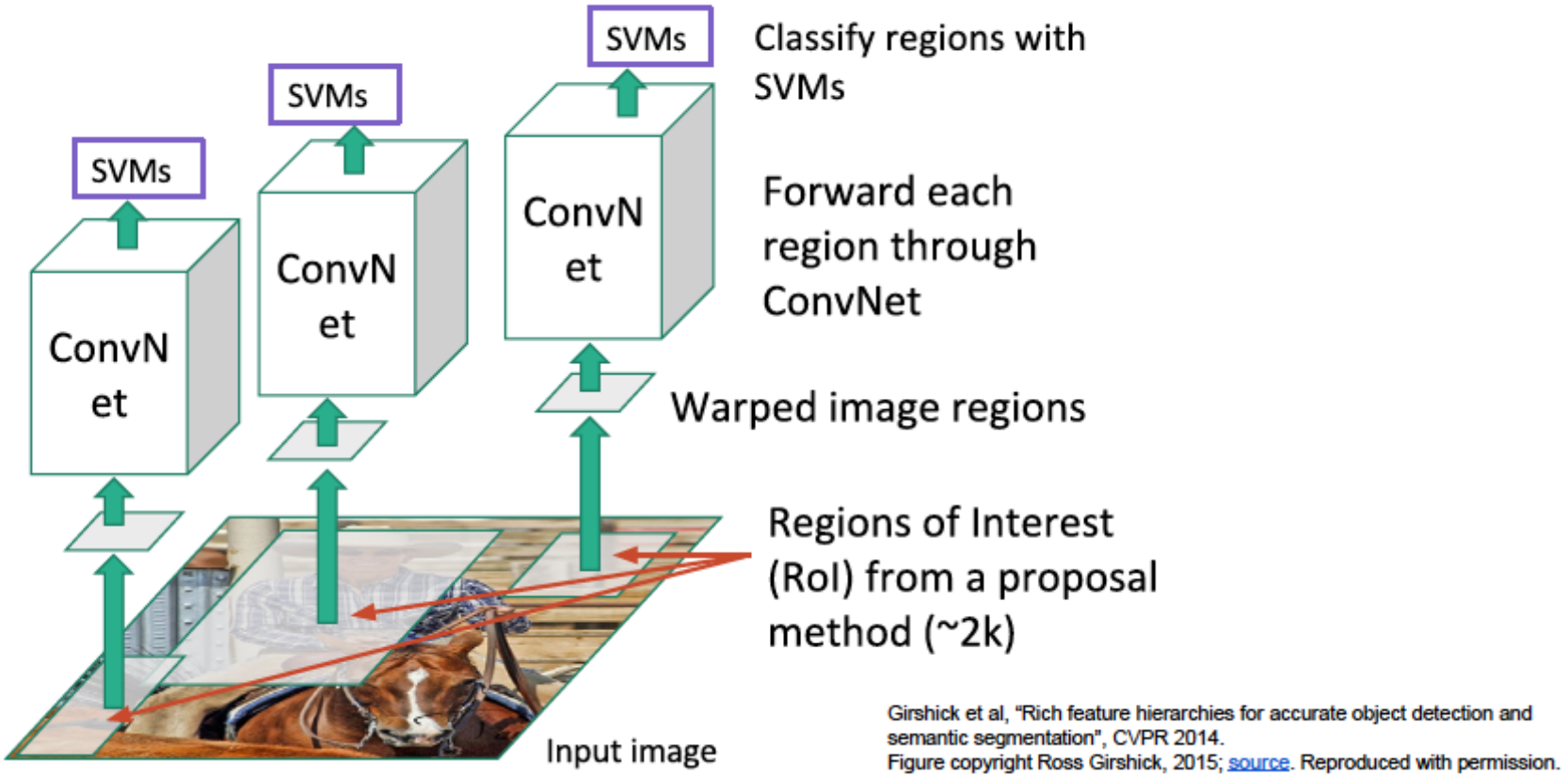

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

# R-CNN



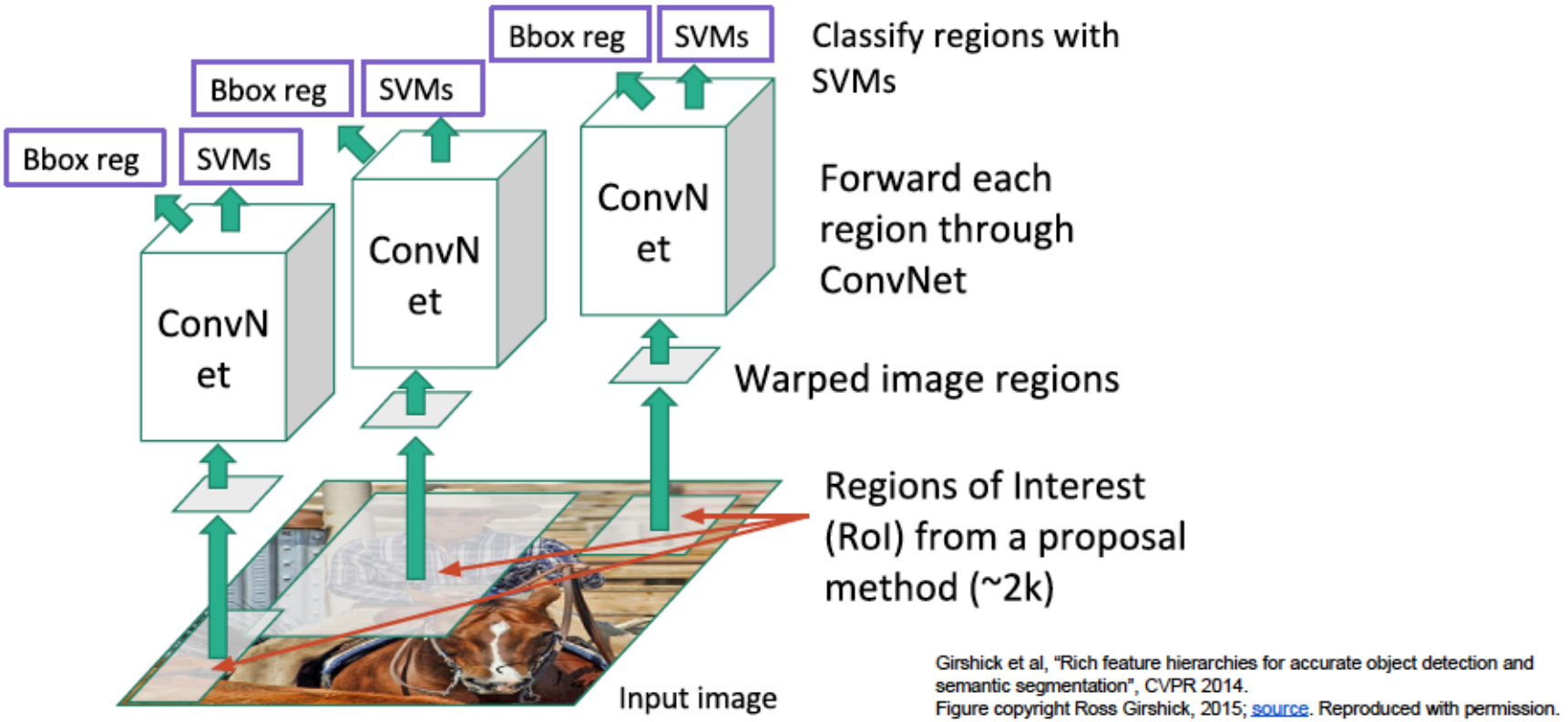

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

# R-CNN




Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

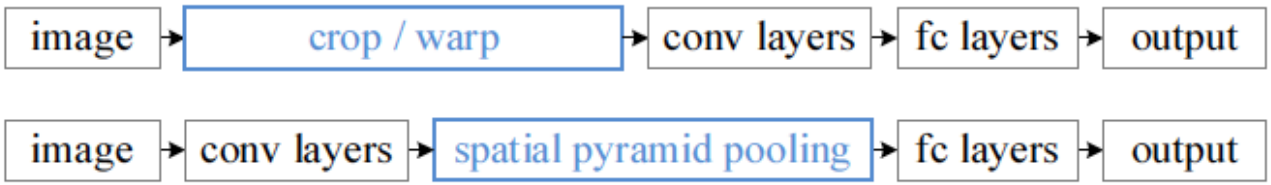
# R-CNN




Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

# R-CNN

- Problems
  - Ad hoc training objectives
    - Fine-tune network with softmax classifier (log loss)
    - Training post-hoc linear SVMs (hinge loss)
    - Training post-hoc bound-box regressions (least squares)
  - Training is slow (84h), takes a lot of disk space
  - Inference (detection) is slow
    - 47s / image with VGG16 [Simonyan & Zisserman. ICLR15]
    - Fixed by SPP-net [He et al. ECCV14]

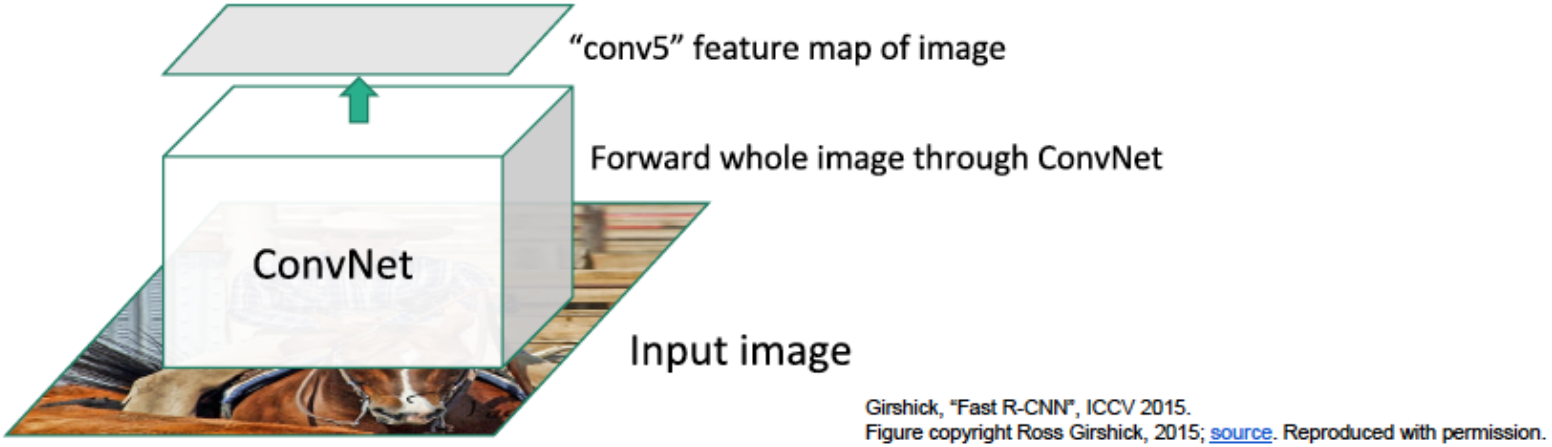


# Fast R-CNN

Input image

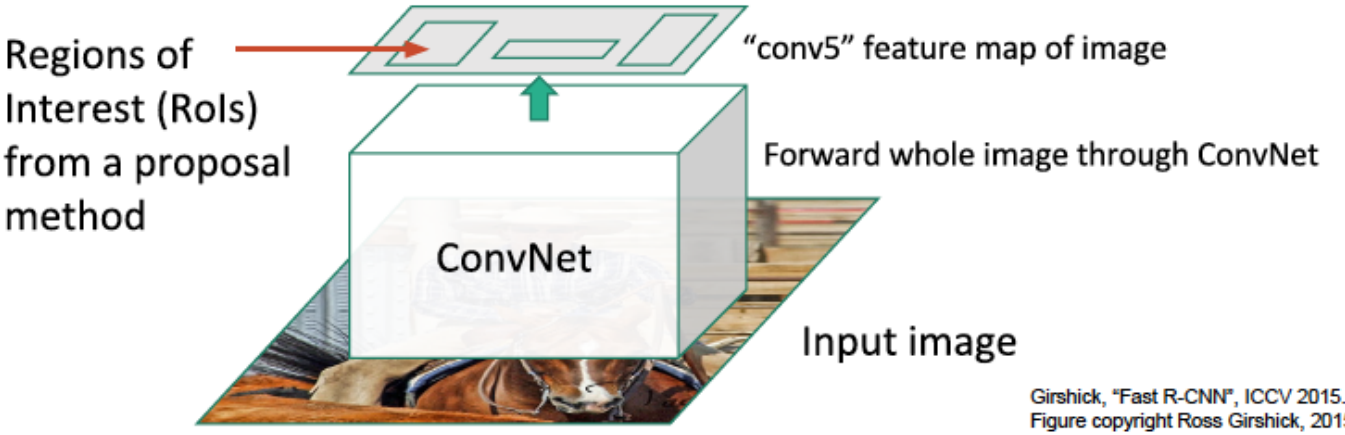
Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

# Fast R-CNN



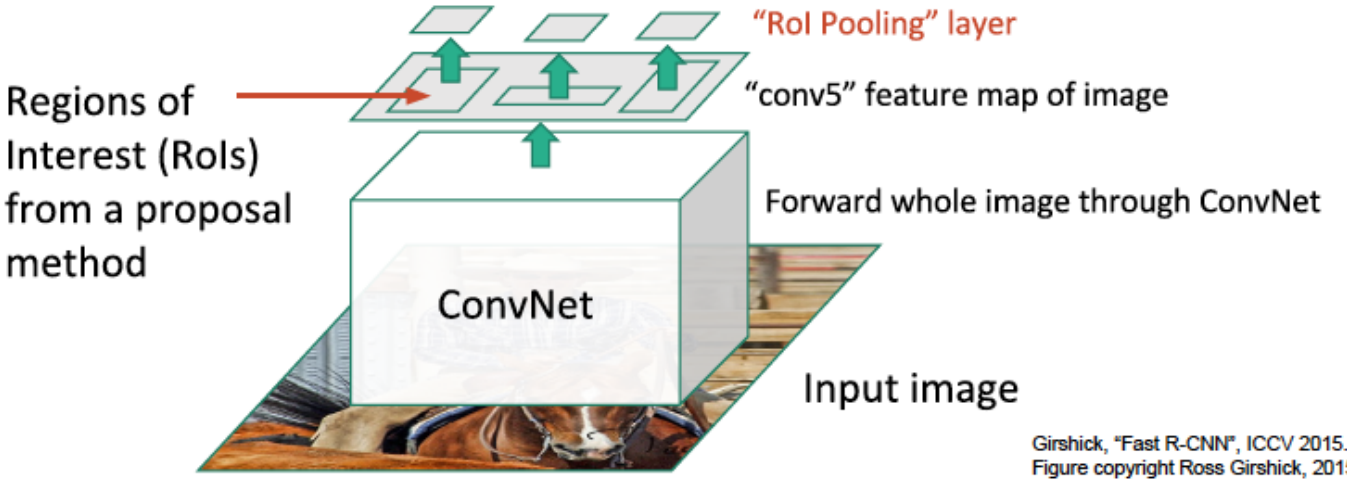

Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

# Fast R-CNN



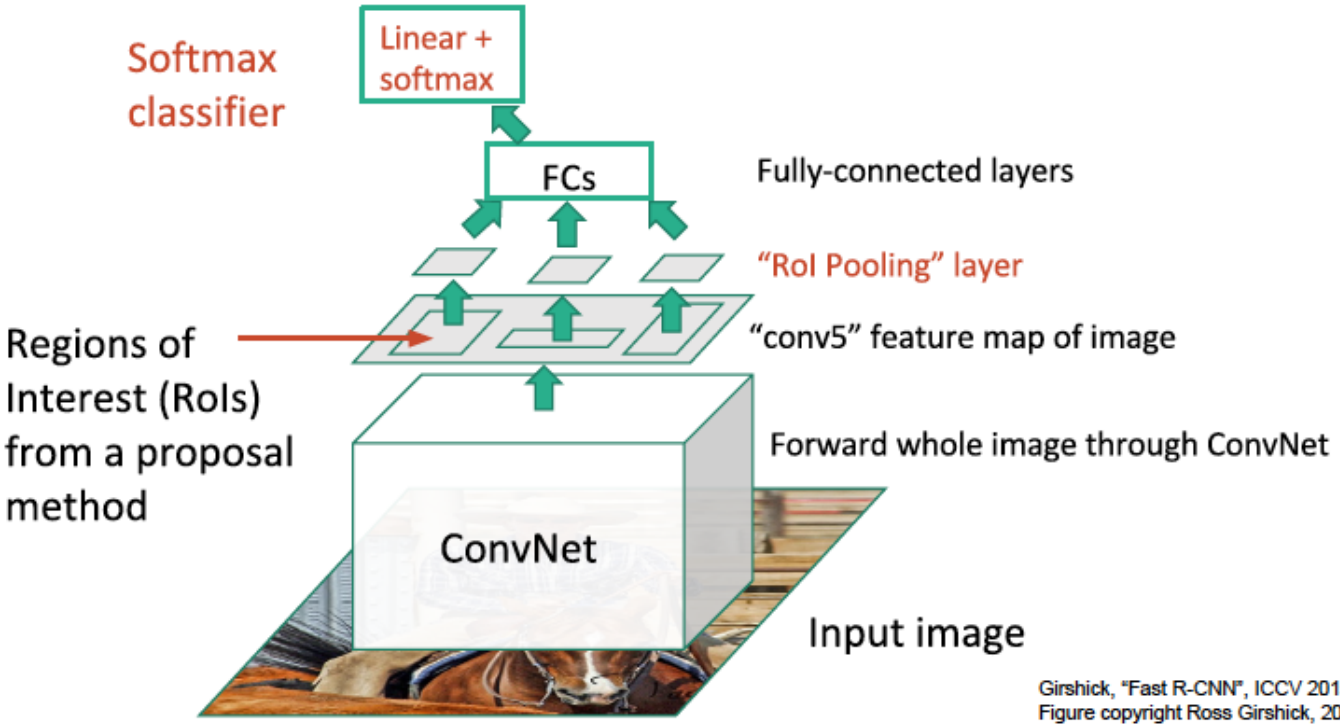

Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

# Fast R-CNN



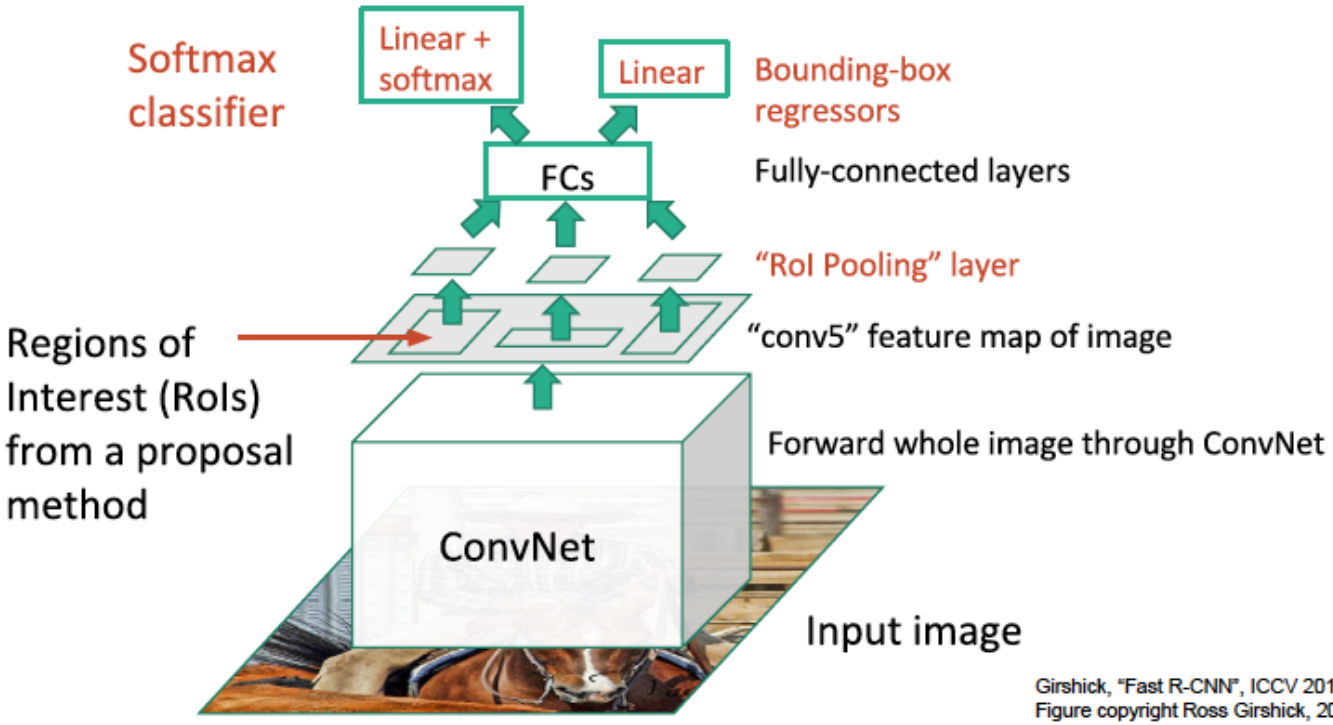

Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

# Fast R-CNN




Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

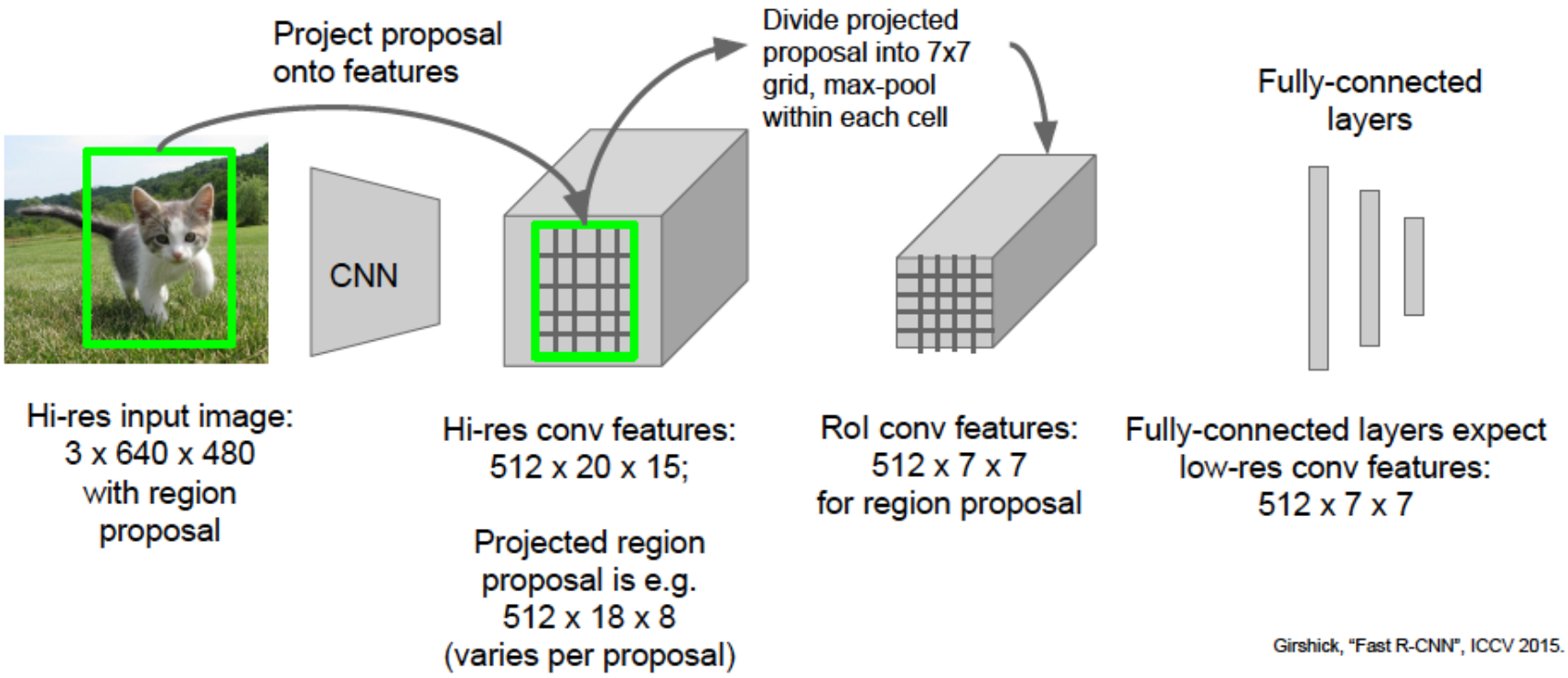
# Fast R-CNN




Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

# Fast R-CNN

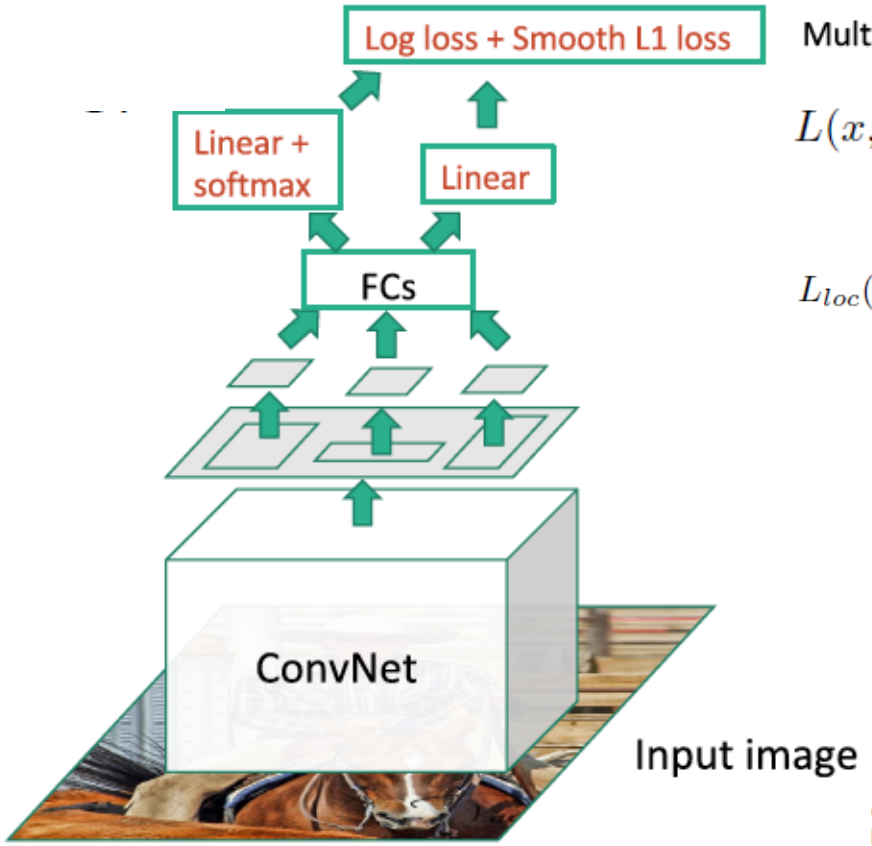
- RoI Pooling




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

# Fast R-CNN

- Deep network training




Multi-task loss

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^{k} \text{smooth}_{\text{L1}}(l_i^m - \hat{g}_j^m)$$

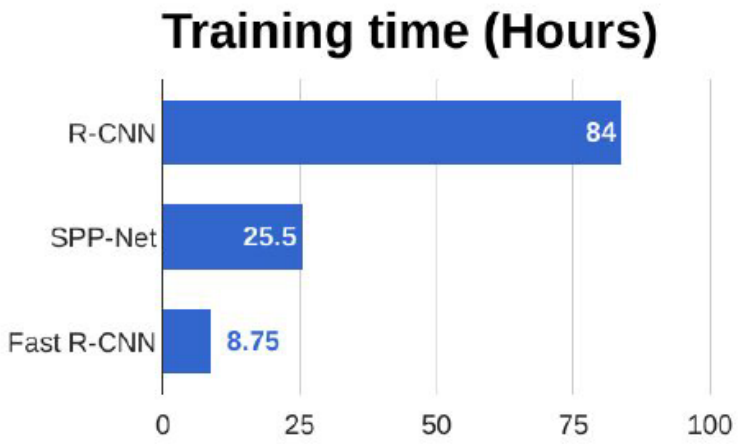
$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h$$

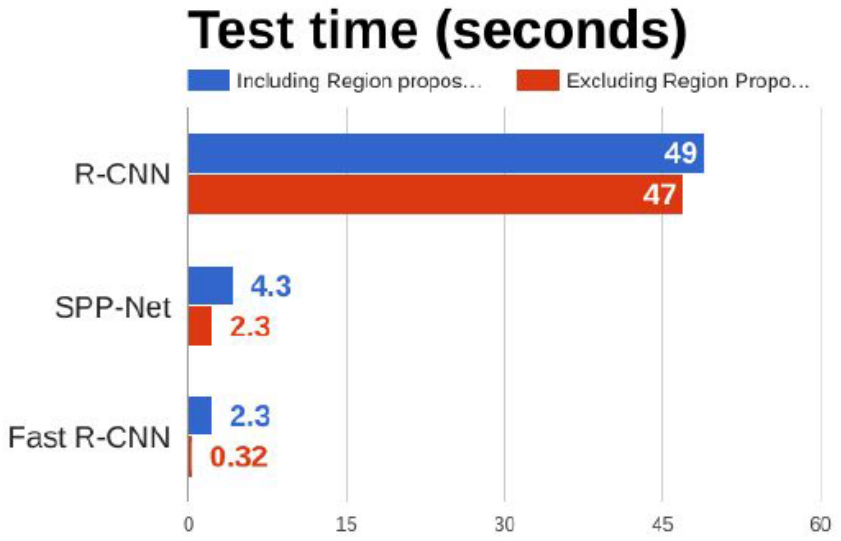
$$\hat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right) \qquad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right)$$

Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

# Speed Comparison

## R-CNN vs SPP vs Fast R-CNN




Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
He et al, "Spatial pyramid pooling in deep convolutional networks for visual recognition", ECCV 2014
Girshick, "Fast R-CNN", ICCV 2015

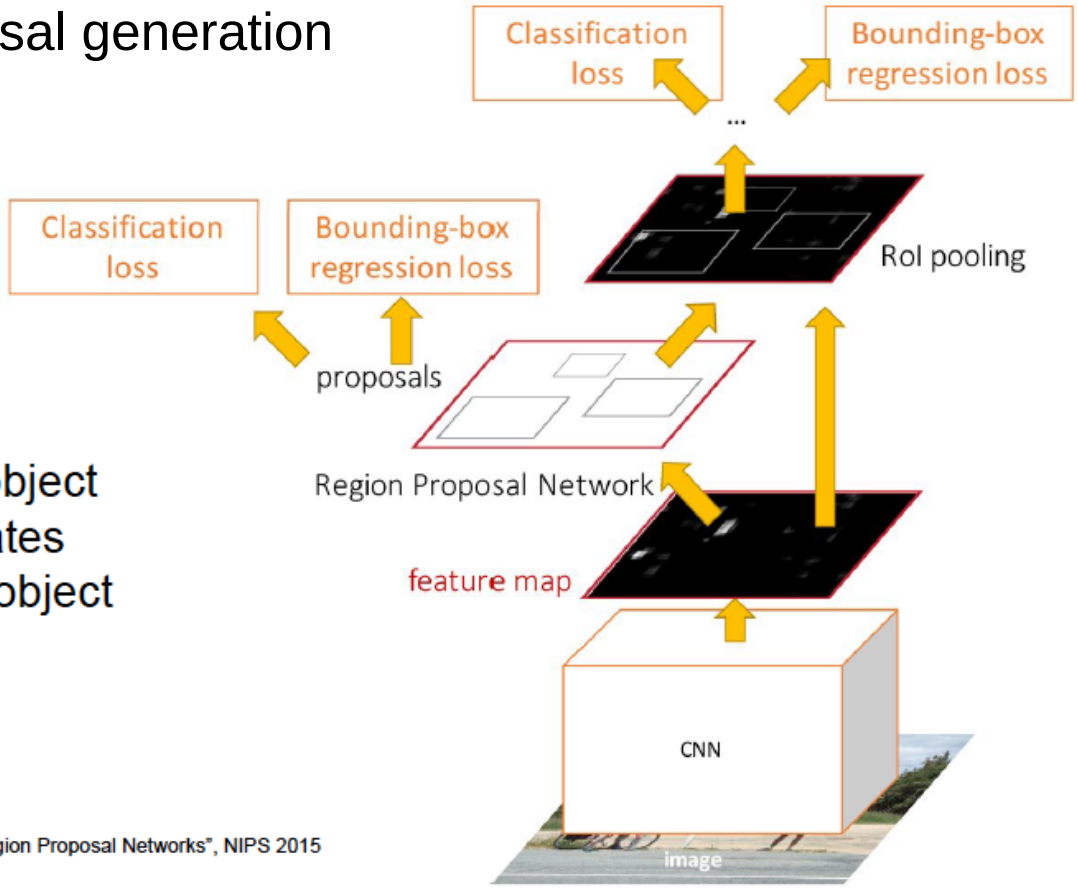
# Faster R-CNN

- Region proposal network
  - Make CNN do proposal generation

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

Jointly train with 4 losses:

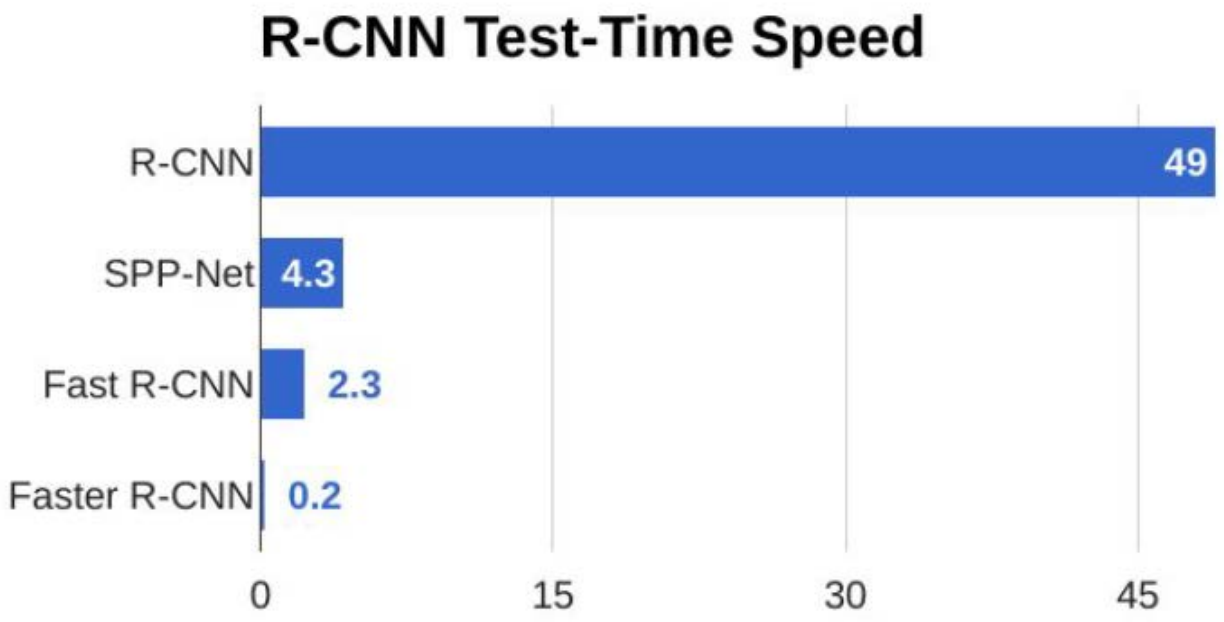
1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates




Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

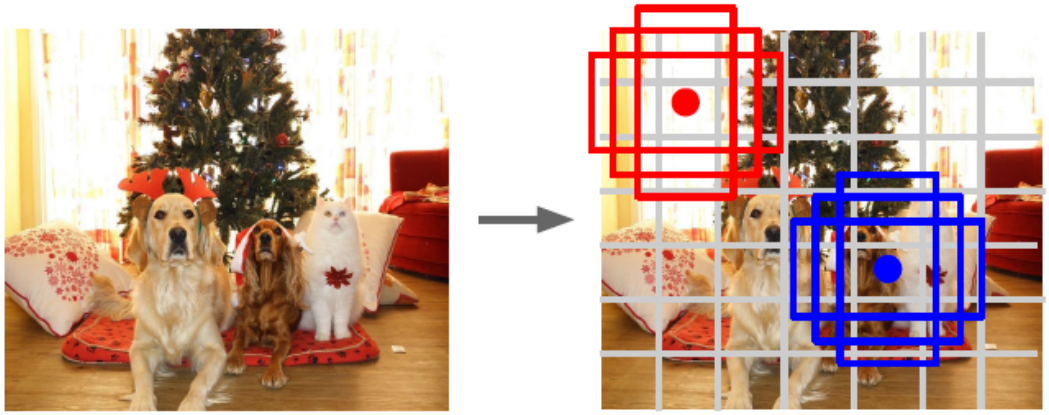
# Faster R-CNN

- Speed comparison



# Object Detection without Proposals

- YOLO / SSD
- Alternative formulation: regression task



Input image
3 x H x W

Divide image into grid
7 x 7

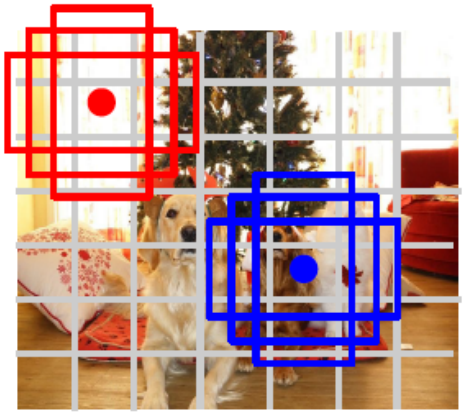
Image a set of **base boxes** centered at each grid cell
Here B = 3

Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016

# Object Detection without Proposals

- YOLO / SSD
- Alternative formulation: regression task

Input image
3 x H x W



Divide image into grid
7 x 7

Image a set of **base boxes** centered at each grid cell
Here B = 3

Within each grid cell:

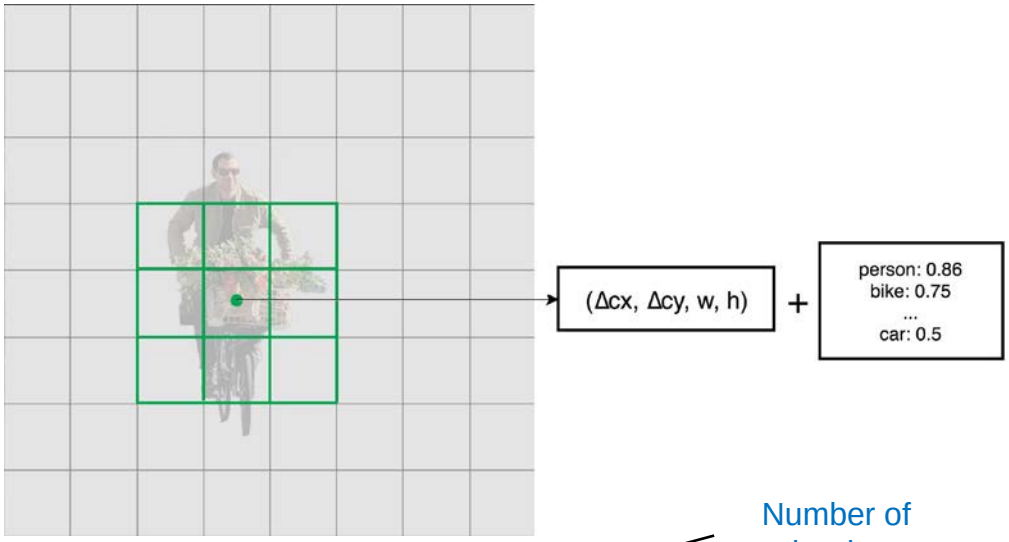
- Regress from each of the B base boxes to a final box with 5 numbers: (dx, dy, dh, dw, confidence)
- Predict scores for each of C classes (including background as a class)

Output:
7 x 7 x (5 * B + C)

Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016

# Object Detection without Proposals

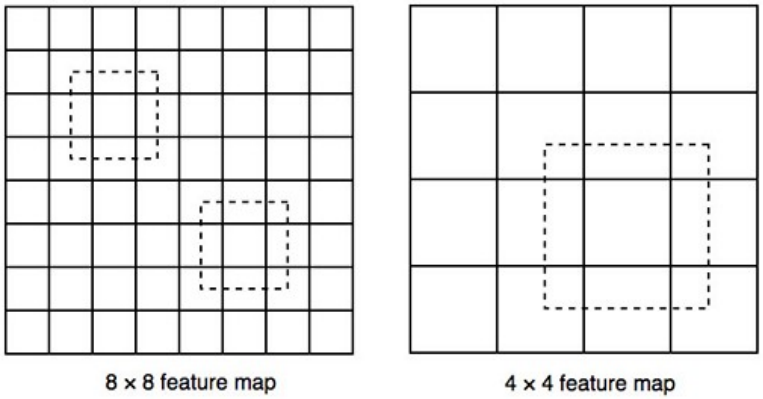
- YOLO / SSD
- Alternative formulation: regression task

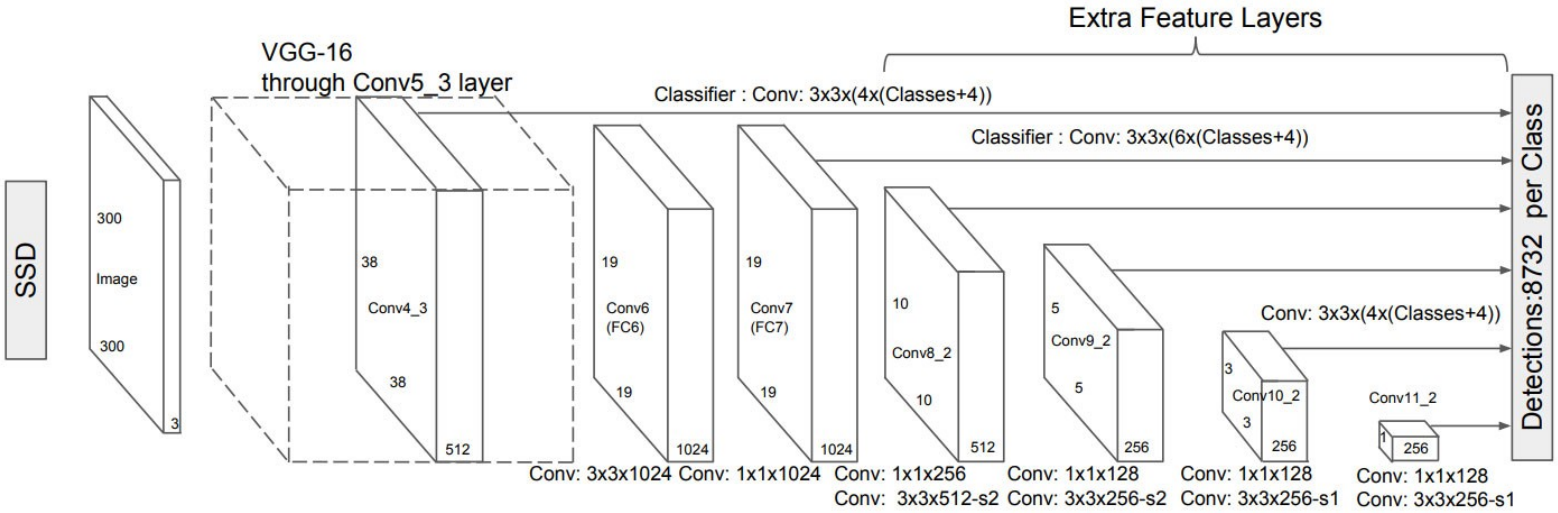



https://medium.com/@jonathan_hui/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06

# Object Detection without Proposals

- SSD: multi-scale feature maps



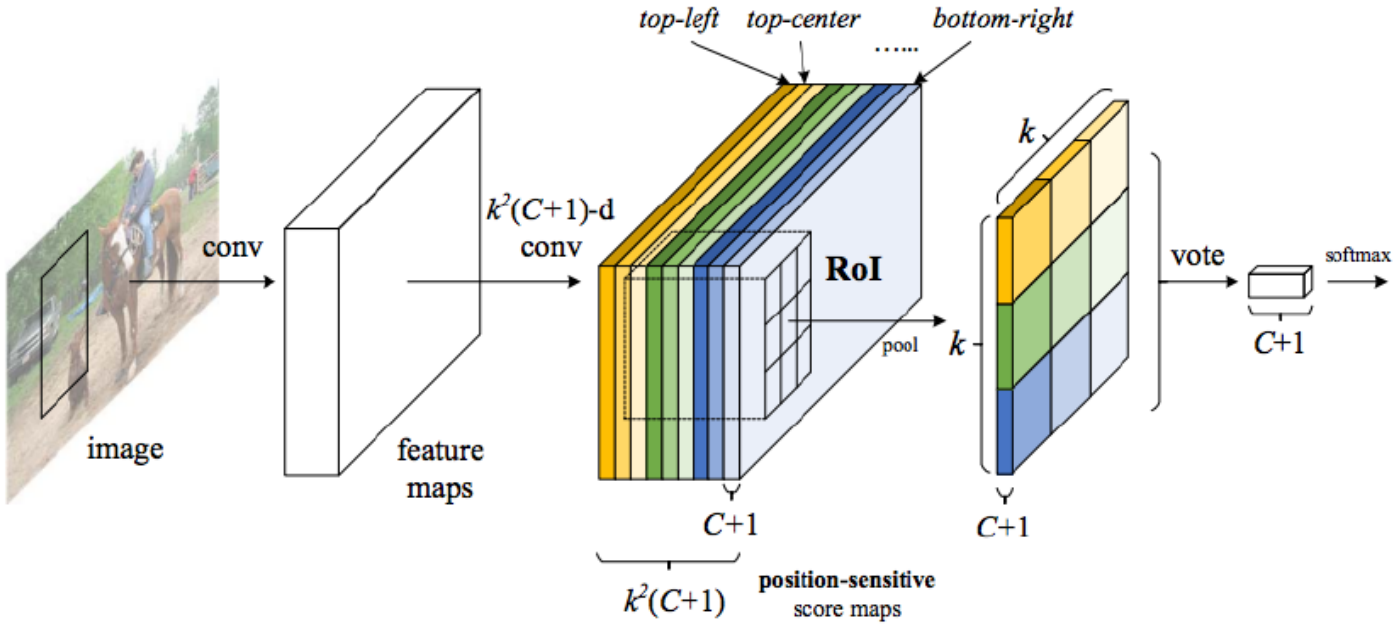
# Object Detection without Proposals

- SSD: multi-scale feature maps
  - Benchmark results

| Method | mAP | FPS | batch size | # Boxes | Input resolution |
|---|---|---|---|---|---|
| Faster R-CNN (VGG16) | 73.2 | 7 | 1 | ~ 6000 | ~ 1000 × 600 |
| Fast YOLO | 52.7 | 155 | 1 | 98 | 448 × 448 |
| YOLO (VGG16) | 66.4 | 21 | 1 | 98 | 448 × 448 |
| SSD300 | 74.3 | 46 | 1 | 8732 | 300 × 300 |
| SSD512 | 76.8 | 19 | 1 | 24564 | 512 × 512 |
| SSD300 | 74.3 | 59 | 8 | 8732 | 300 × 300 |
| SSD512 | 76.8 | 22 | 8 | 24564 | 512 × 512 |

# Objection Detection with Parts

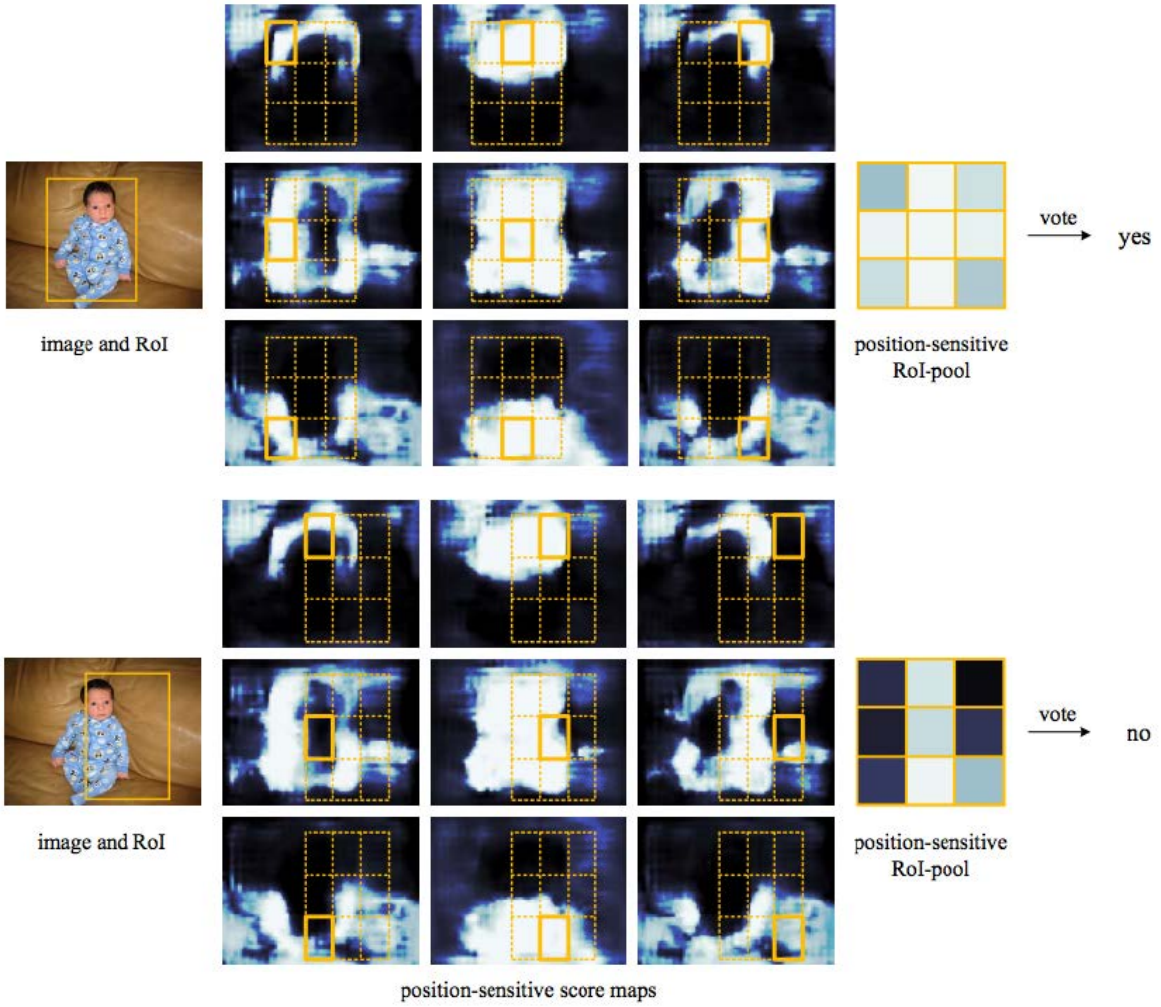
- R-FCN: Position-sensitive score maps




  - Channels take responsibility for relative spatial locations

# R-FCN

- Position-sensitive score maps



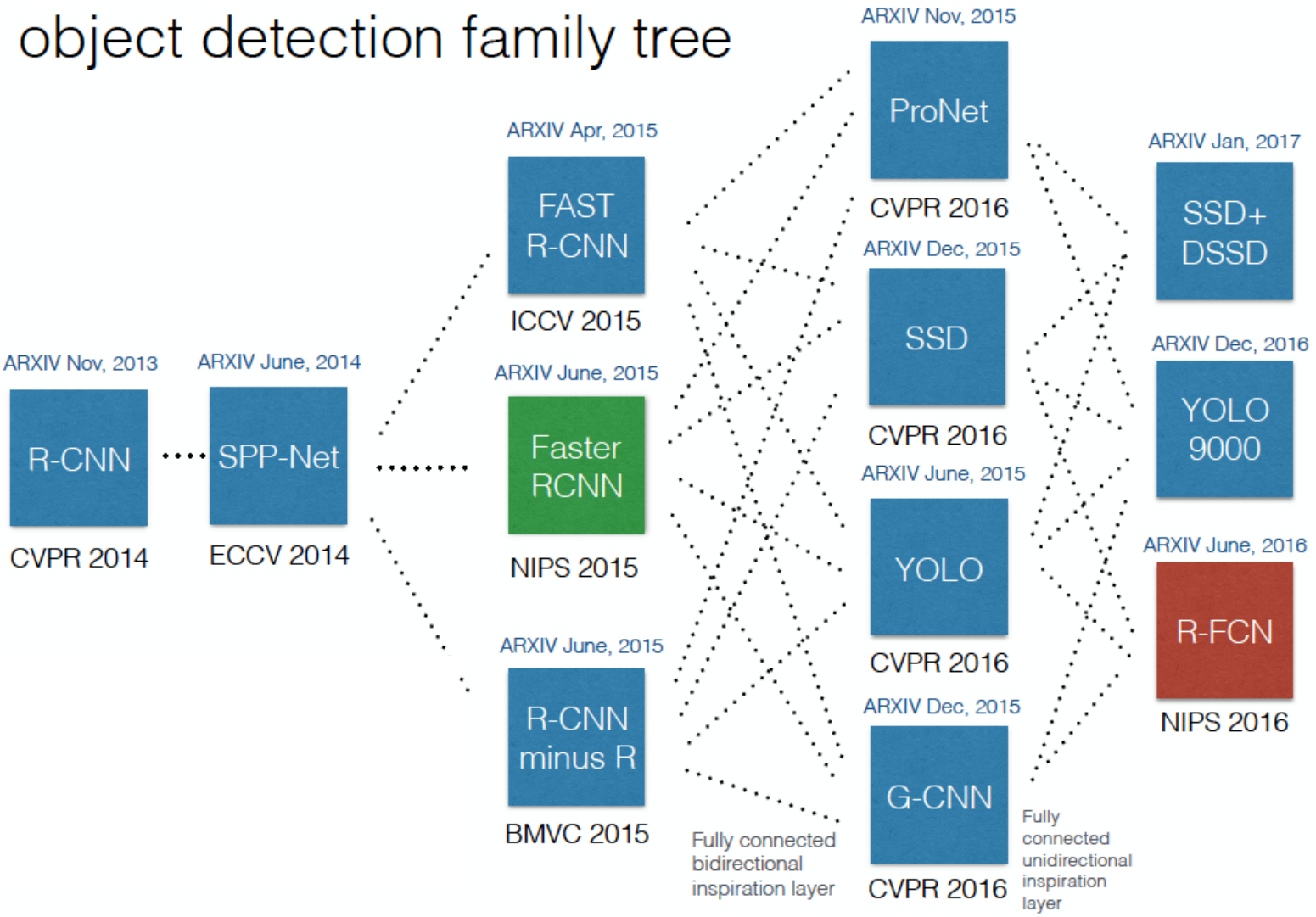
# R-FCN

- Standard benchmarks

Table 5: Comparisons on PASCAL VOC 2012 *test* set using **ResNet-101**. "07++12" [6] denotes the union set of 07 *trainval+test* and 12 *trainval*. †: `http://host.robots.ox.ac.uk:8080/anonymous/44L5HI.html` ‡: `http://host.robots.ox.ac.uk:8080/anonymous/MVCM2L.html`

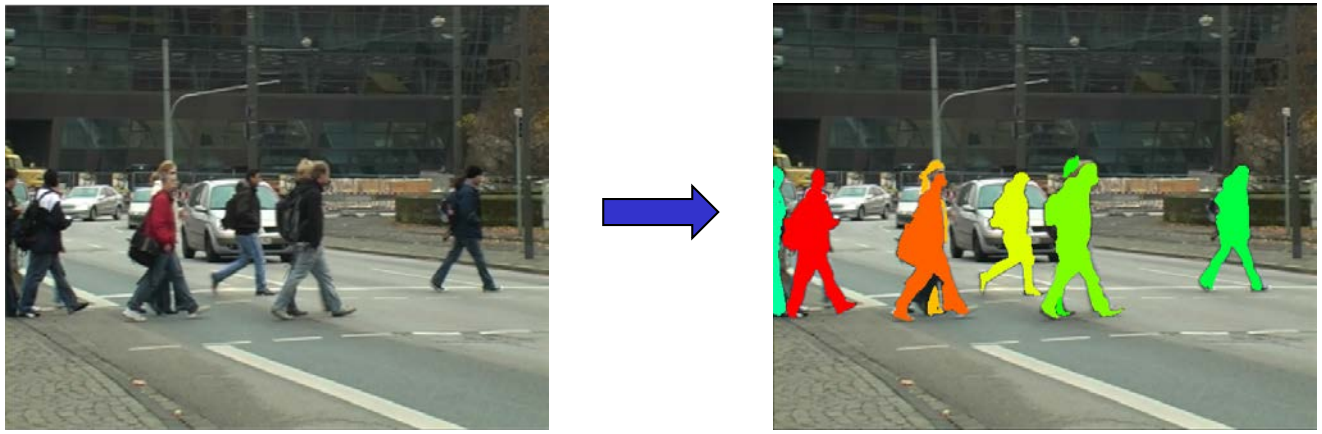
| | training data | mAP (%) | test time (sec/img) |
|---|---|---|---|
| Faster R-CNN [9] | 07++12 | 73.8 | 0.42 |
| Faster R-CNN +++ [9] | 07++12+COCO | **83.8** | 3.36 |
| **R-FCN** multi-sc train | 07++12 | 77.6† | 0.17 |
| **R-FCN** multi-sc train | 07++12+COCO | **82.0**‡ | 0.17 |

# Object Detection Methods




Slides from Sam Albanie

# Outline



- Object Detection
- Semantic Instance Segmentation

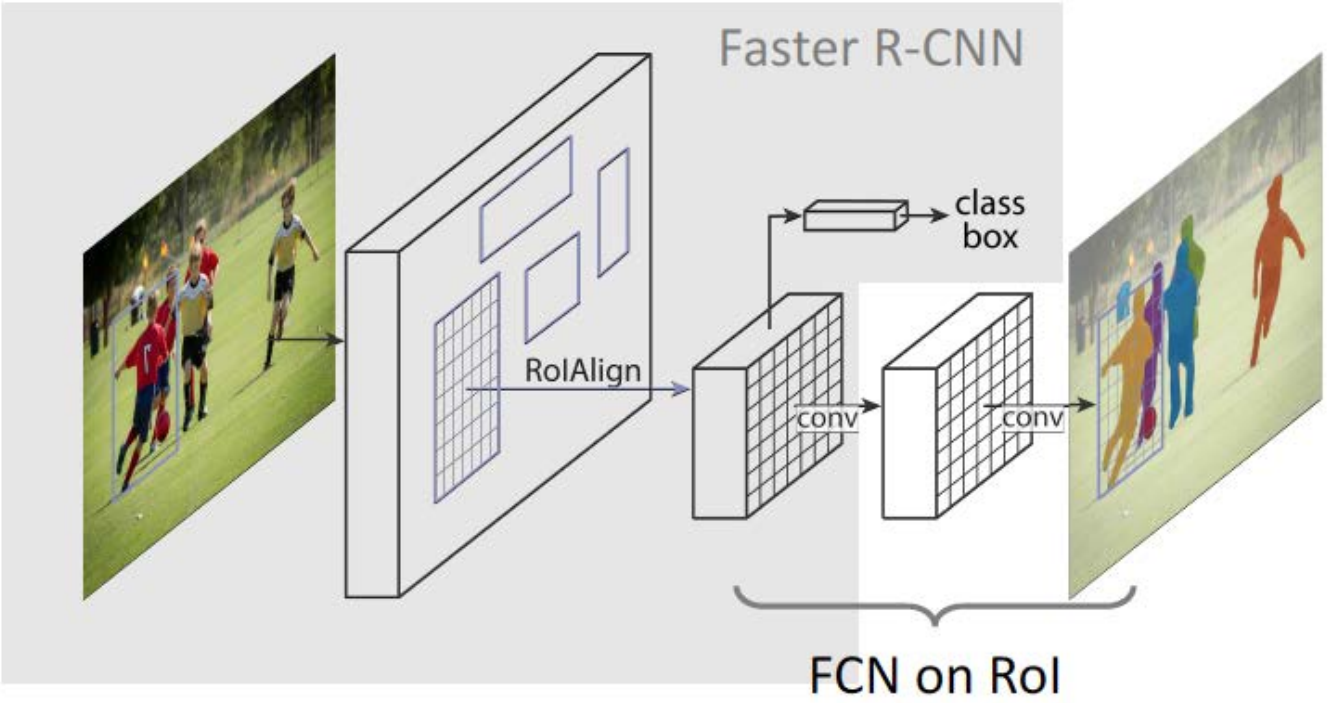
*Acknowledgement: Feifei Li et al's cs231n notes*

# Object Instance Segmentation

- Problem setup
  - Input: image, object class(es)
  - Output: object instance masks + object scores

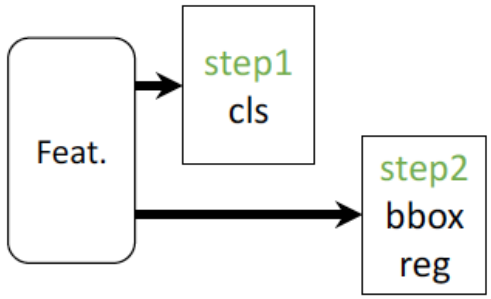
# Mask R-CNN

- Problem formulation
  - Mask R-CNN = **Faster R-CNN** with **FCN** on RoIs

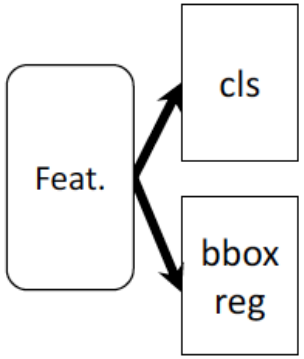


# Mask R-CNN

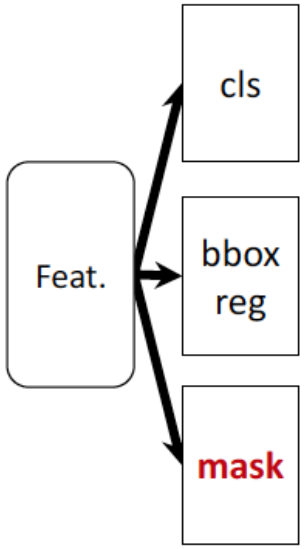
- Parallel heads
  - Easy, fast to implement and train




(slow) R-CNN




Fast/er R-CNN

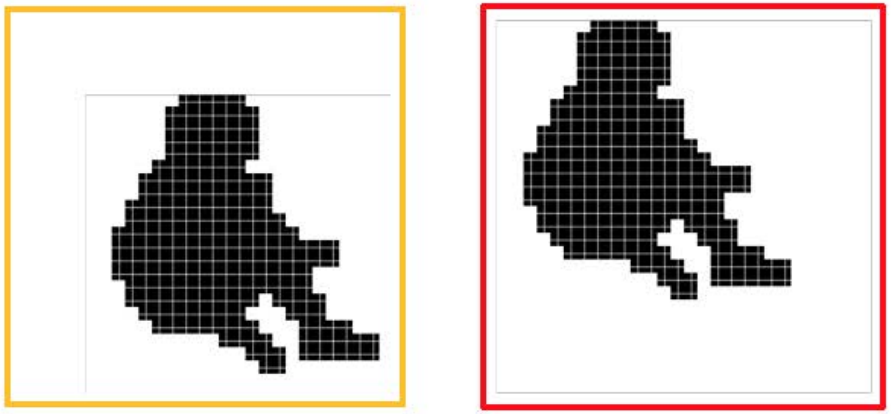



Mask R-CNN

# Mask R-CNN

- Fully-Convolution on RoI

target masks on RoIs

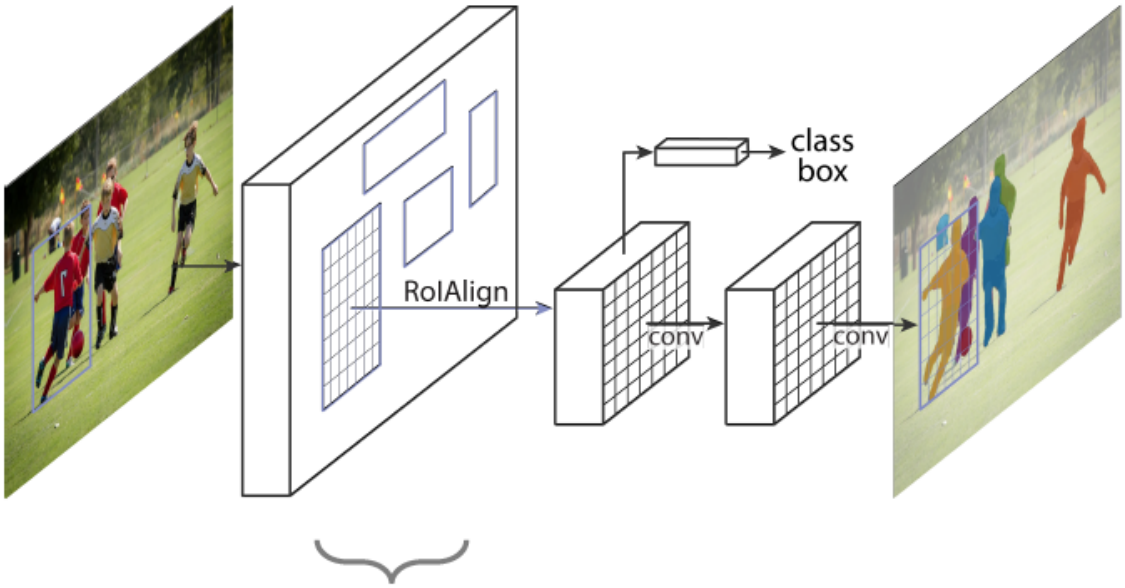


Translation of object in RoI => Same translation of mask in RoI

- Equivariant to small translation of RoIs
- More robust to RoI's localization imperfection

# Mask R-CNN

- Fully-Convolution on RoI

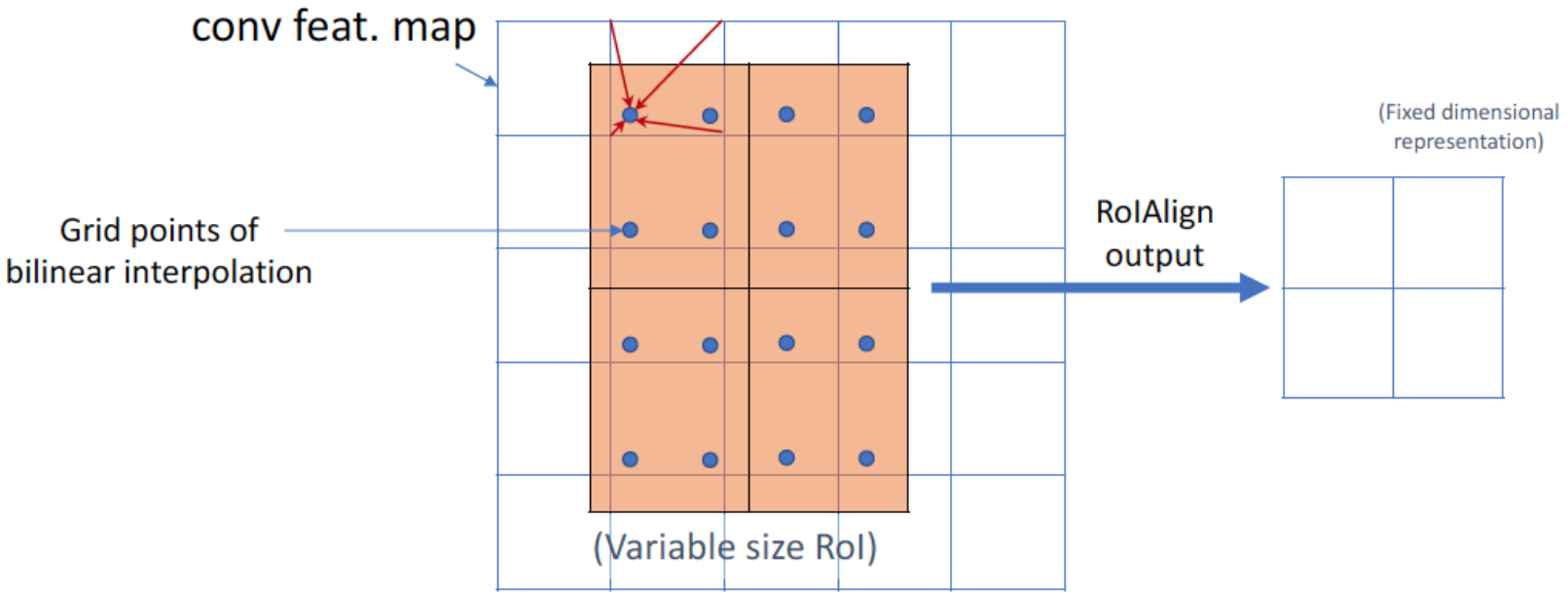



3. RoIAlign:

3a. maintain translation-equivariance before/after RoI

# Mask R-CNN

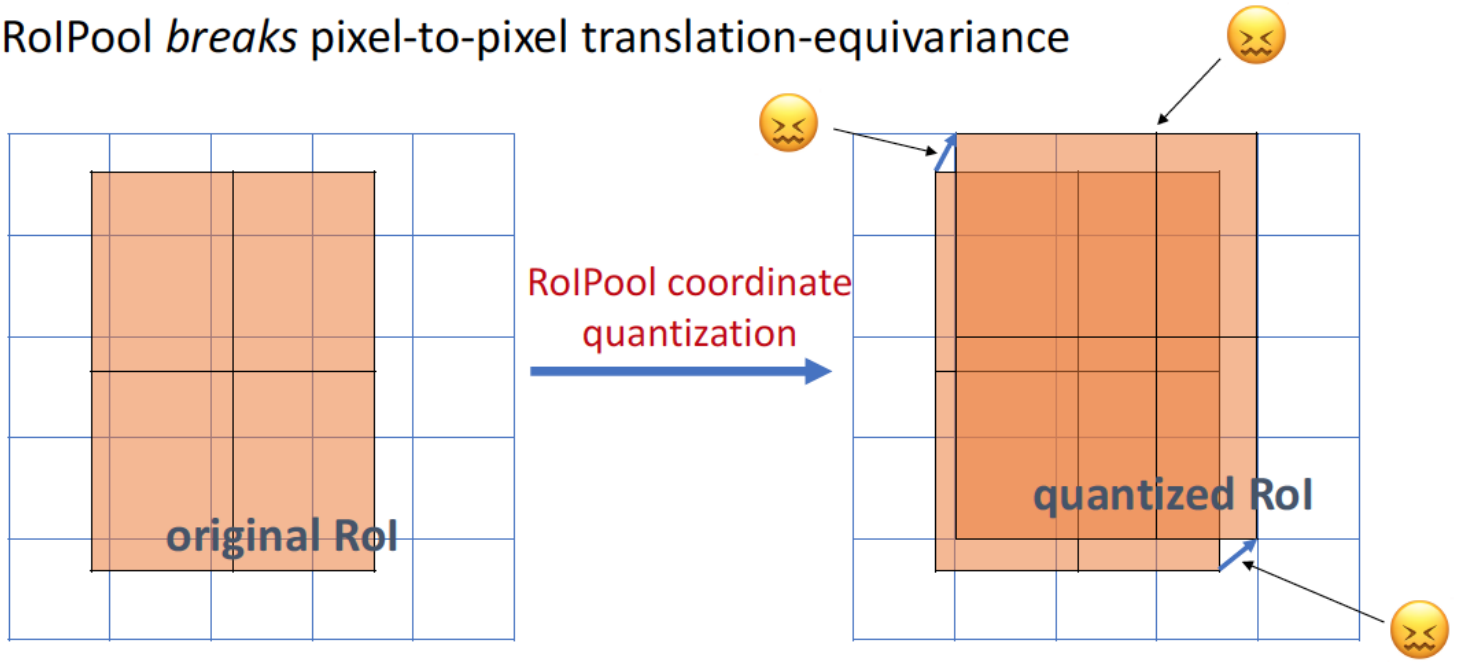
- Fully-Convolution on RoI
  - RoIAlign



# Mask R-CNN

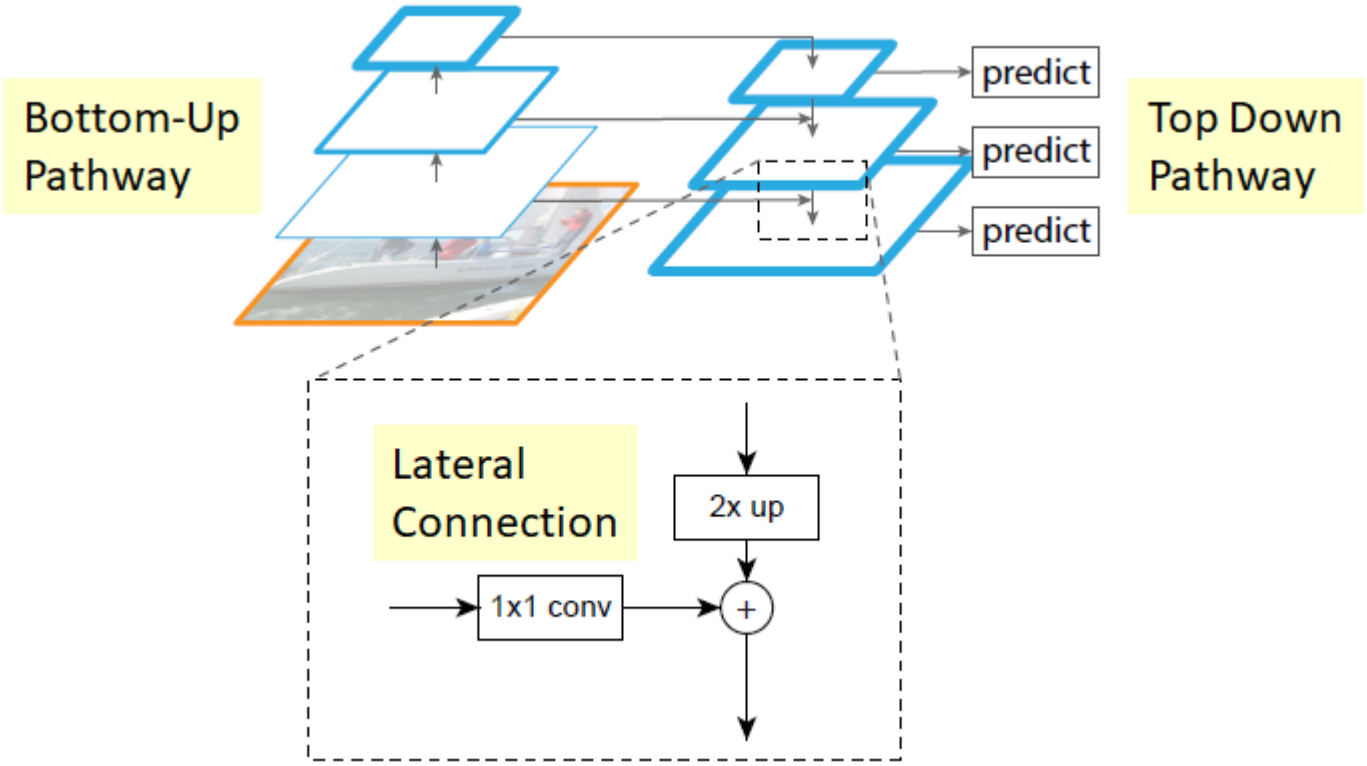
- Fully-Convolution on RoI
  - RoIAlign vs RoIPool

• RoIPool *breaks* pixel-to-pixel translation-equivariance



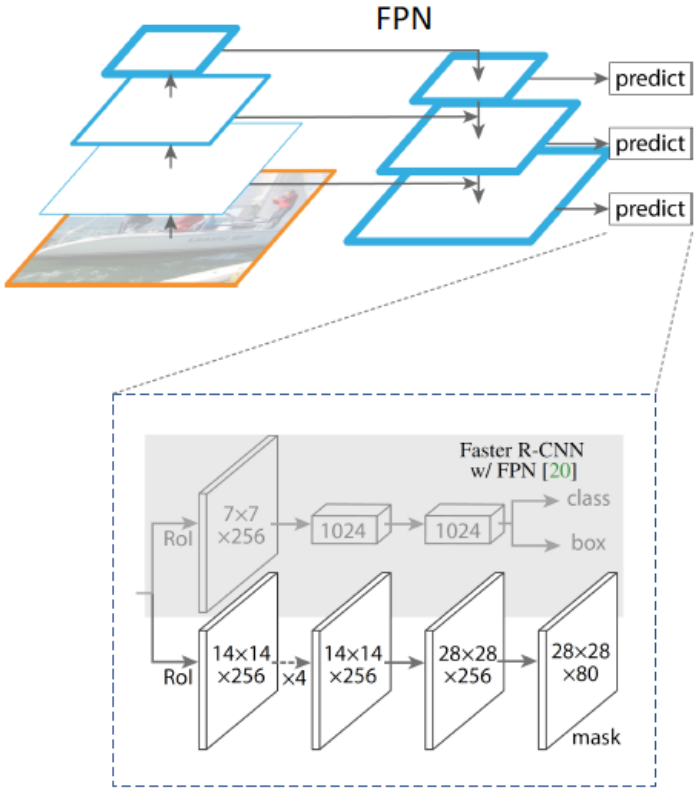
# Mask R-CNN

- Multiscale representation
  - Feature Pyramid Network (FPN)



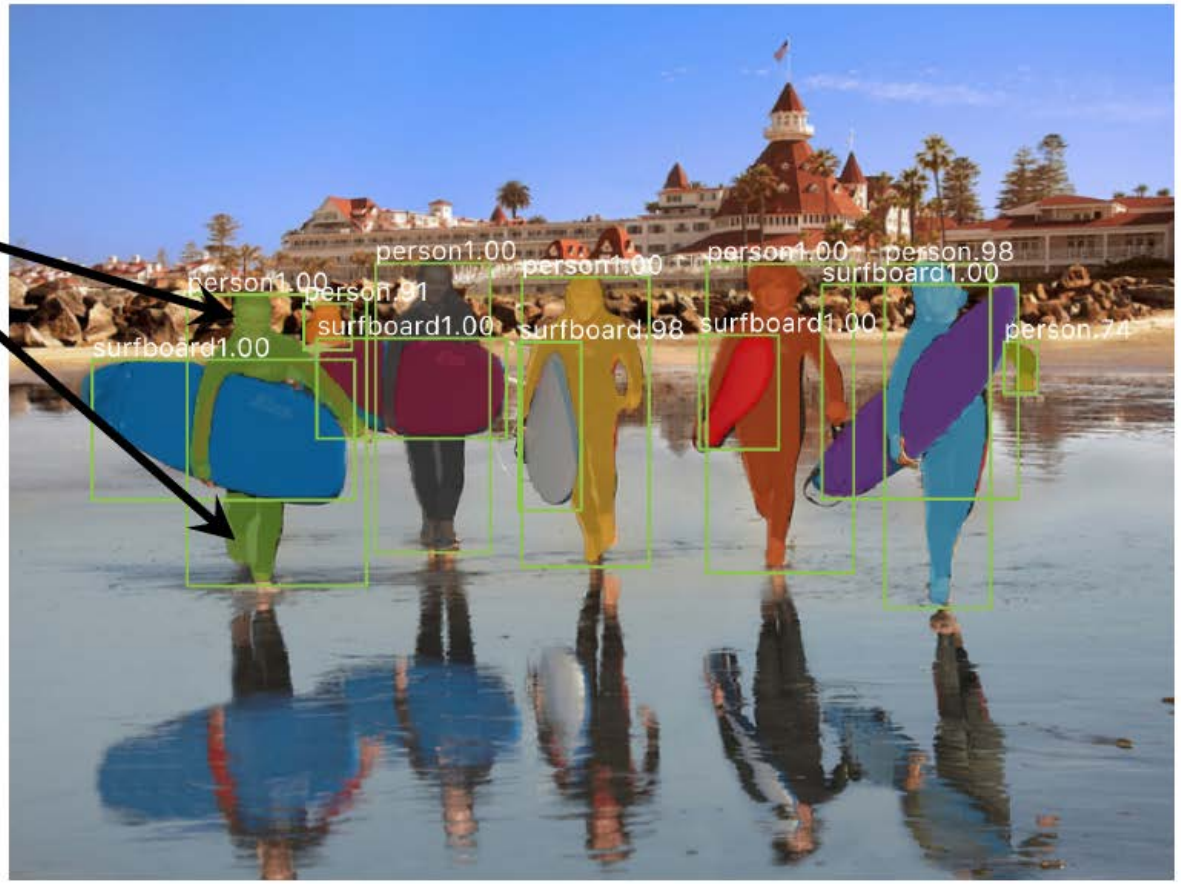
# Mask R-CNN

- Multiscale representation
  - Feature Pyramid Network (FPN)



# Mask R-CNN

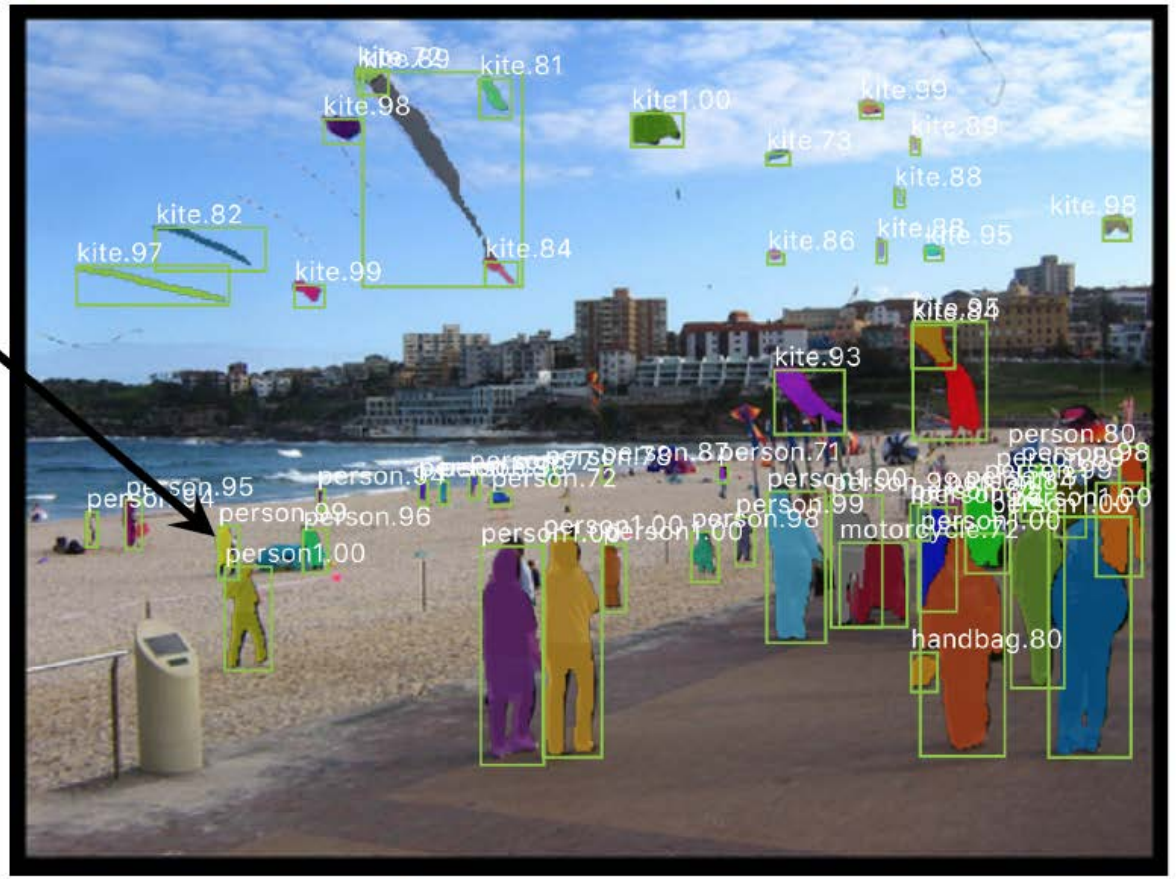
- Results




Mask R-CNN results on COCO

# Mask R-CNN

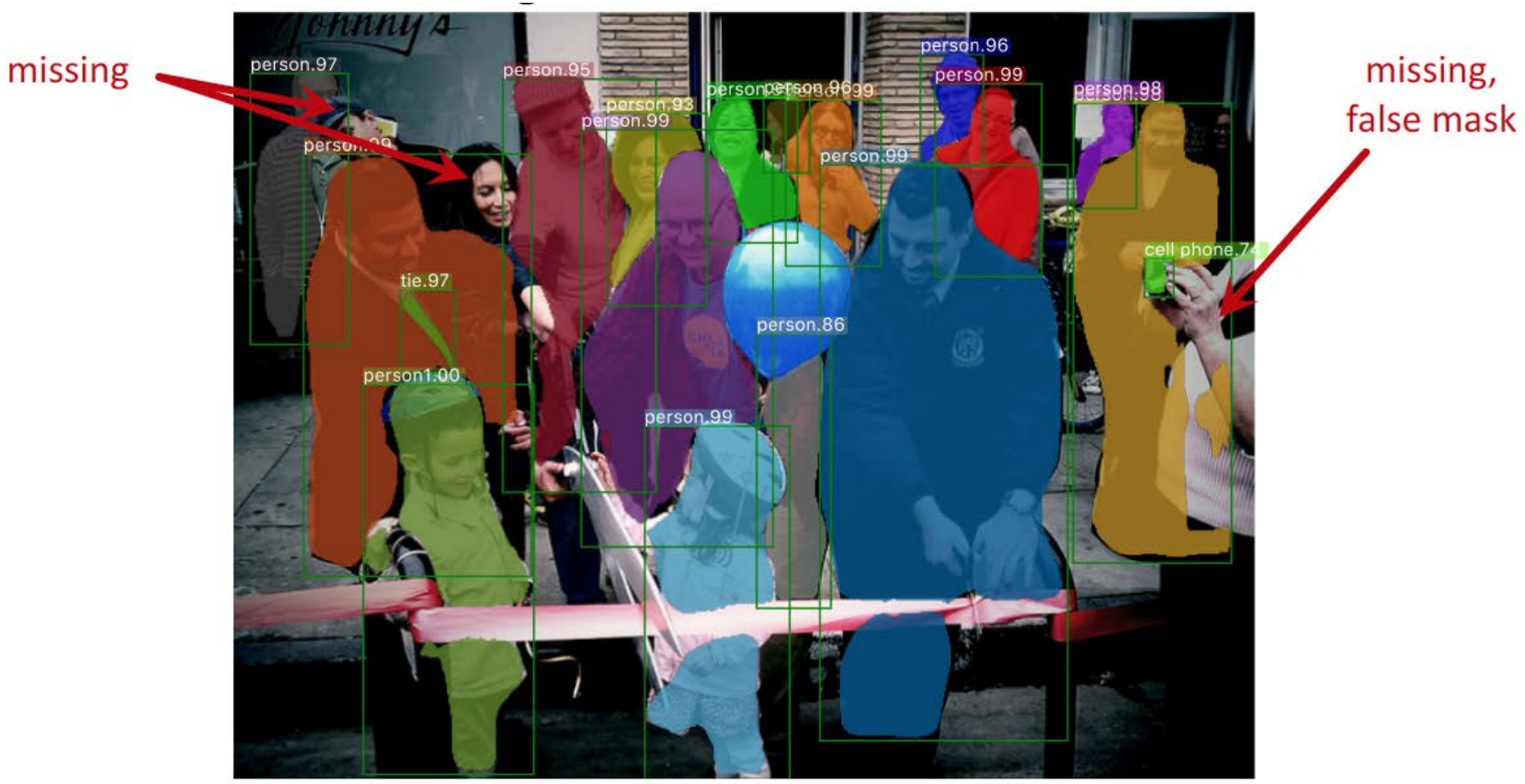
- Results




Mask R-CNN results on COCO

# Mask R-CNN

- Results




Mask R-CNN results on COCO

# Summary

- CNNs in computer vision
  - Localization, Detection
  - Instance segmentation
- Other research topics (not discussed)
  - *3D geometry: stereo, multiview correspondence, reconstruction*
  - *Video: action and activity recognition and detection*
  - *Volumetric/Multimodality: RGB-D images, medical imaging, etc.*
- Next time:
  - Understanding CNNs
  - Limitations of CNNs