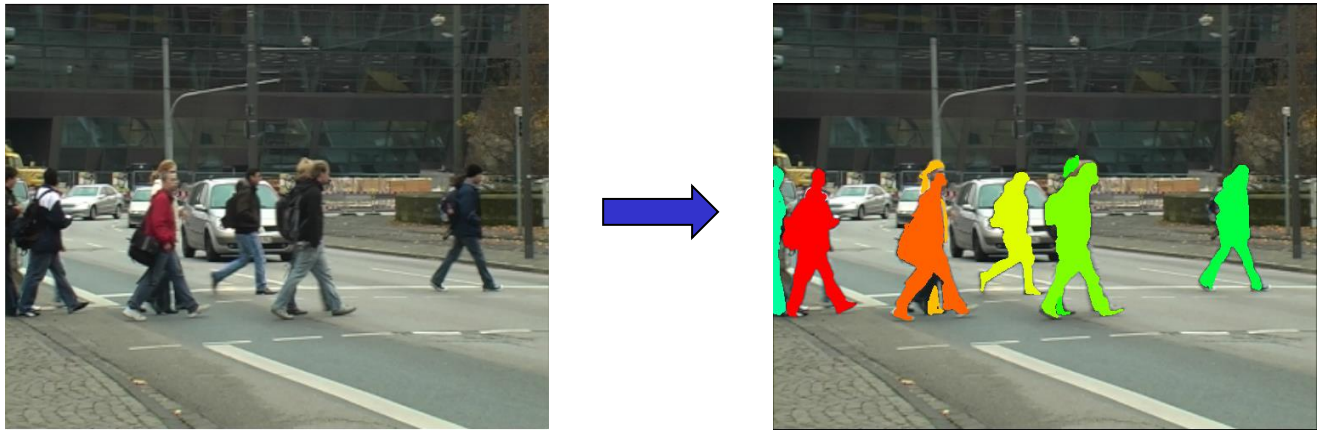
Lecture 9: CNNs in Computer Vision II: Object Localization

Lan Xu

SIST, ShanghaiTech

Fall, 2021

Outline



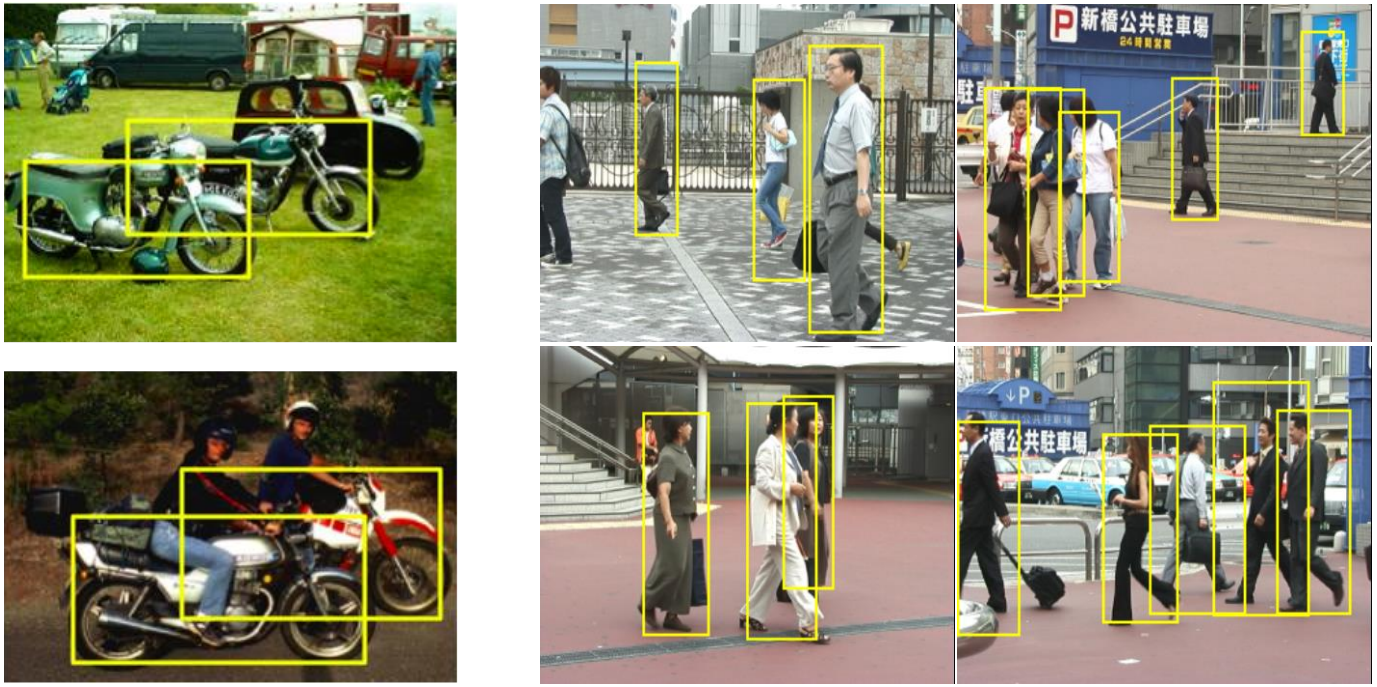
- Object Detection
- Semantic Instance Segmentation

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

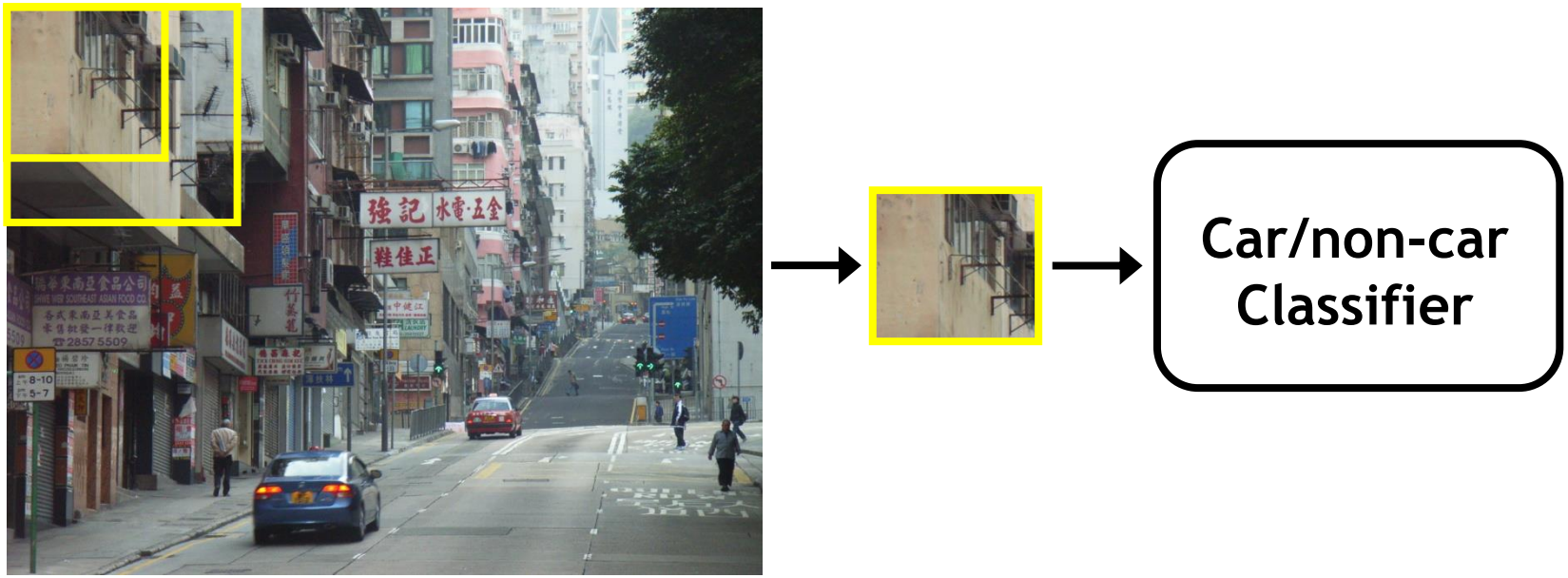
Detection: Why it is so hard?



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

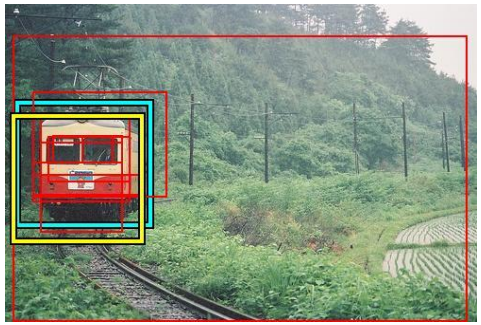
Object Detection

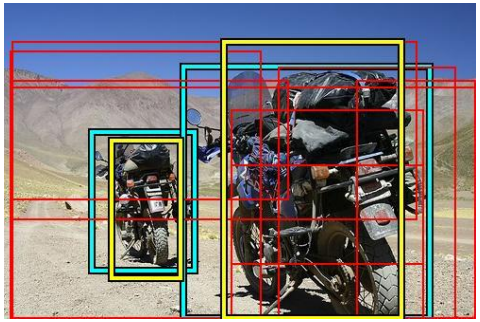
- Problem formulation
  - Object detection as a series of classification problems
  - Step1: Generate object candidate boxes
  - Step2: Scoring the object candidate with a classifier
- Example: sliding window method



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.

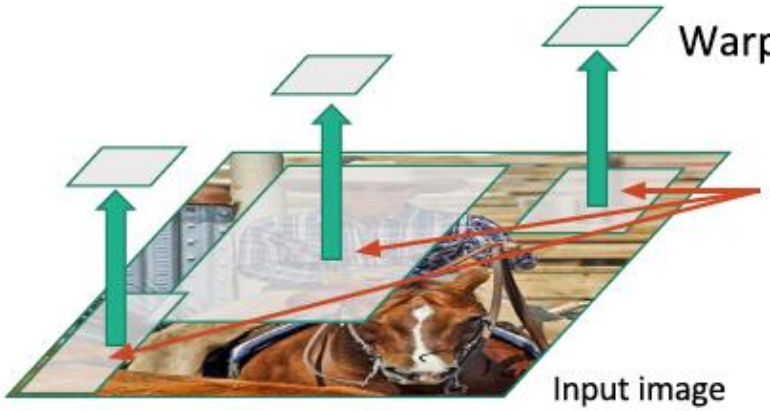
R-CNN

Input image

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

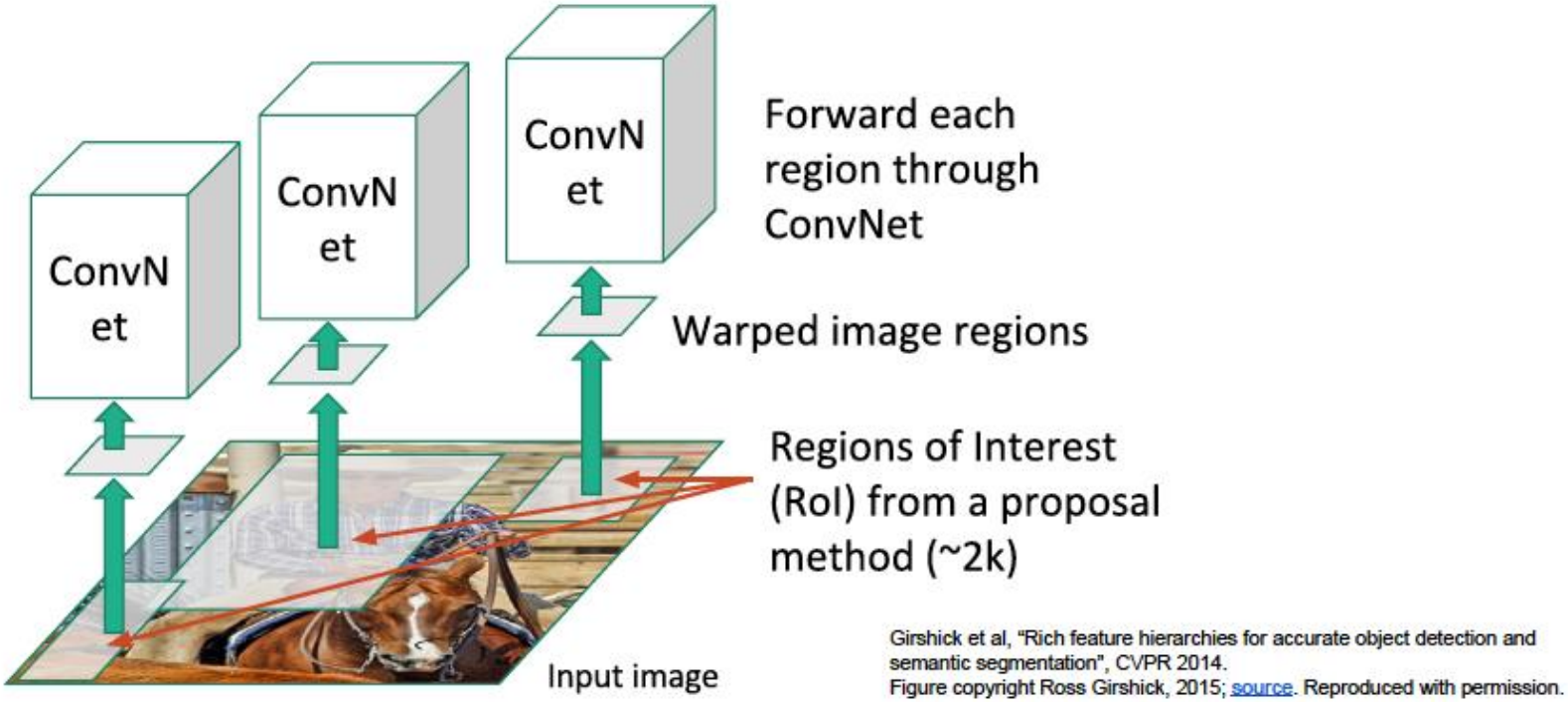
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

R-CNN



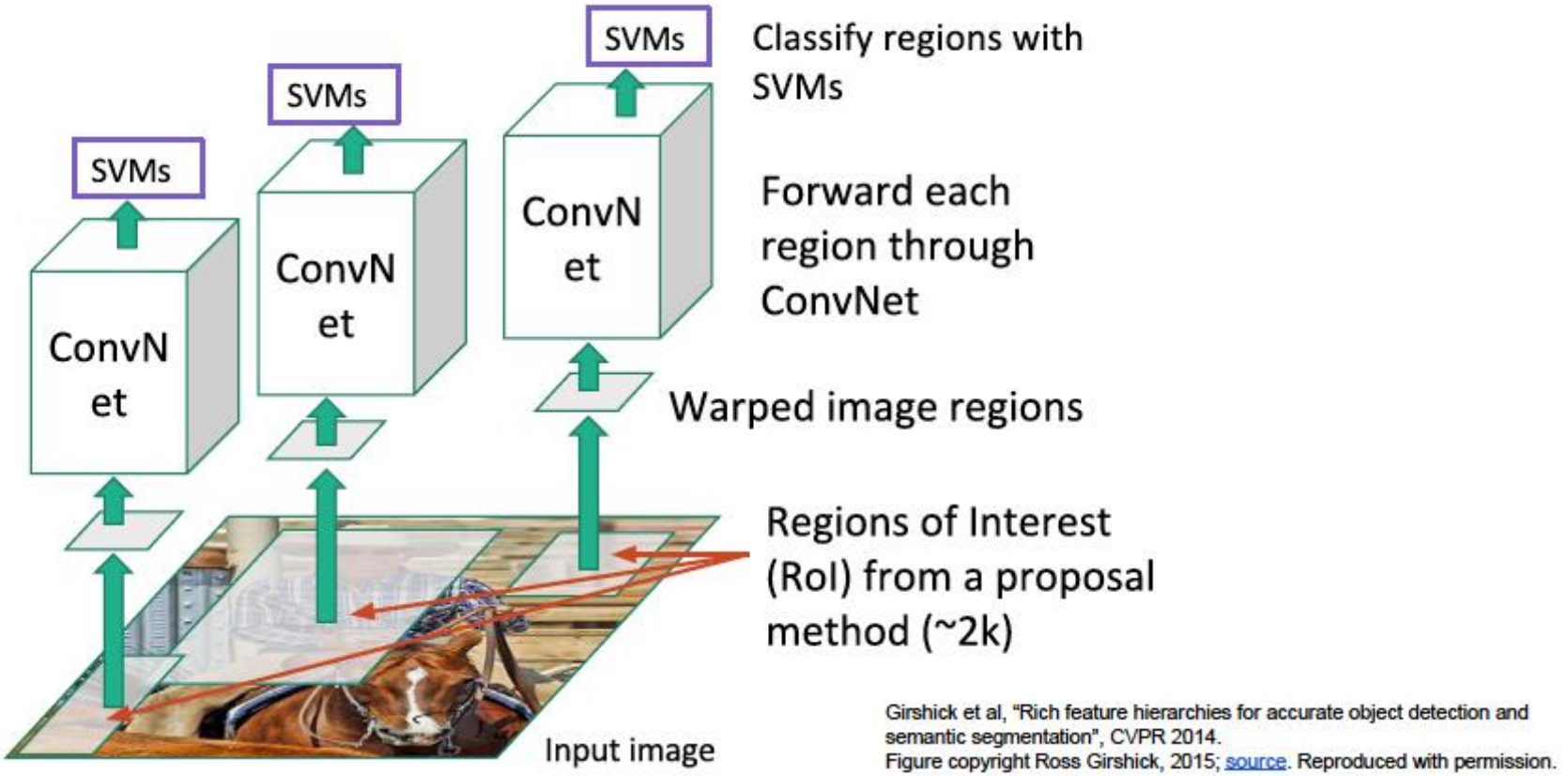
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

R-CNN



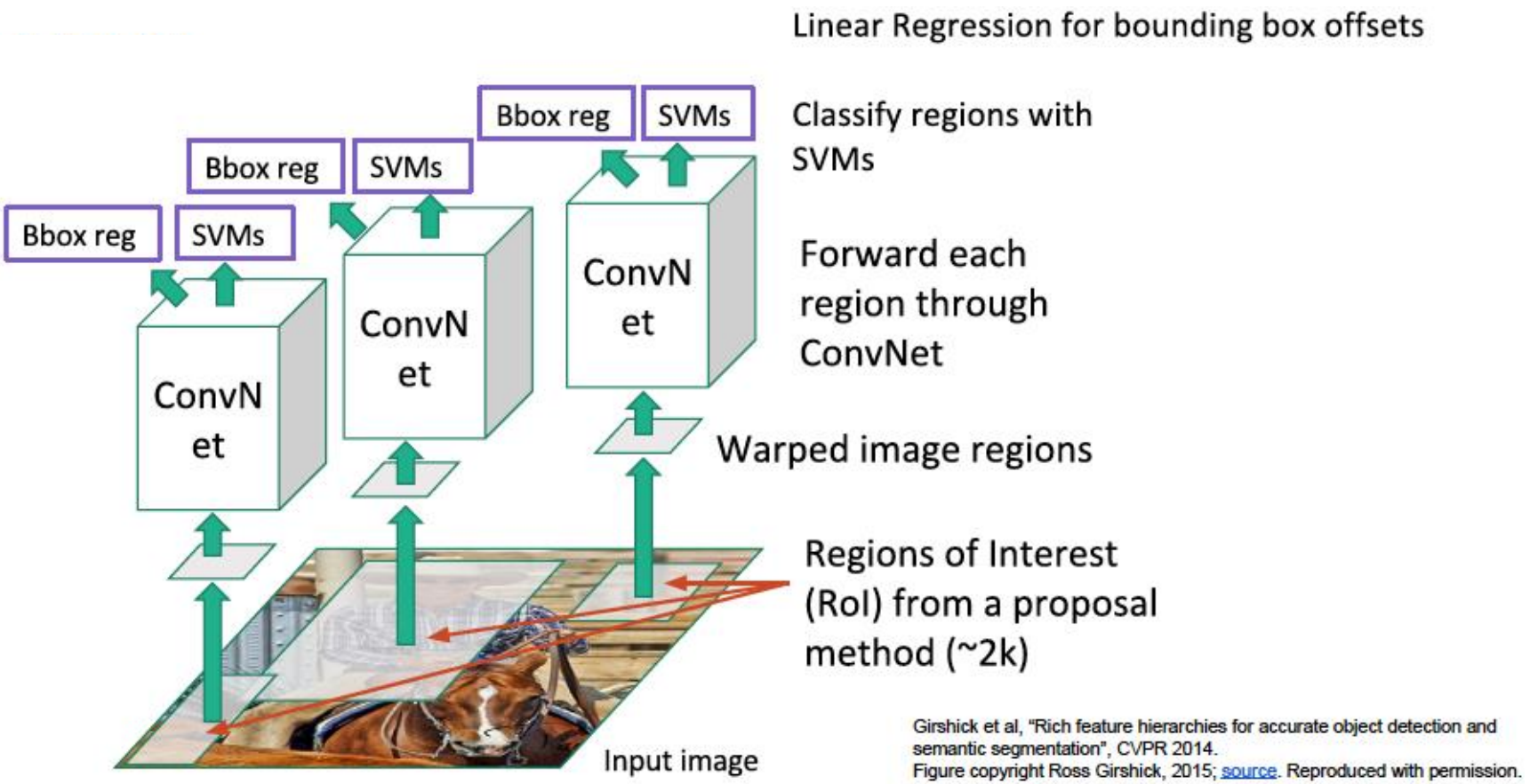
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

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.

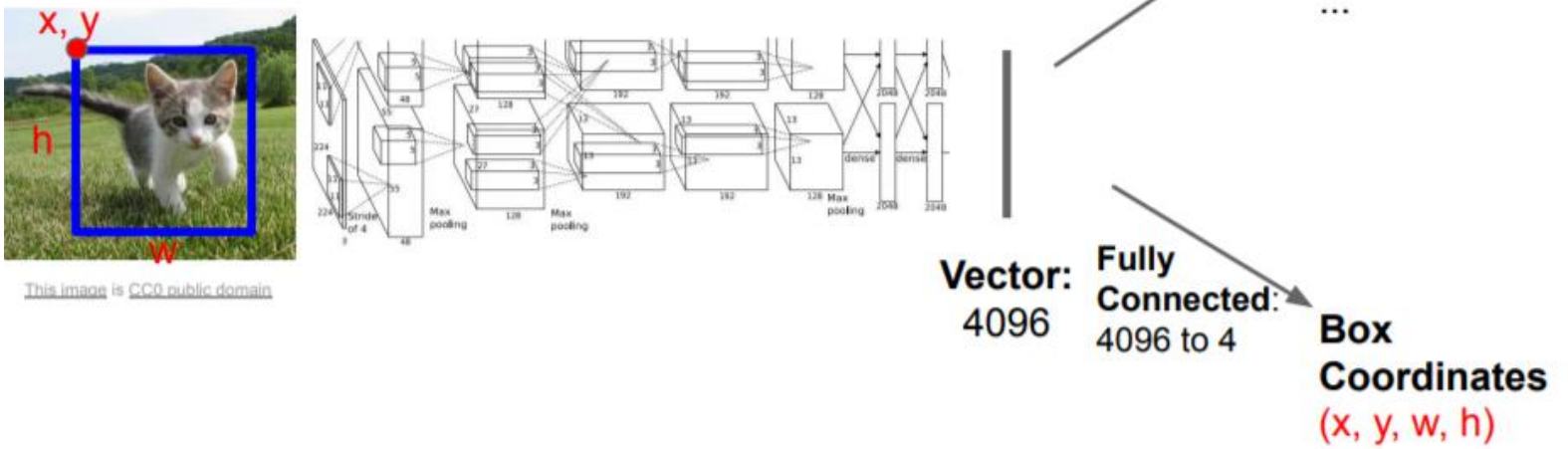
R-CNN



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

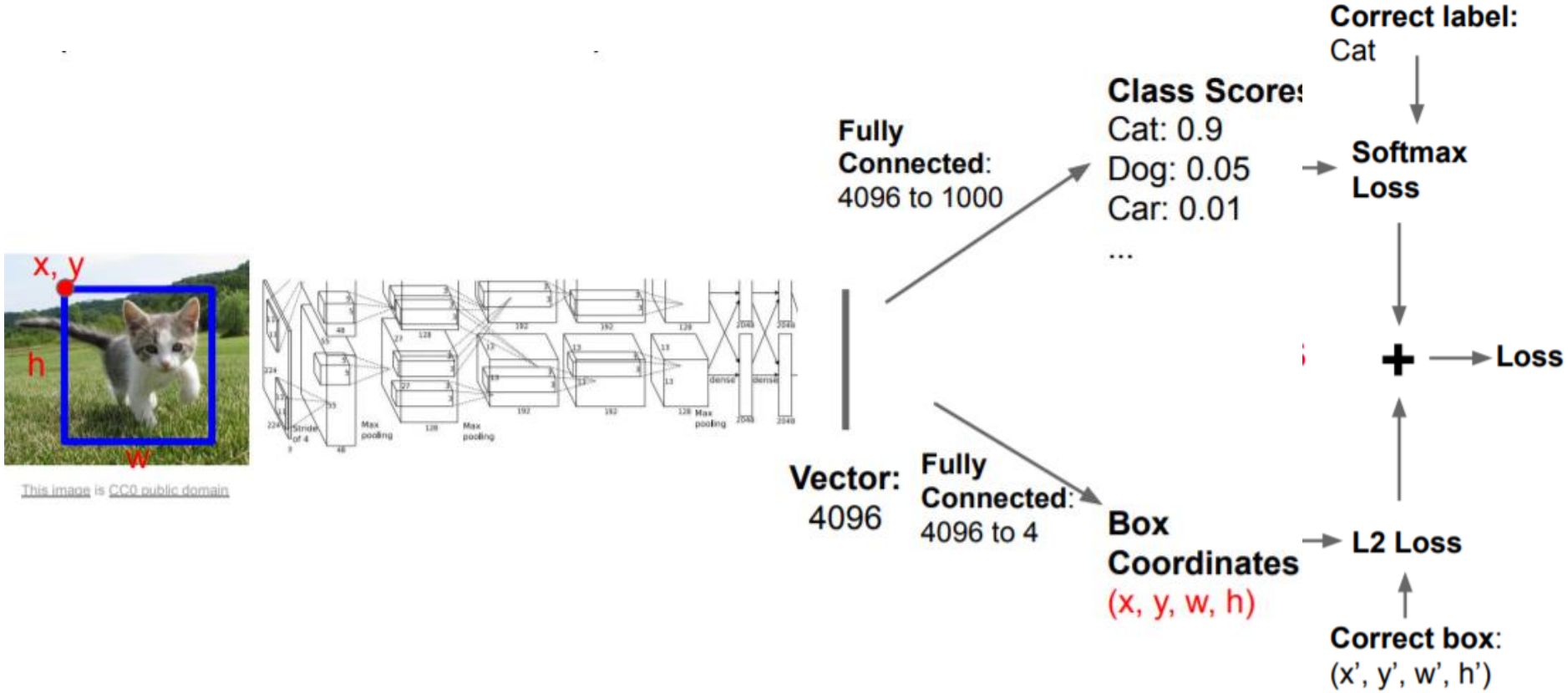
R-CNN

- Each proposal



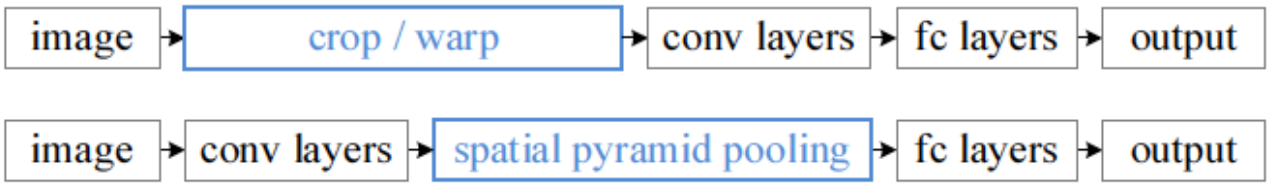
R-CNN

- Multi-task loss



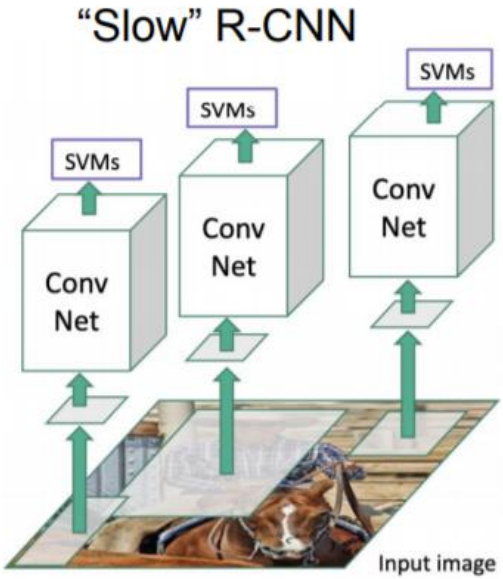
R-CNN

- 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]



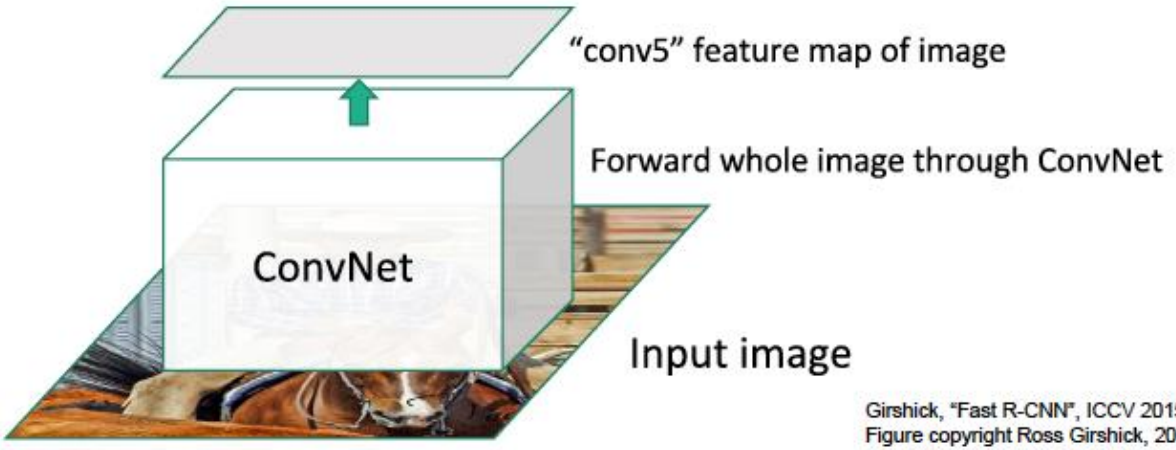
Fast R-CNN

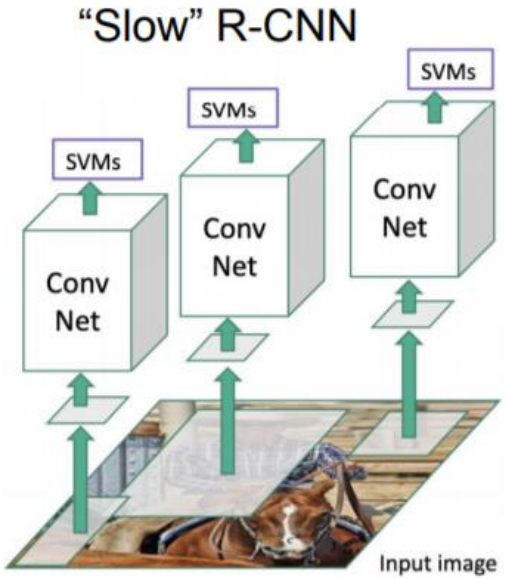
Input image



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

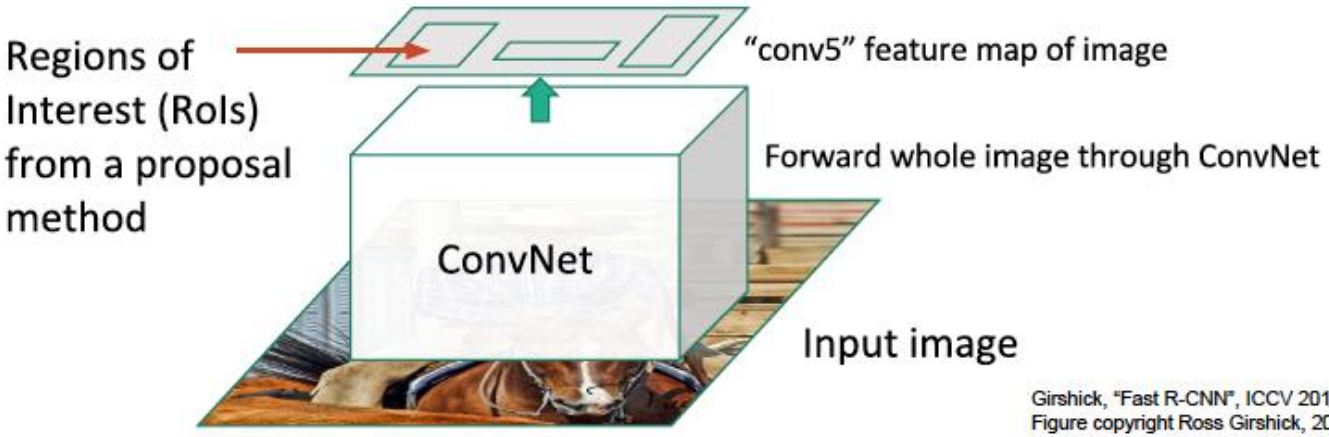
Fast R-CNN

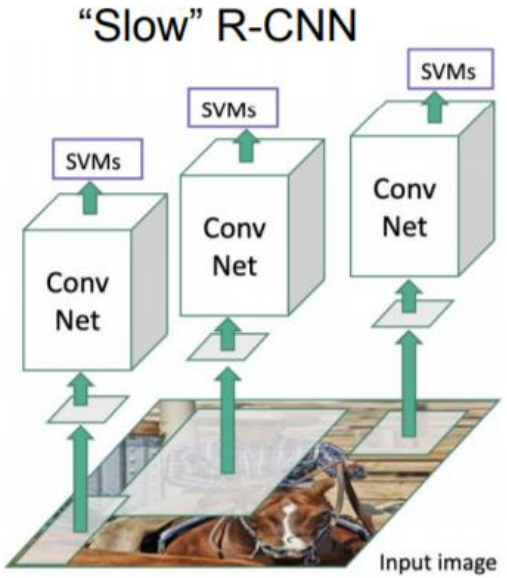




Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

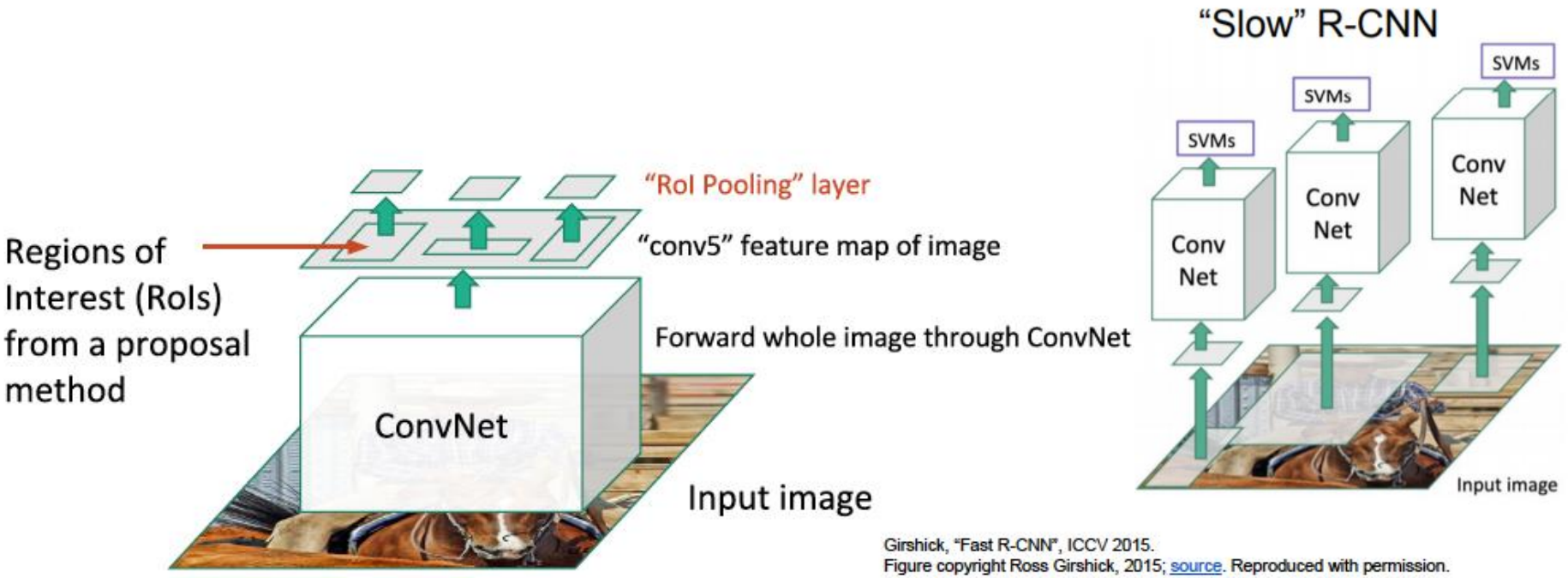
Fast R-CNN





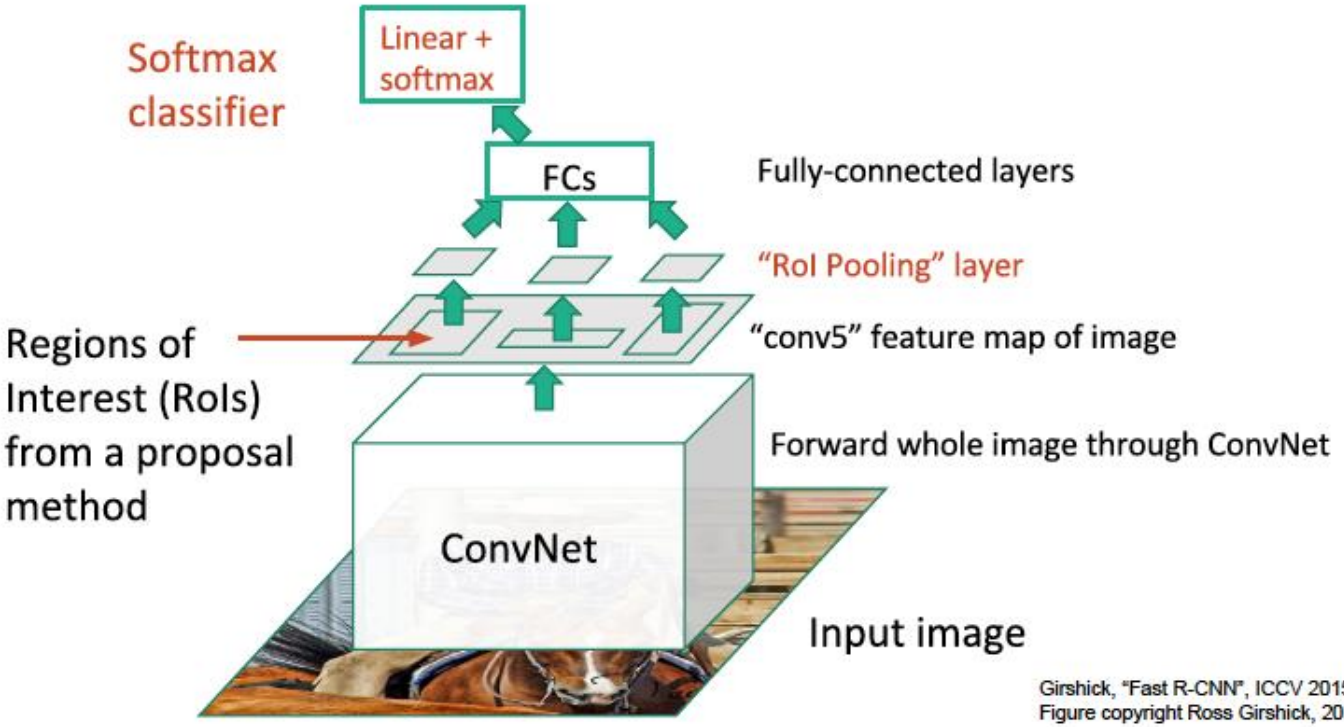
Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

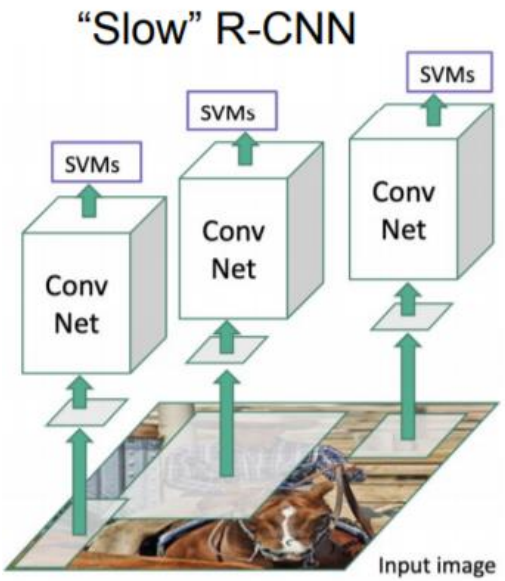
Fast R-CNN



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

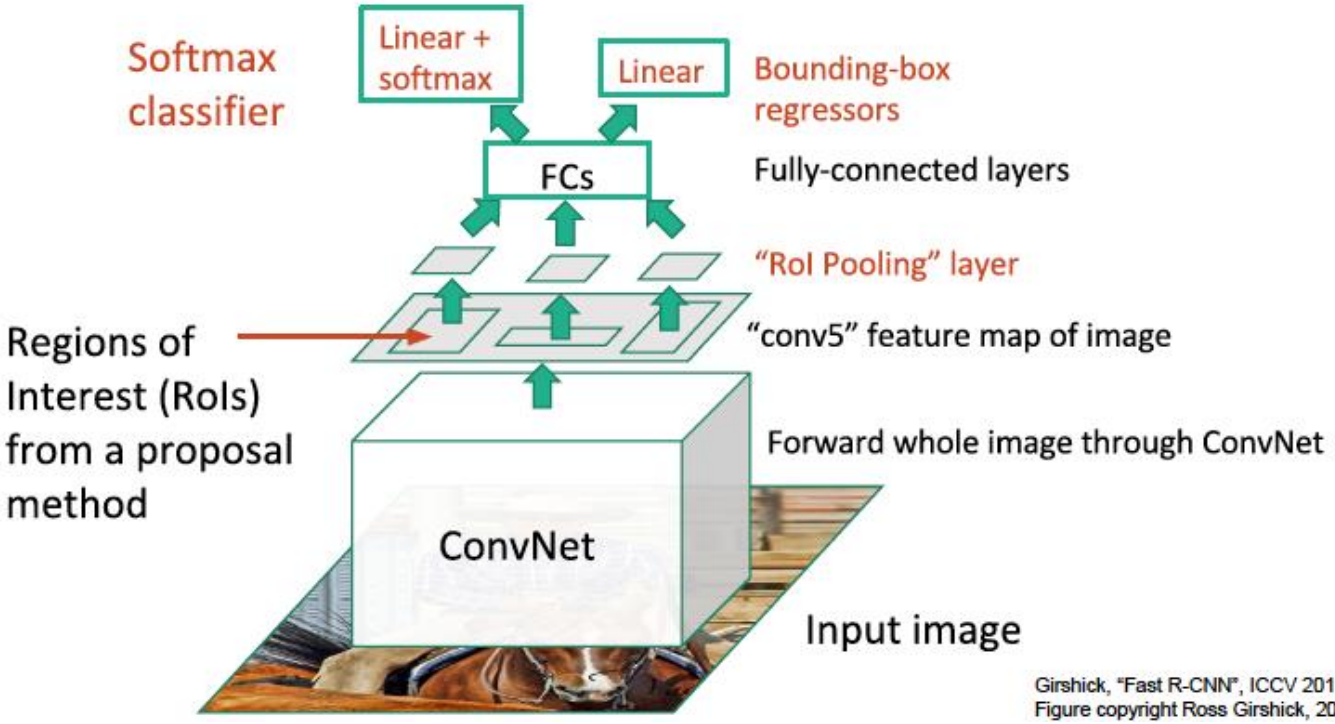
Fast R-CNN

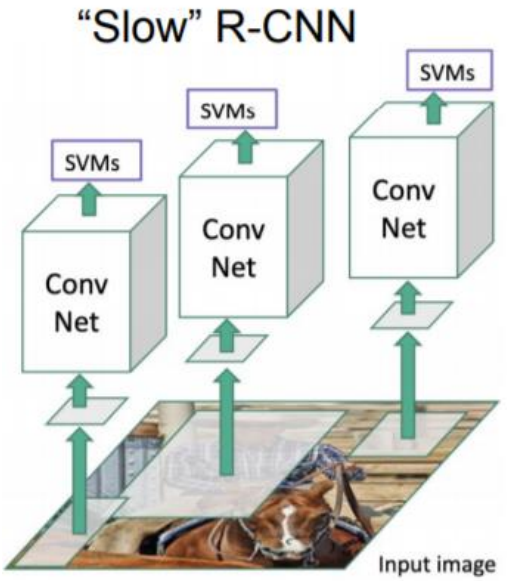




Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Fast R-CNN

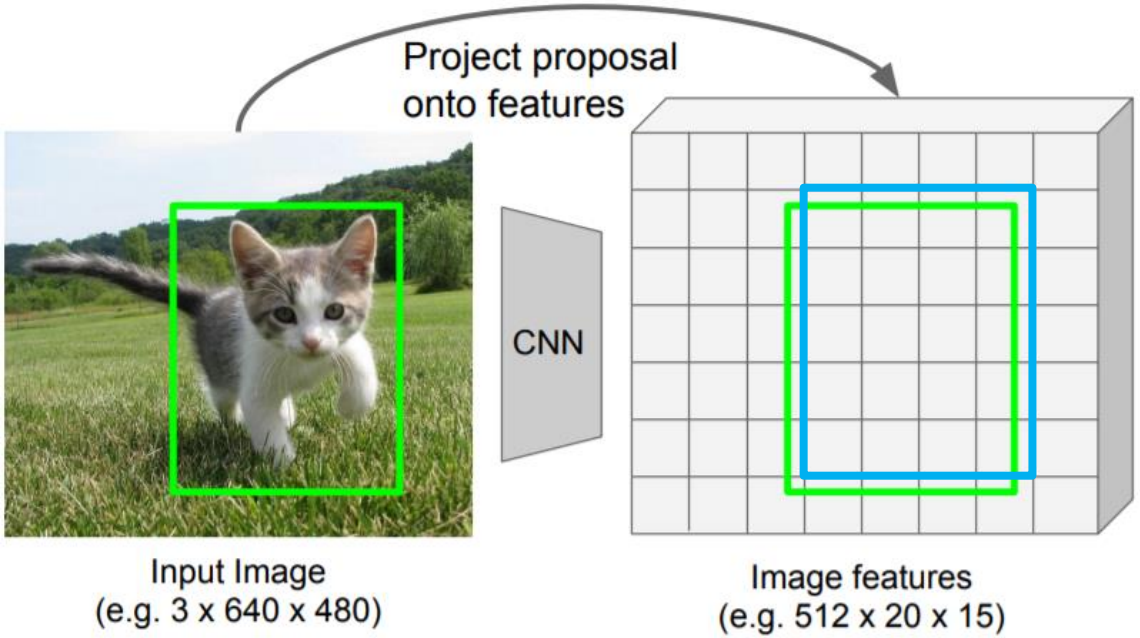




Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Fast R-CNN

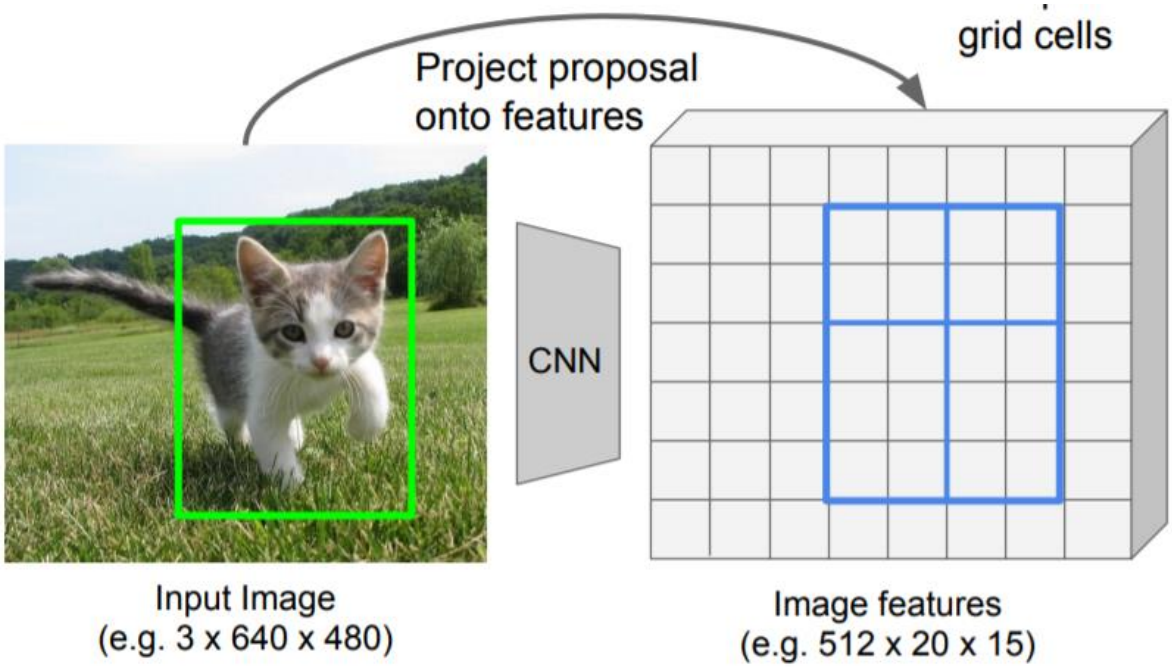
- RoI Pooling



Snap to
grid cells

Fast R-CNN

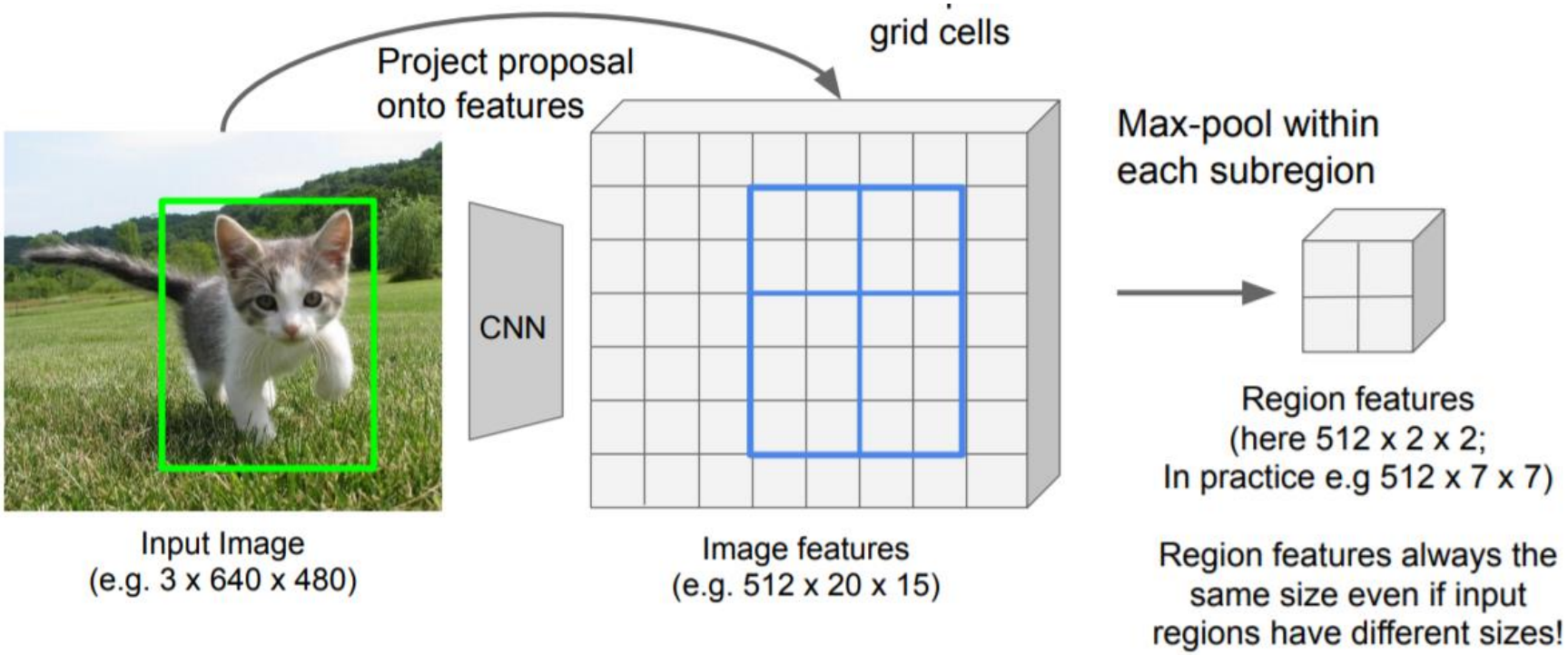
- RoI Pooling



Divide into 2x2 grid of (roughly) equal subregions

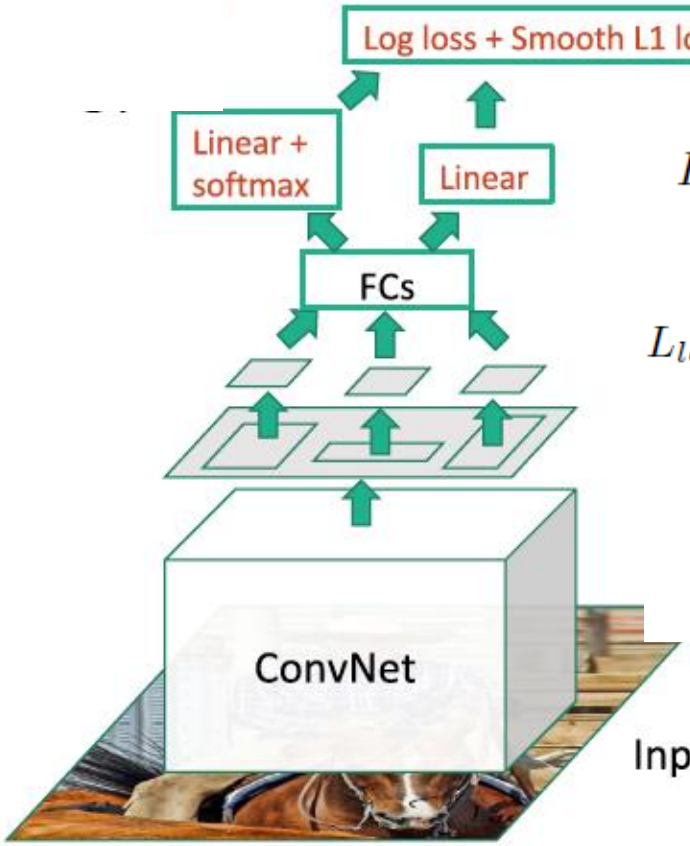
Fast R-CNN

- RoI Pooling



Fast R-CNN

- Deep network training



Multi-task loss

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

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

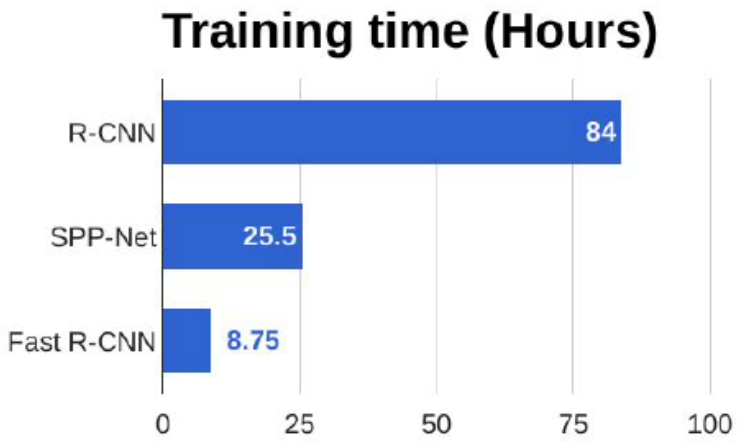
$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h$$

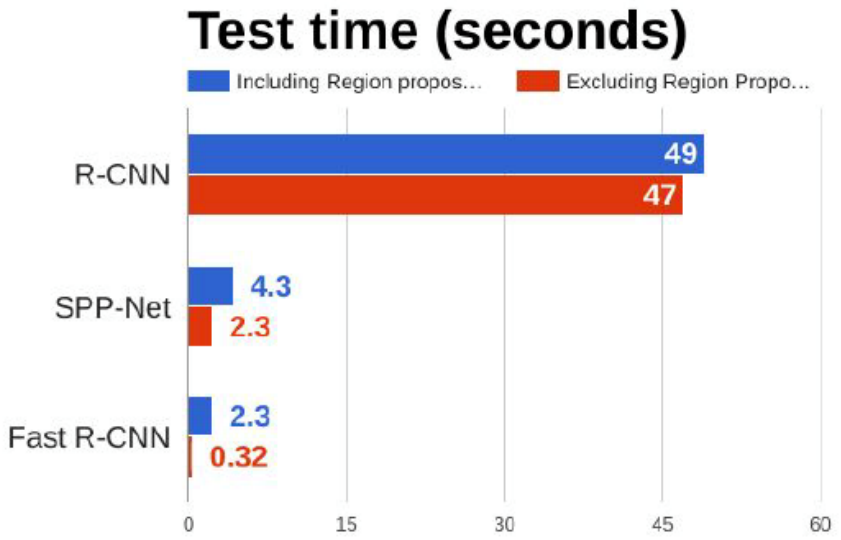
$$\hat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right) \qquad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right)$$

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

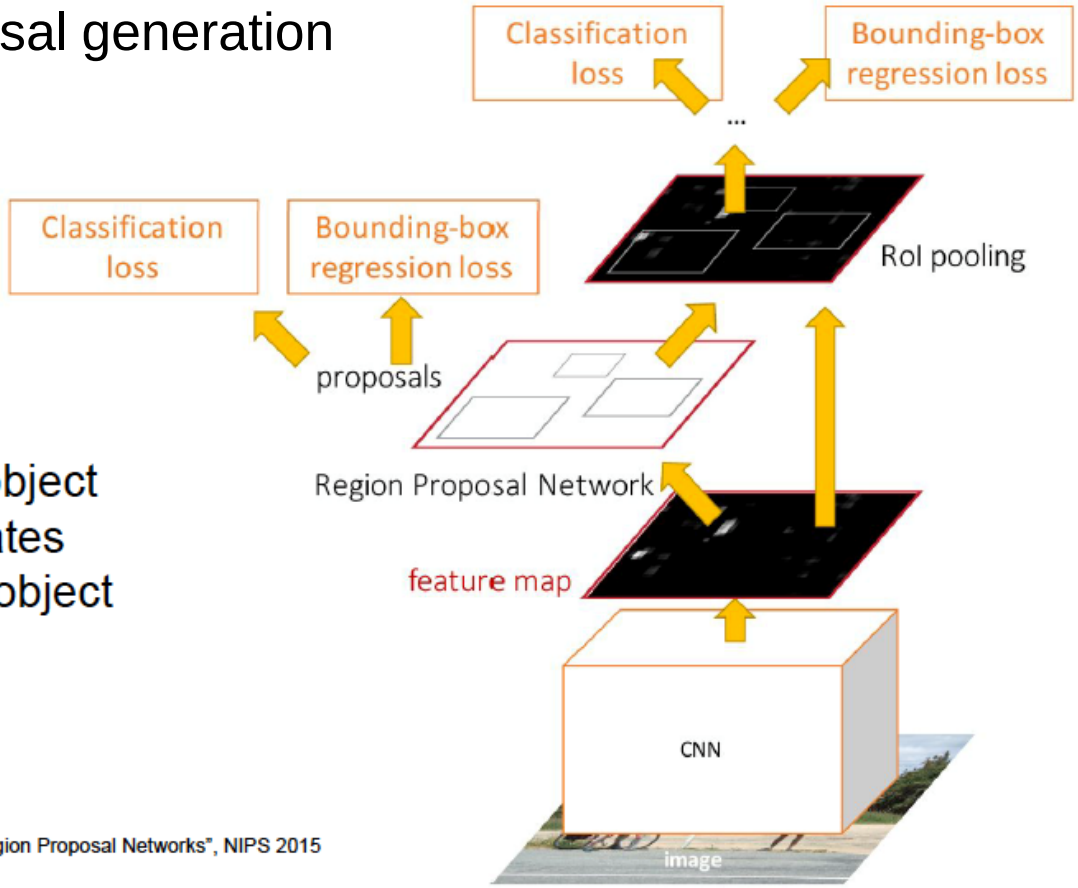
Faster R-CNN

- Region proposal network
  - Make CNN do proposal generation

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

Jointly train with 4 losses:

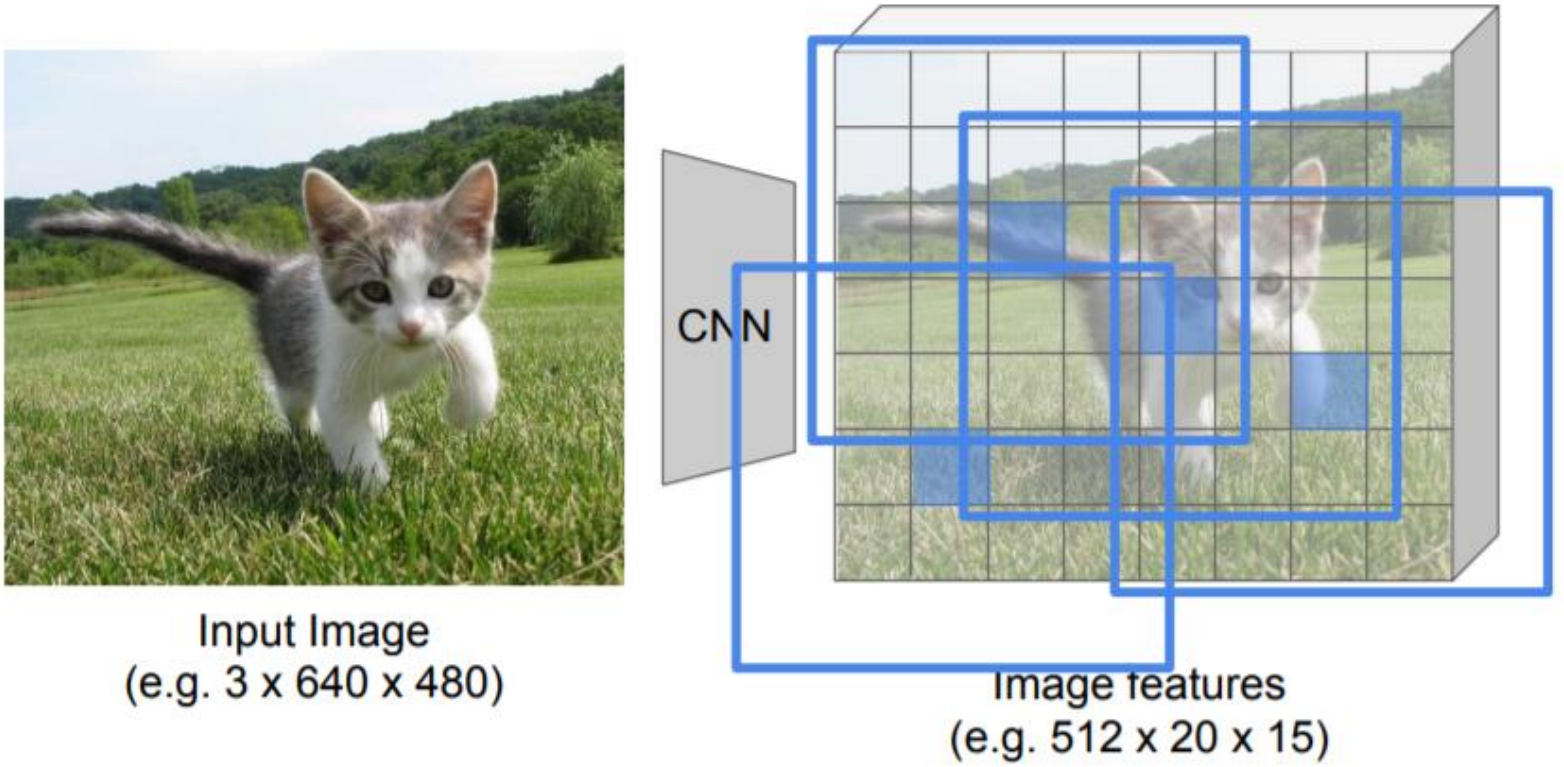
1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates



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

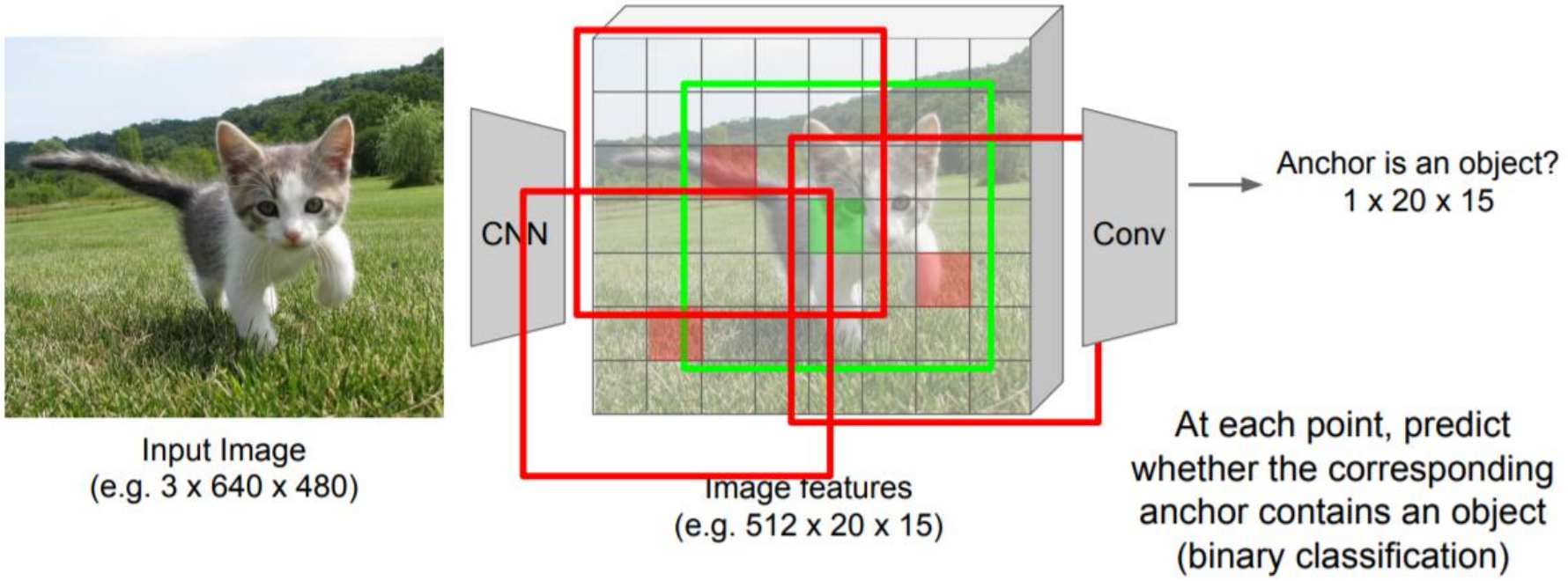
Region Proposal Network

- Imagine an anchor box of fixed size at each point in the feature map



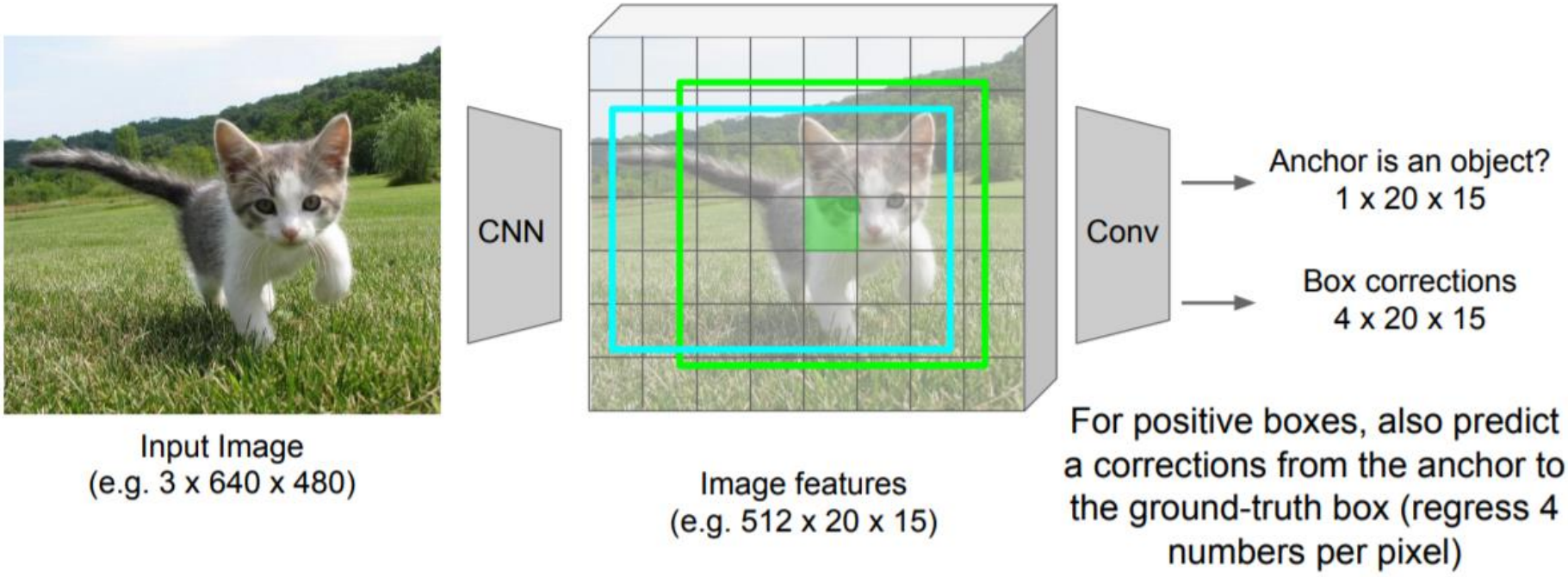
Region Proposal Network

- Imagine an anchor box of fixed size at each point in the feature map



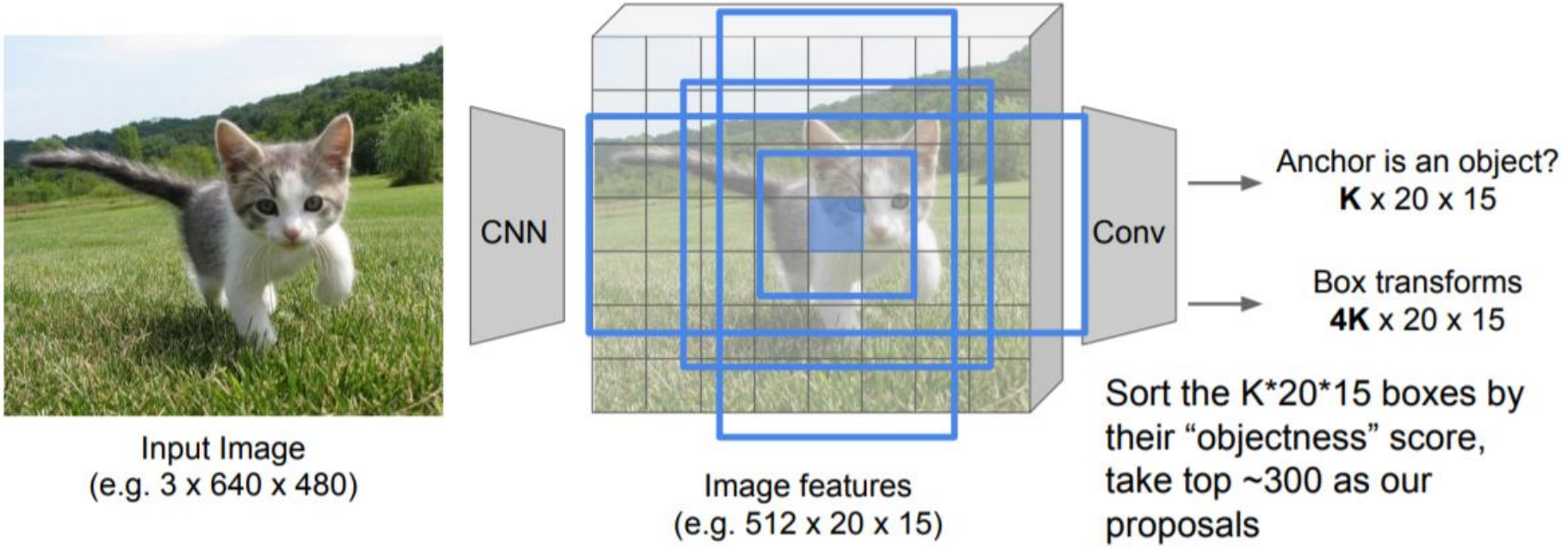
Region Proposal Network

- Imagine an anchor box of fixed size at each point in the feature map



Region Proposal Network

- In practice use K different anchor boxes of different size / scale at each point



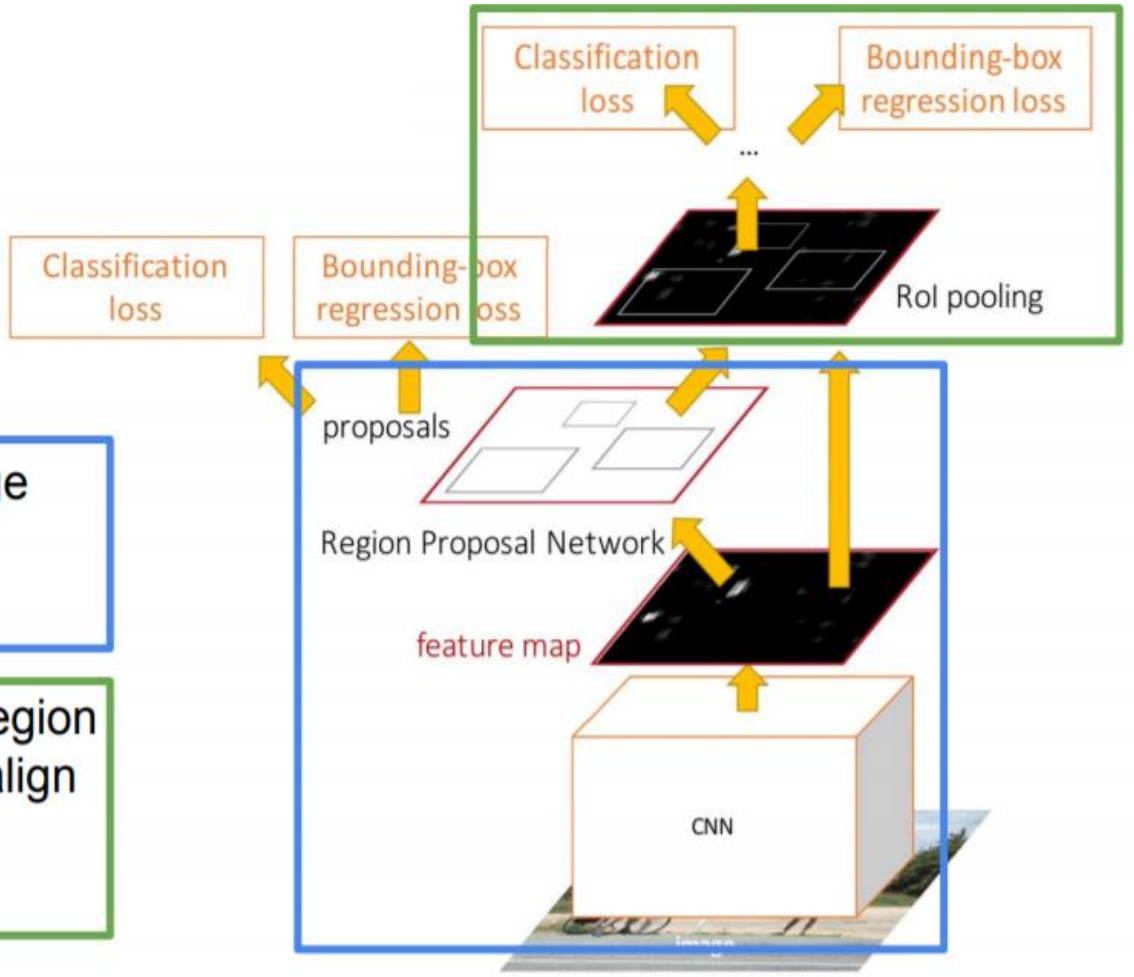
Faster R-CNN

- Region proposal network
  - Make CNN do proposal generation

Faster R-CNN is a
Two-stage object detector

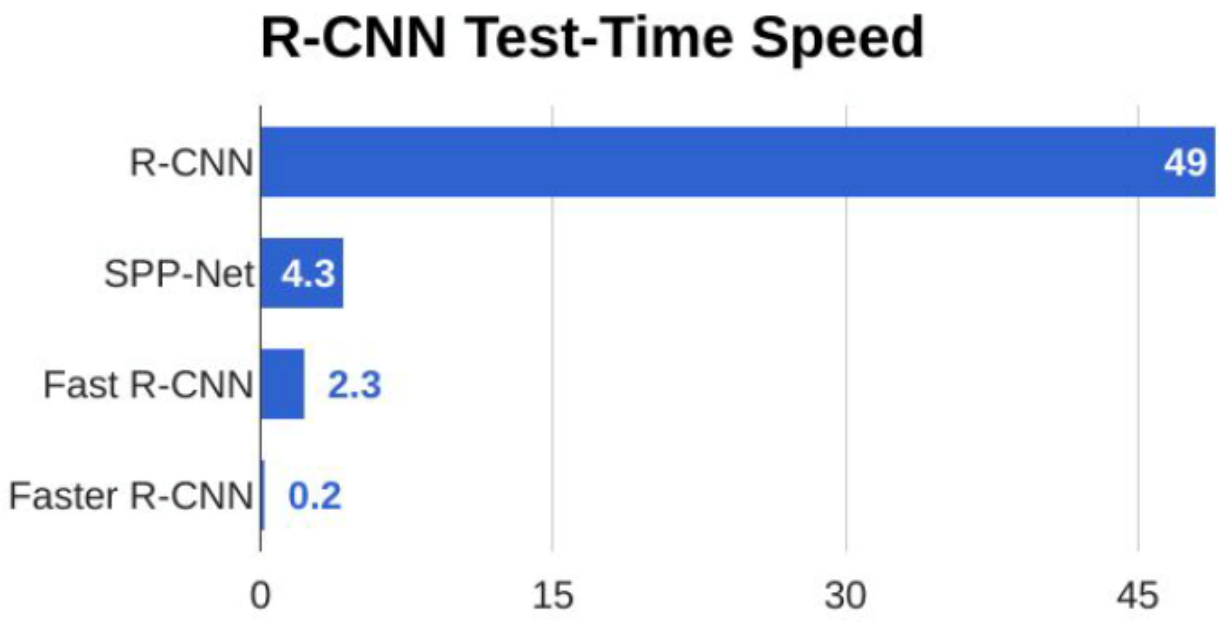
First stage: Run once per image
- Backbone network
- Region proposal network

Second stage: Run once per region
- Crop features: RoI pool / align
- Predict object class
- Prediction bbox offset



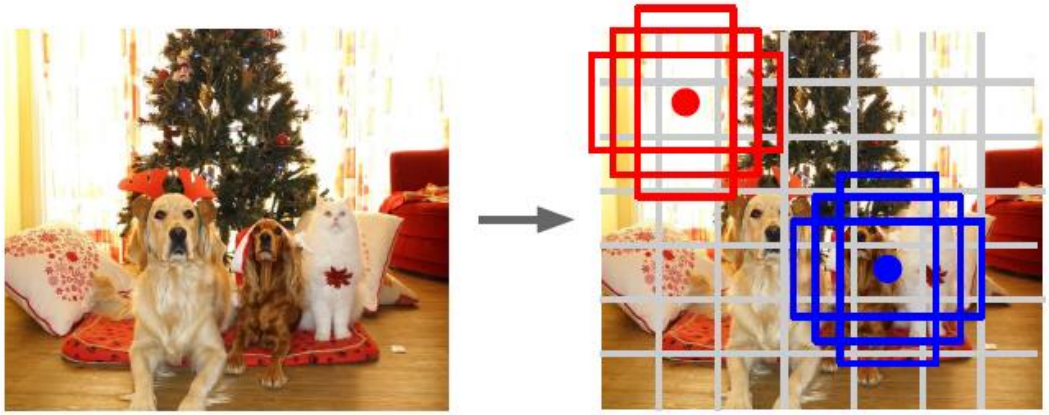
Faster R-CNN

- 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

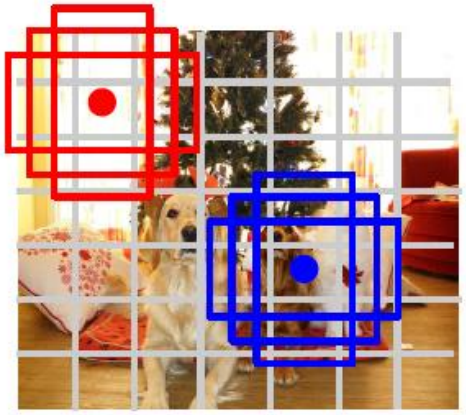
Image a set of **base boxes** centered at each grid cell
Here B = 3

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)

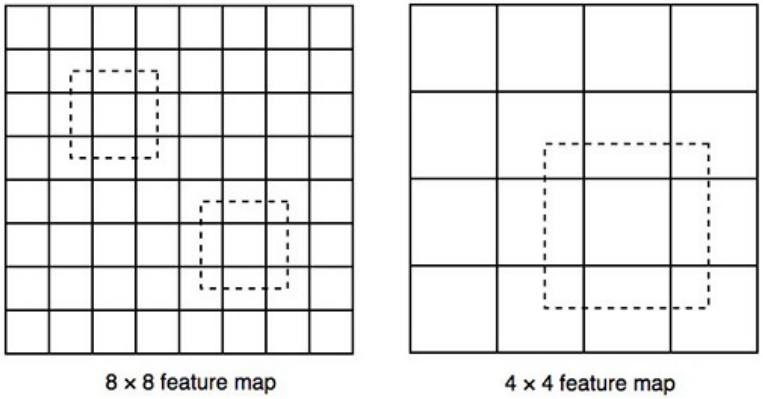
Output:
7 x 7 x (5 * B + C)

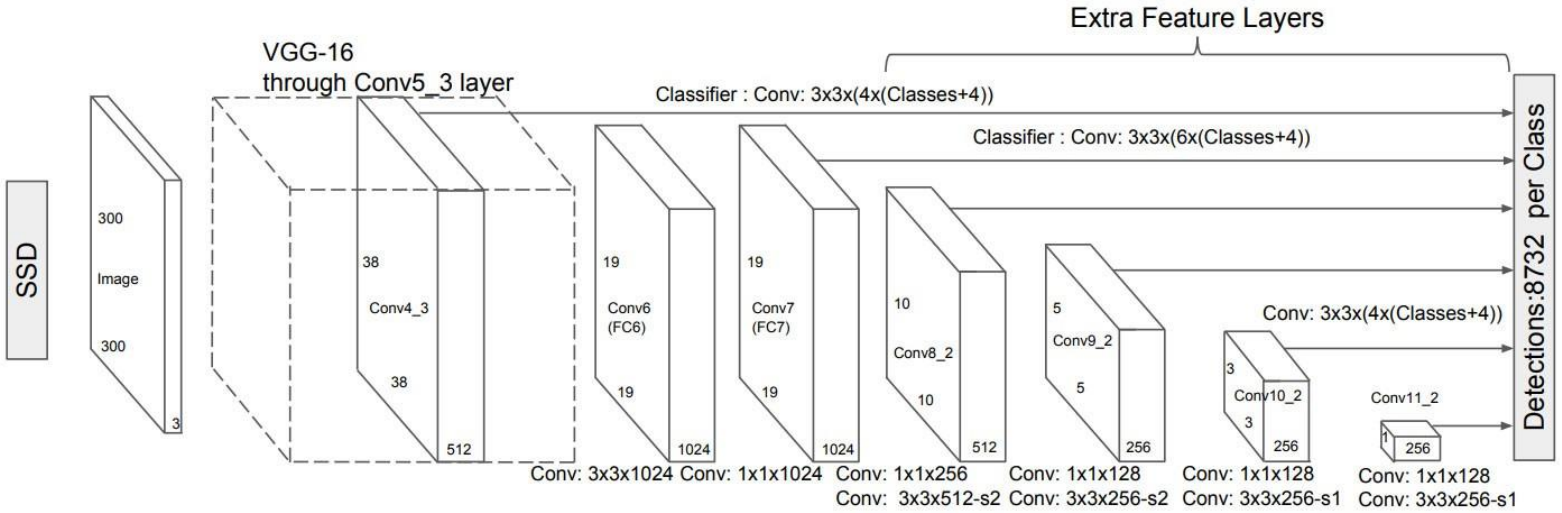
C class scores per location

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

- SSD: multi-scale feature maps





Object Detection Summary

- Many configurations

Backbone Network
VGG16
ResNet-101
Inception V2
Inception V3
Inception
ResNet
MobileNet

"Meta-Architecture"
Two-stage: Faster R-CNN
Single-stage: YOLO / SSD
Hybrid: R-FCN

Image Size
 # Region Proposals
…

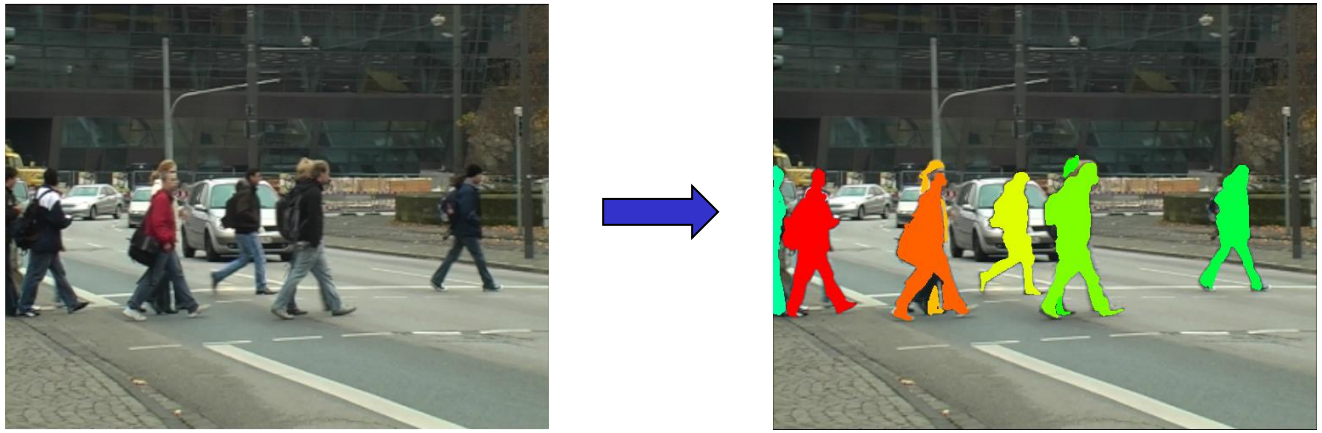
Takeaways
Faster R-CNN is slower but more accurate

SSD is much faster but not as accurate

Bigger / Deeper backbones work better

Huang et al, "Speed/accuracy trade-offs for modern convolutional object detectors", CVPR 2017
Zou et al, "Object Detection in 20 Years: A Survey", arXiv 2019

Outline



- Object Detection
- Semantic Instance Segmentation

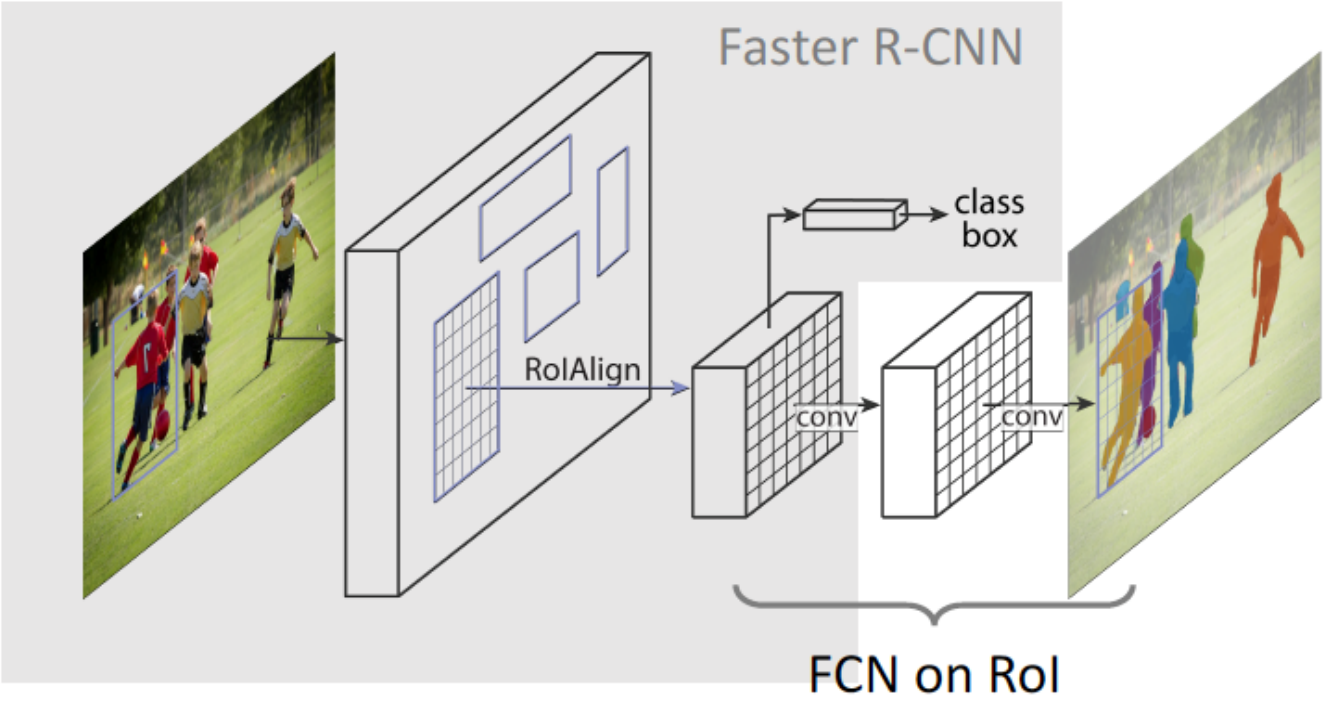
Acknowledgement: Feifei Li et al's cs231n notes

Object Instance Segmentation

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

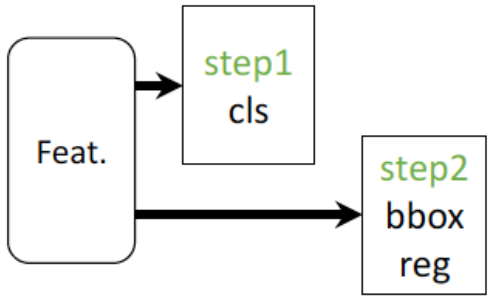
Mask R-CNN

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

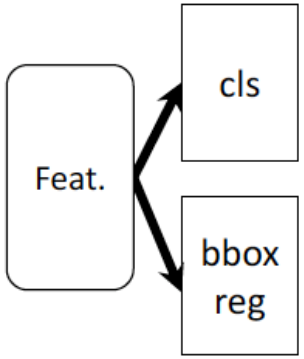


Mask R-CNN

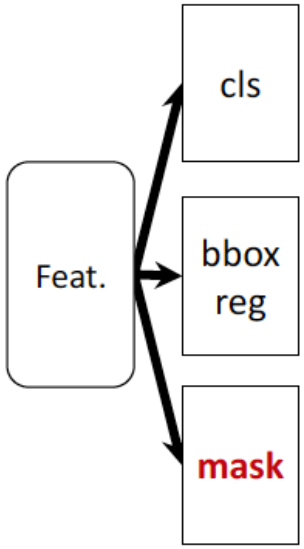
- Parallel heads
  - Easy, fast to implement and train



(slow) R-CNN



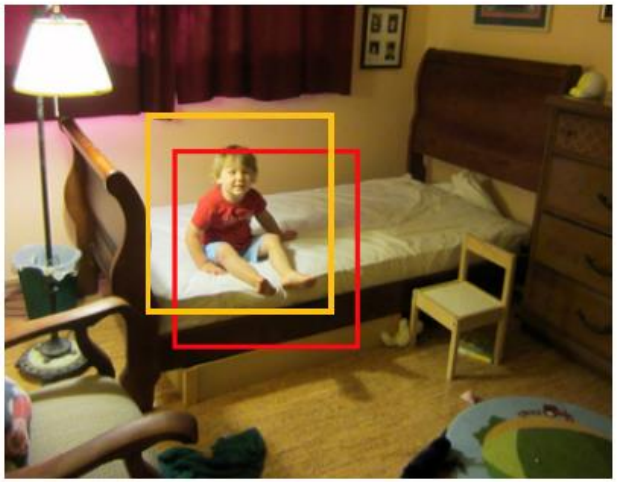
Fast/er R-CNN



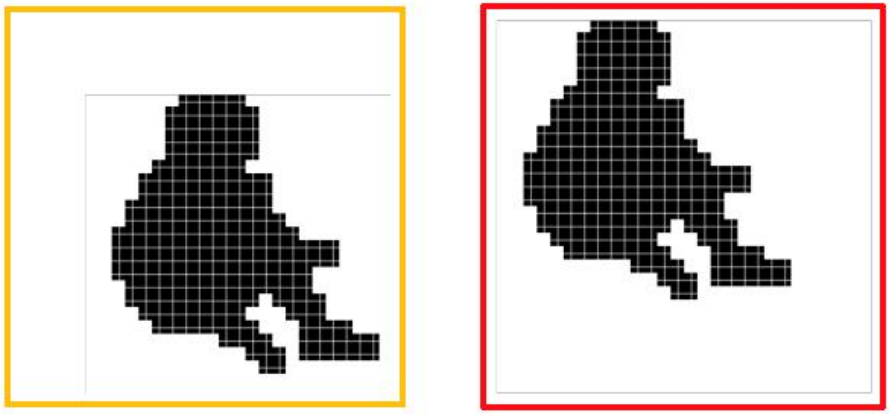
Mask R-CNN

Mask R-CNN

- Fully-Convolution on RoI



target masks on RoIs

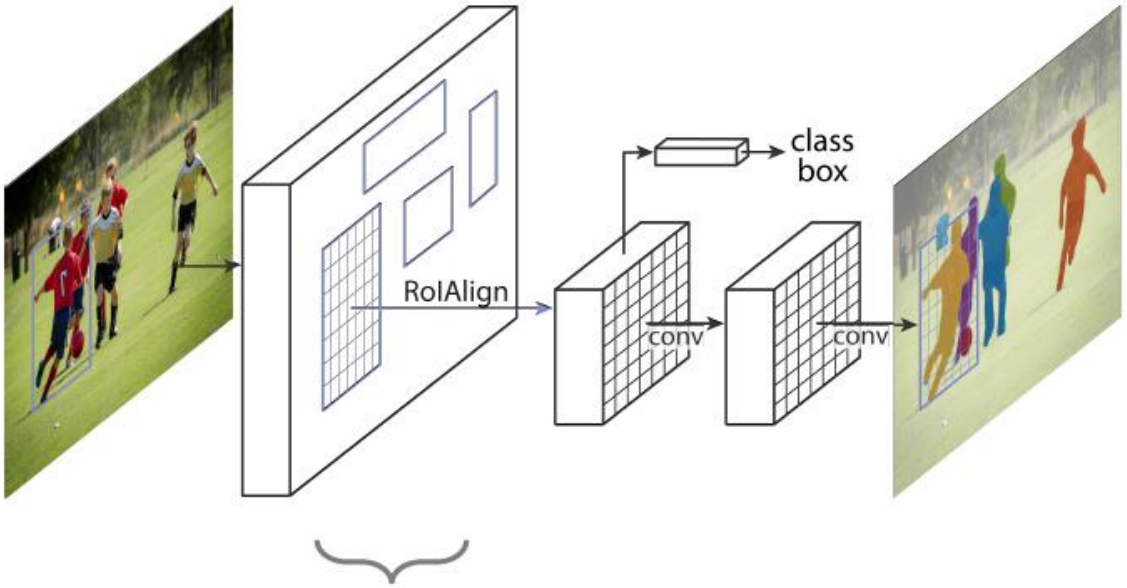


Translation of object in RoI => Same translation of mask in RoI

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

Mask R-CNN

- Fully-Convolution on RoI



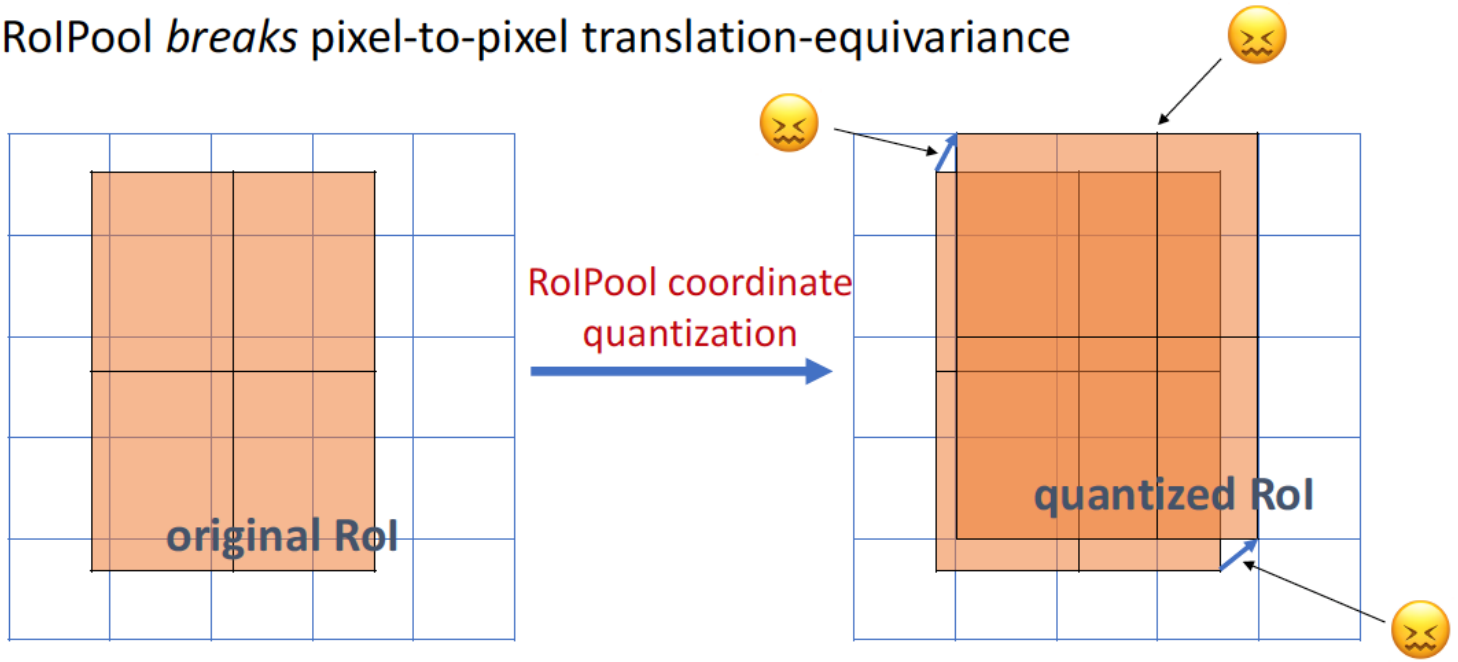
3. RoIAlign:

3a. maintain translation-equivariance before/after RoI

Mask R-CNN

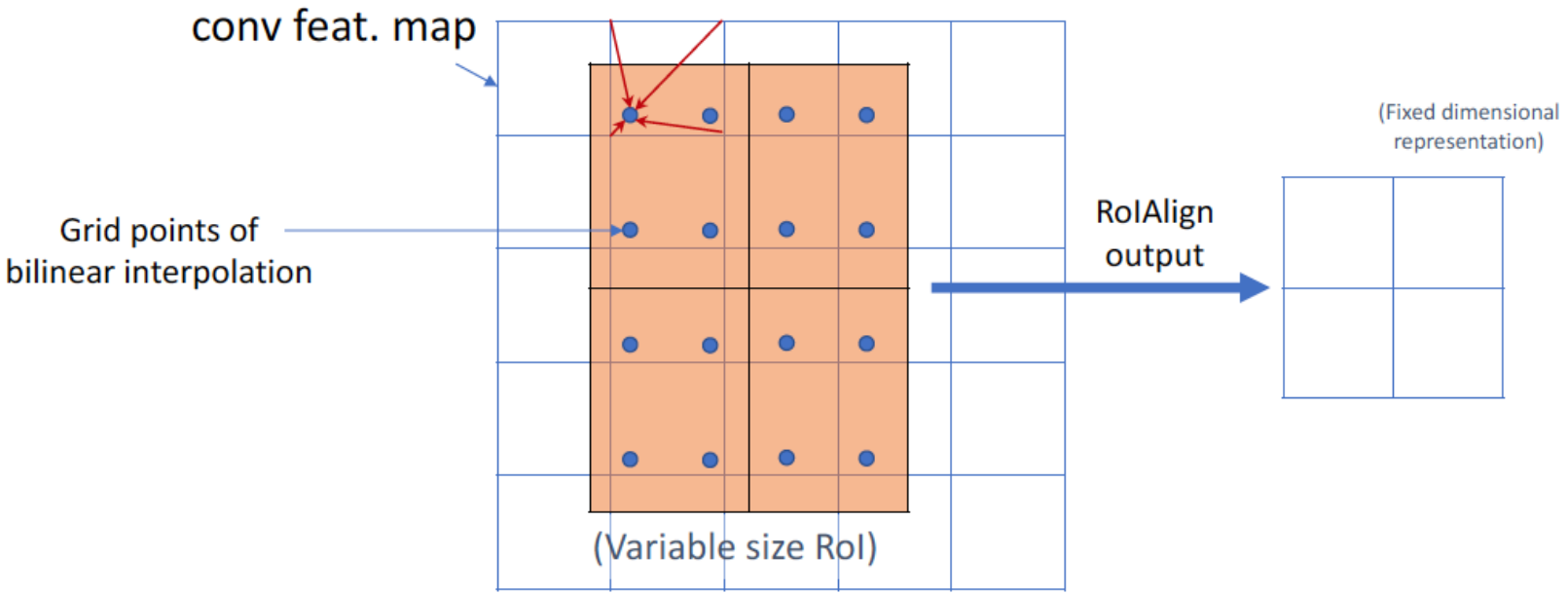
- Fully-Convolution on RoI
  - RoIAlign vs RoIPool

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



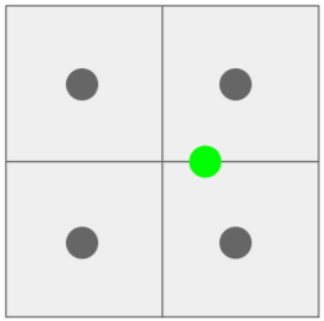
Mask R-CNN

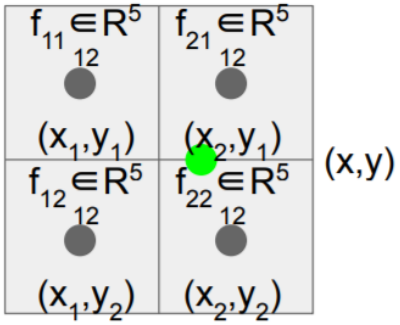
- Fully-Convolution on RoI
  - RoIAlign



Mask R-CNN

- Fully-Convolution on RoI
  - RoIAlign
  - Sample at regular points in each subregion using bilinear interpolation



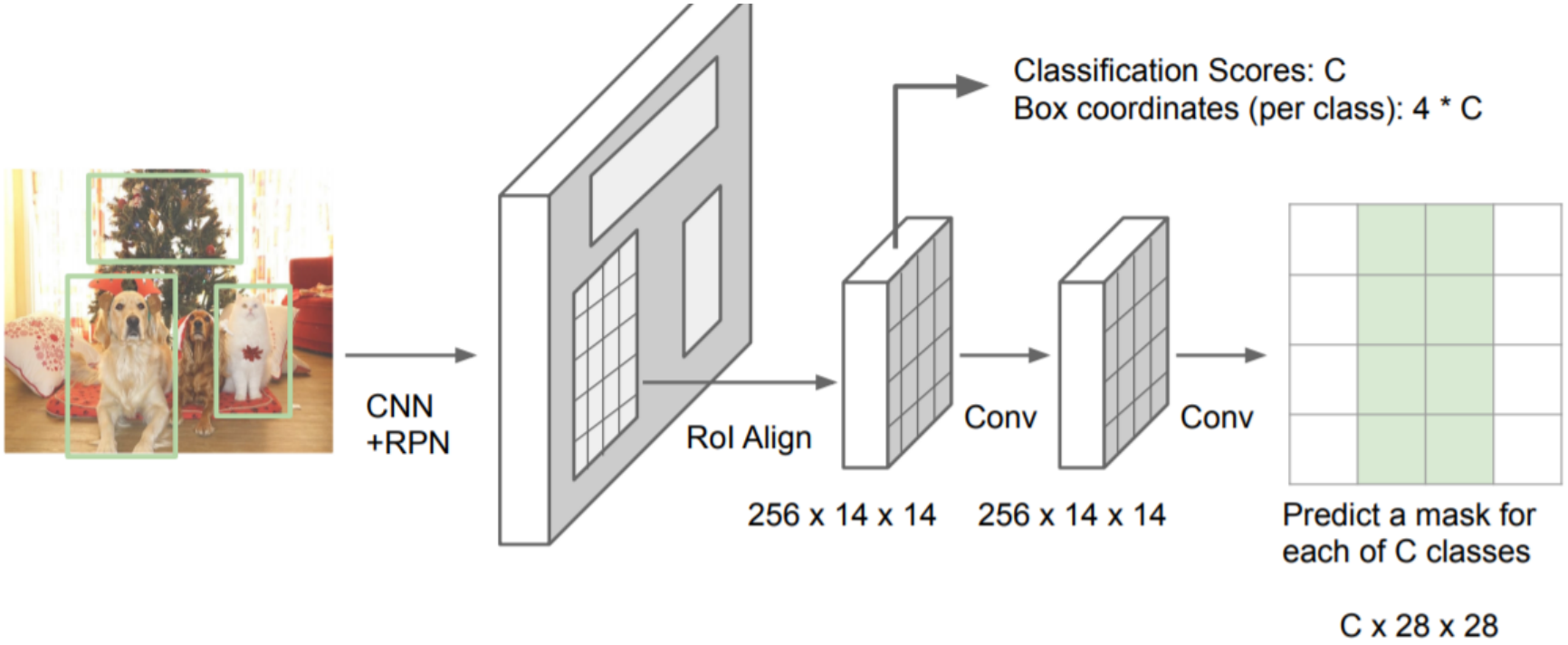


  - Feature f_{xy} for point (x, y) is a linear combination of features at its four neighboring grid cells:

$$f_{xy} = \sum_{i,j=1}^{2} f_{i,j} \max(0, 1 - |x - x_i|) \max(0, 1 - |y - y_j|)$$

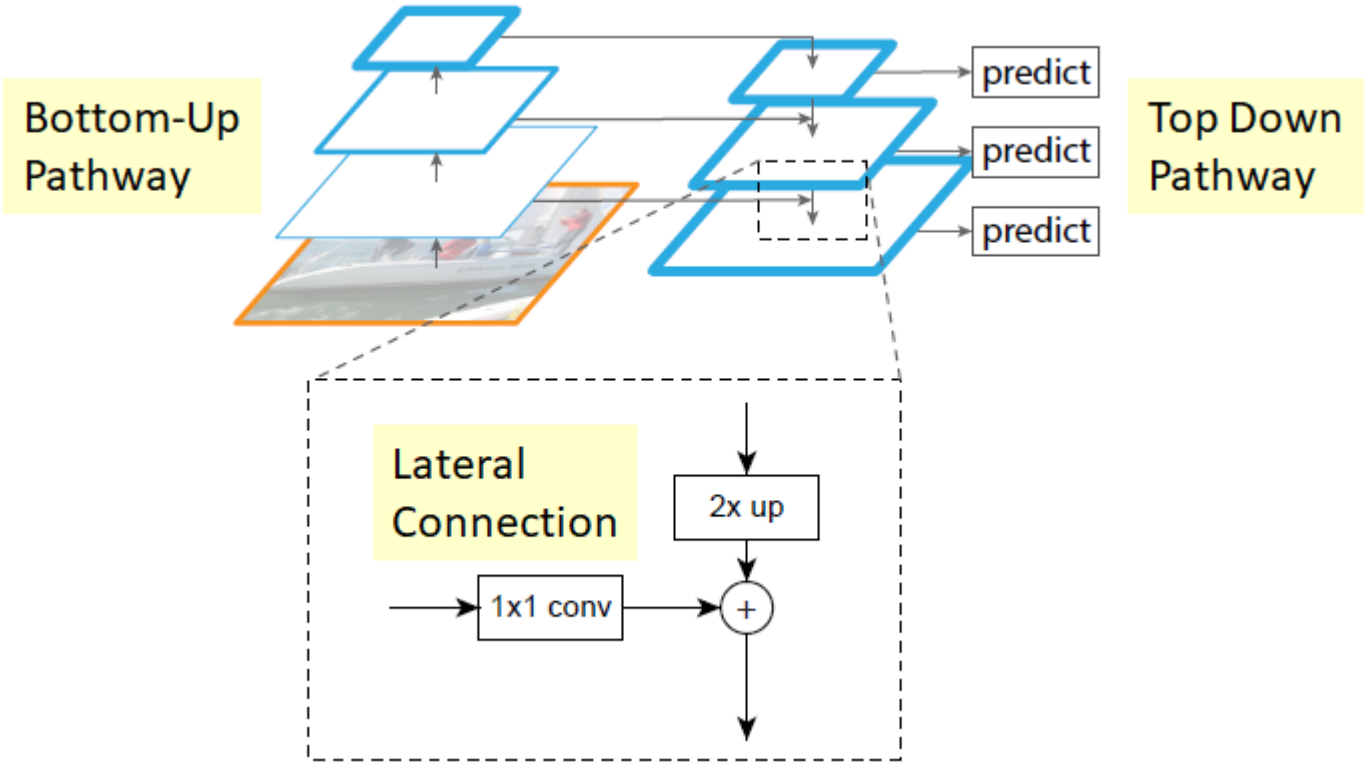
Mask R-CNN

- Overall architecture



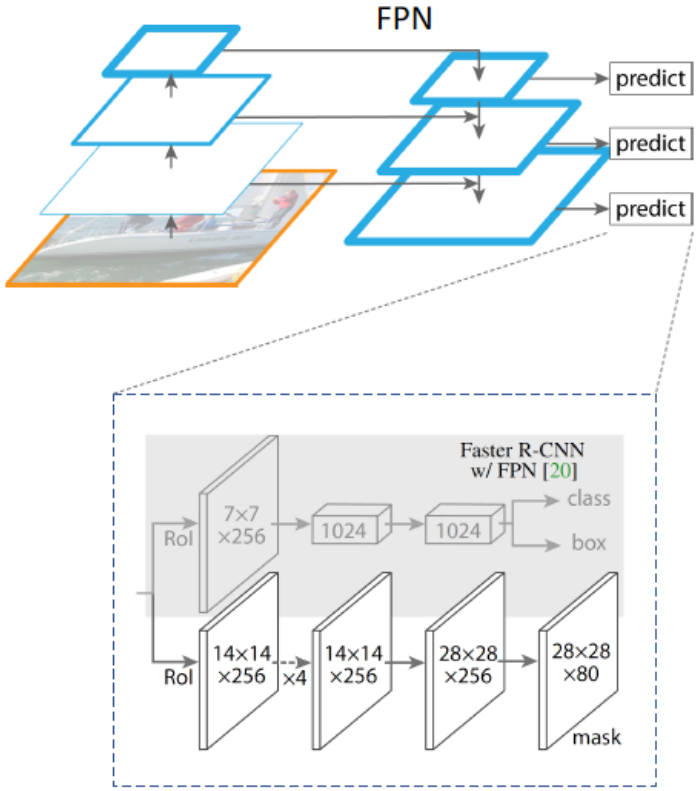
Mask R-CNN

- Multiscale representation
  - Feature Pyramid Network (FPN)



Mask R-CNN

- Multiscale representation
  - Feature Pyramid Network (FPN)



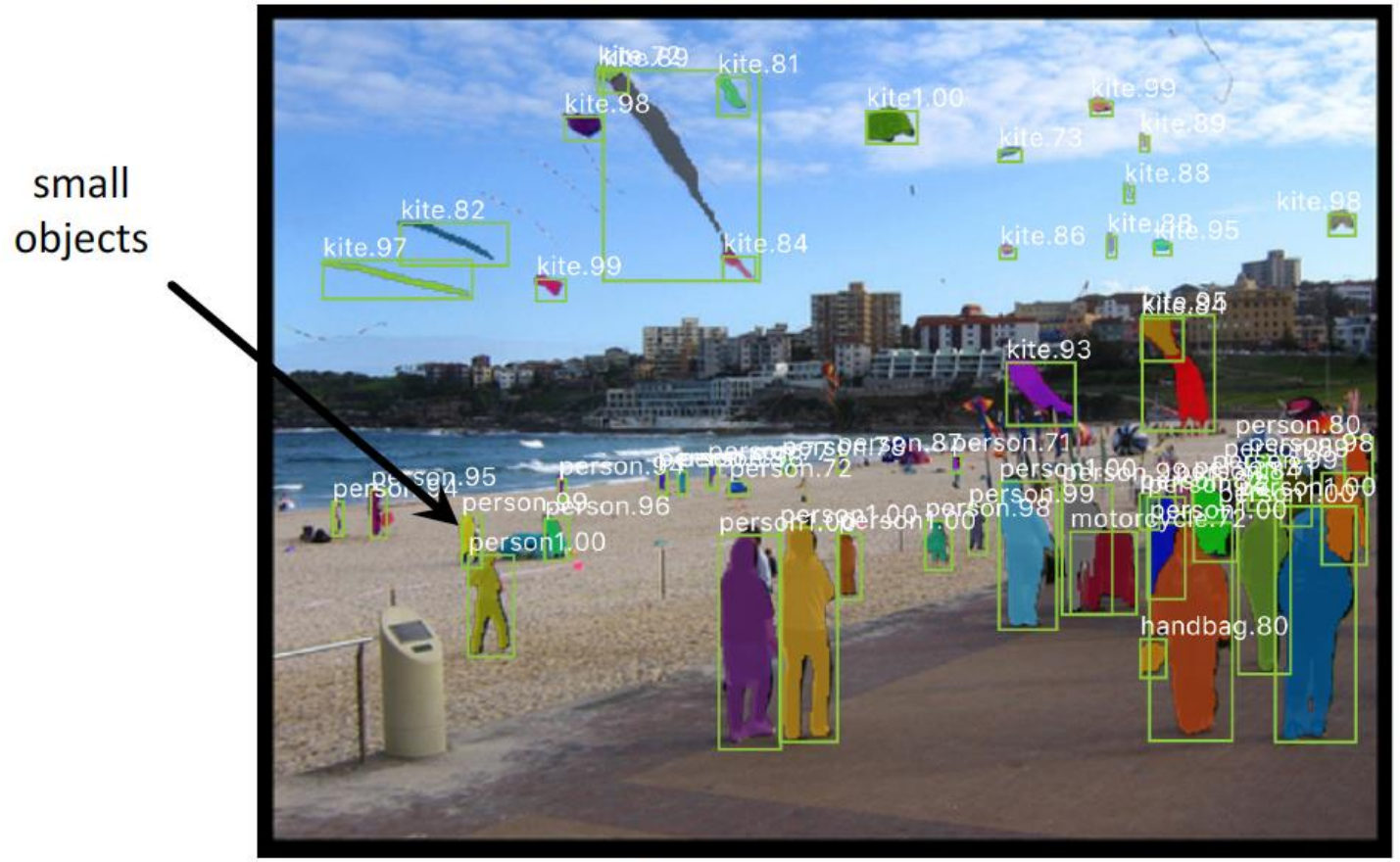
Mask R-CNN

- Results

Mask R-CNN results on COCO

Mask R-CNN

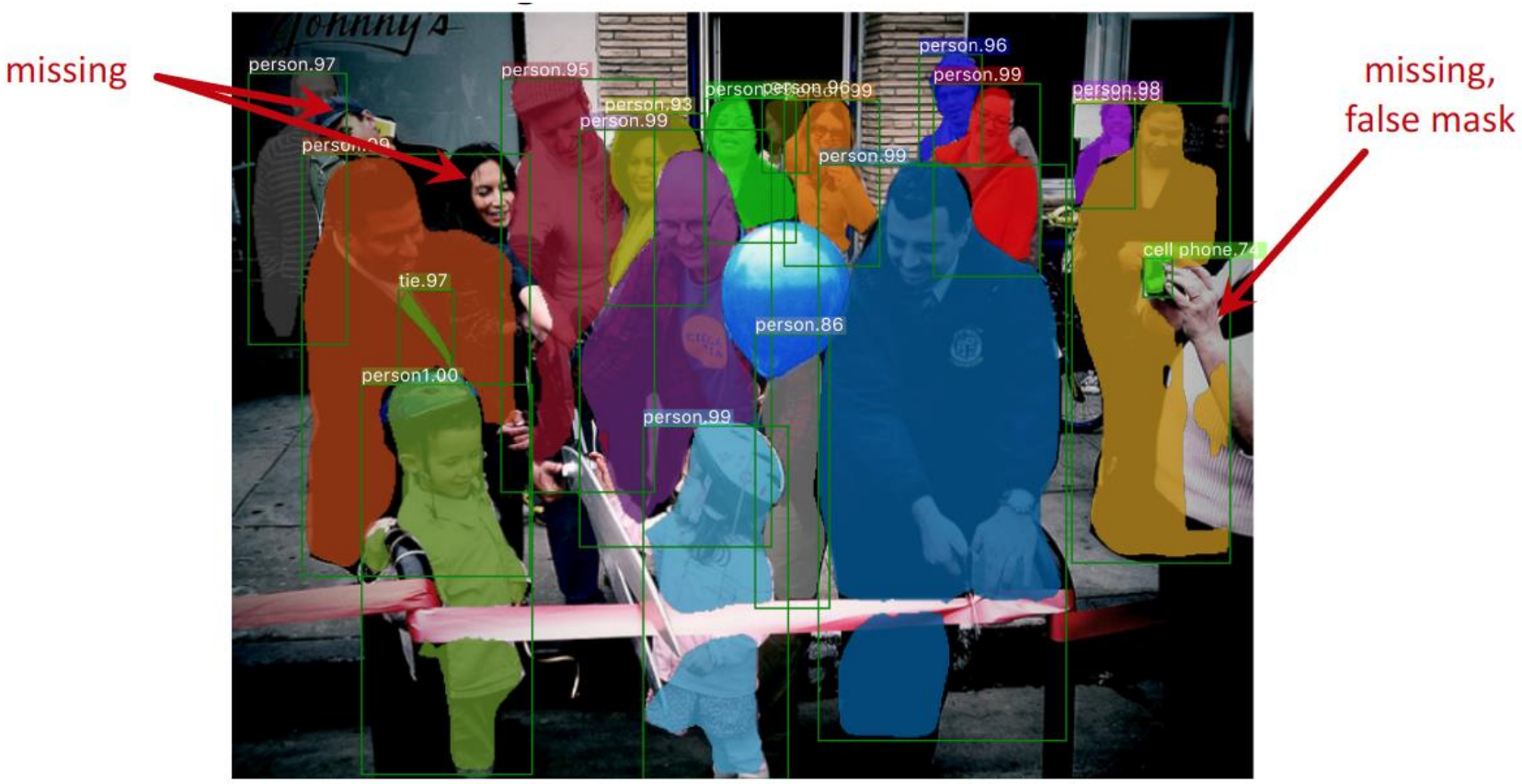
- Results



Mask R-CNN results on COCO

Mask R-CNN

- Results



Mask R-CNN results on COCO

Mask R-CNN

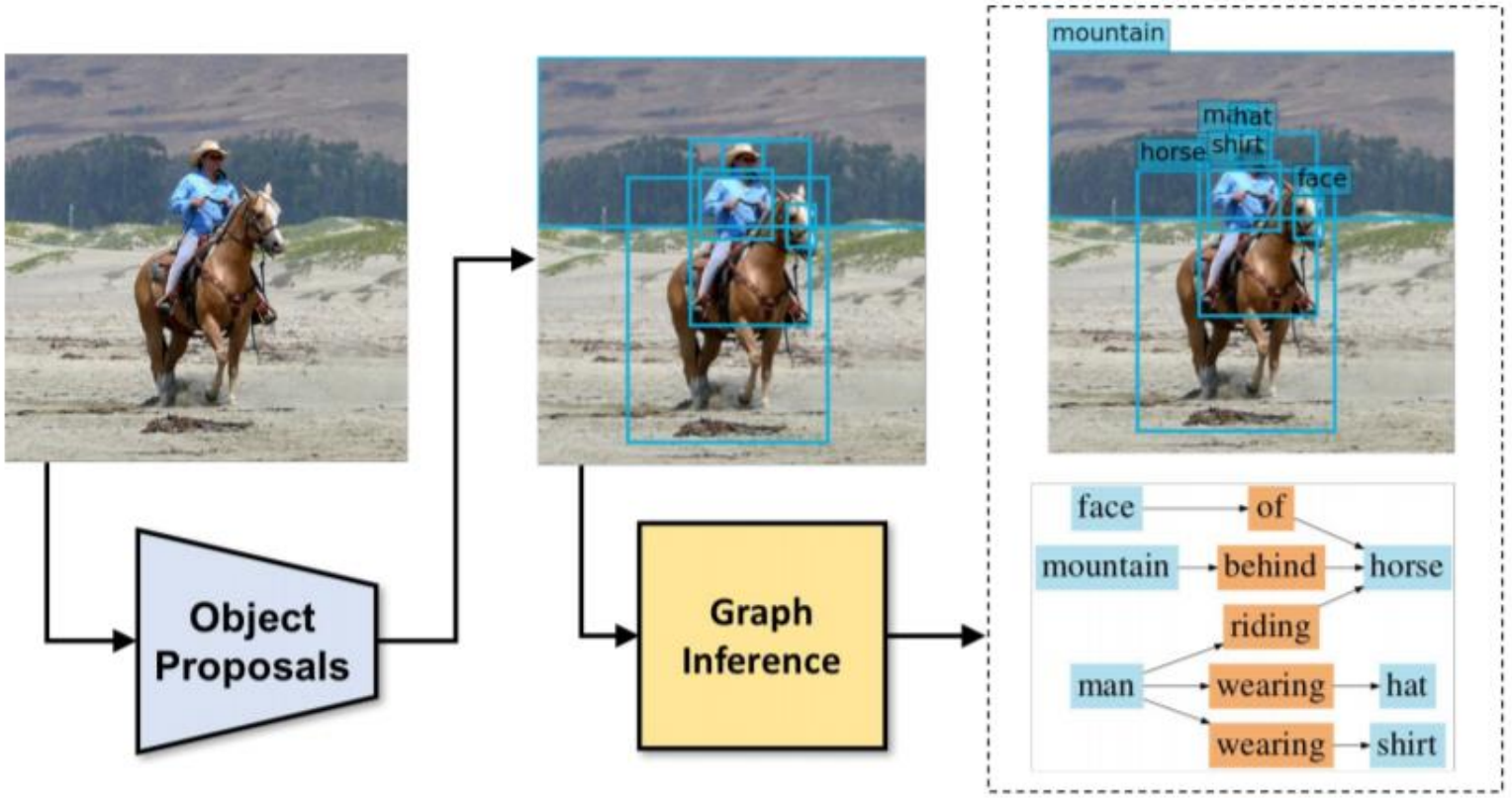
- Pose results

Open Source Implementations

- TensorFlow Detection API: https://github.com/tensorflow/models/tree/master/research/object_detection
  - Faster RCNN, SSD, RFCN, Mask R-CNN, ...
- Detectron2 (PyTorch) https://github.com/facebookresearch/detectron2
  - Mask R-CNN, RetinaNet, Faster R-CNN, RPN, Fast R-CNN, R-FCN, ...
- Finetune on your own dataset with pre-trained models

Other tasks

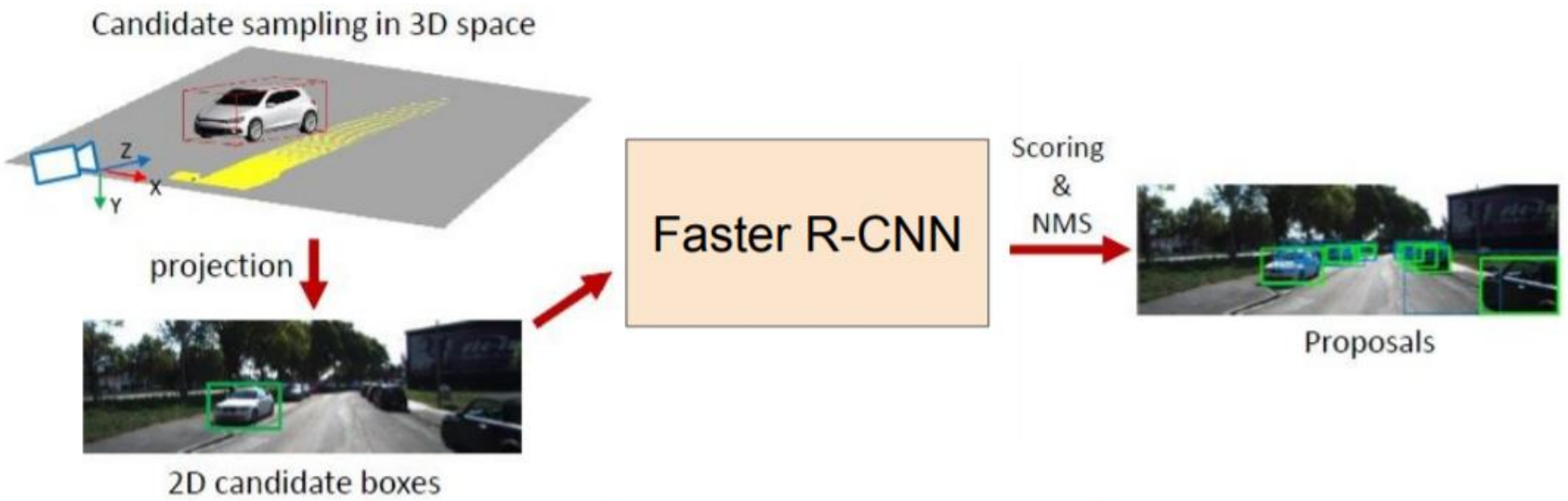
- Objects + Relationships = Scene Graphs



Xu, Zhu, Choy, and Fei-Fei, "Scene Graph Generation by Iterative Message Passing", CVPR 2017

Other tasks

- 3D Object Detection: Single view

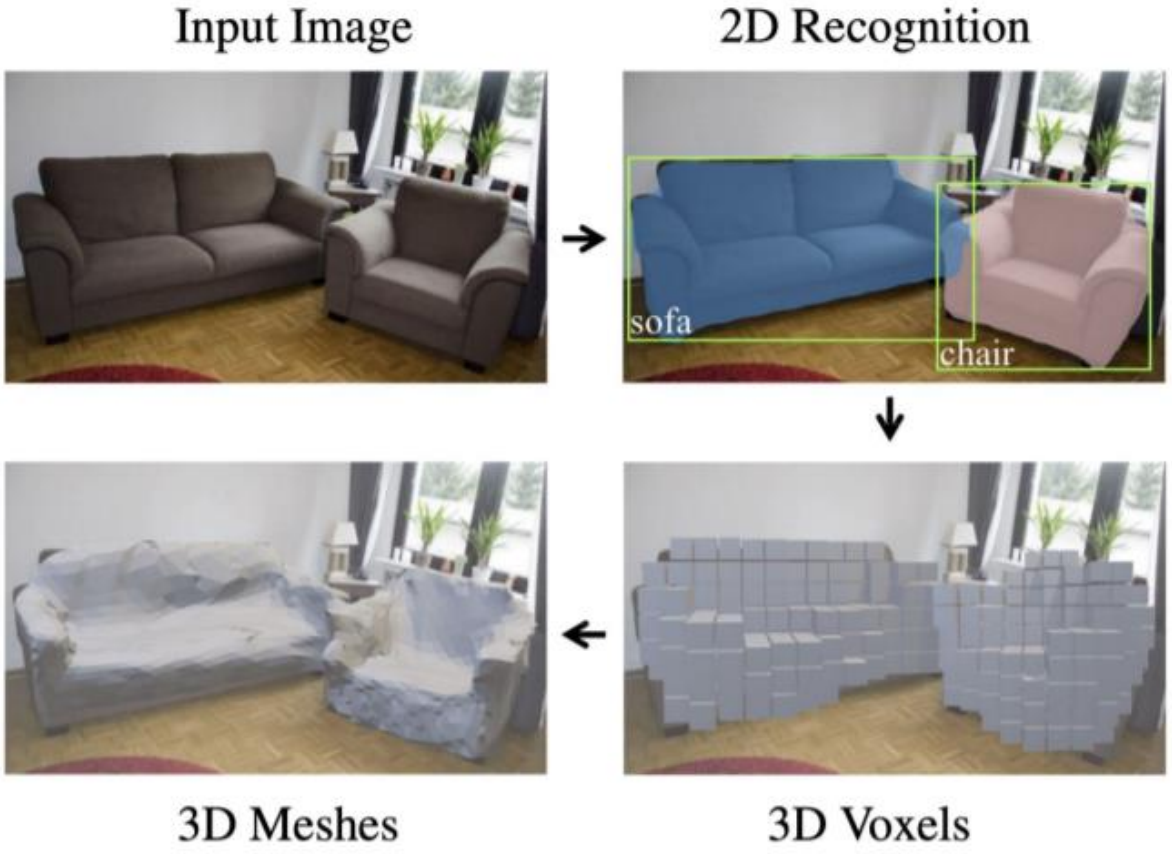


- Same idea as Faster RCNN, but proposals are in 3D
- 3D bounding box proposal, regress 3D box parameters + class score

Chen, Xiaozhi, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. "Monocular 3d object detection for autonomous driving." CVPR 2016.

Other tasks

- 3D Shape Prediction: Mesh R-CNN



Gkioxari et al., Mesh RCNN, ICCV 2019

Summary

- CNNs in computer vision
  - Localization, Detection
  - Instance segmentation

- Next time:
  - Understanding CNNs
  - Limitations of CNNs