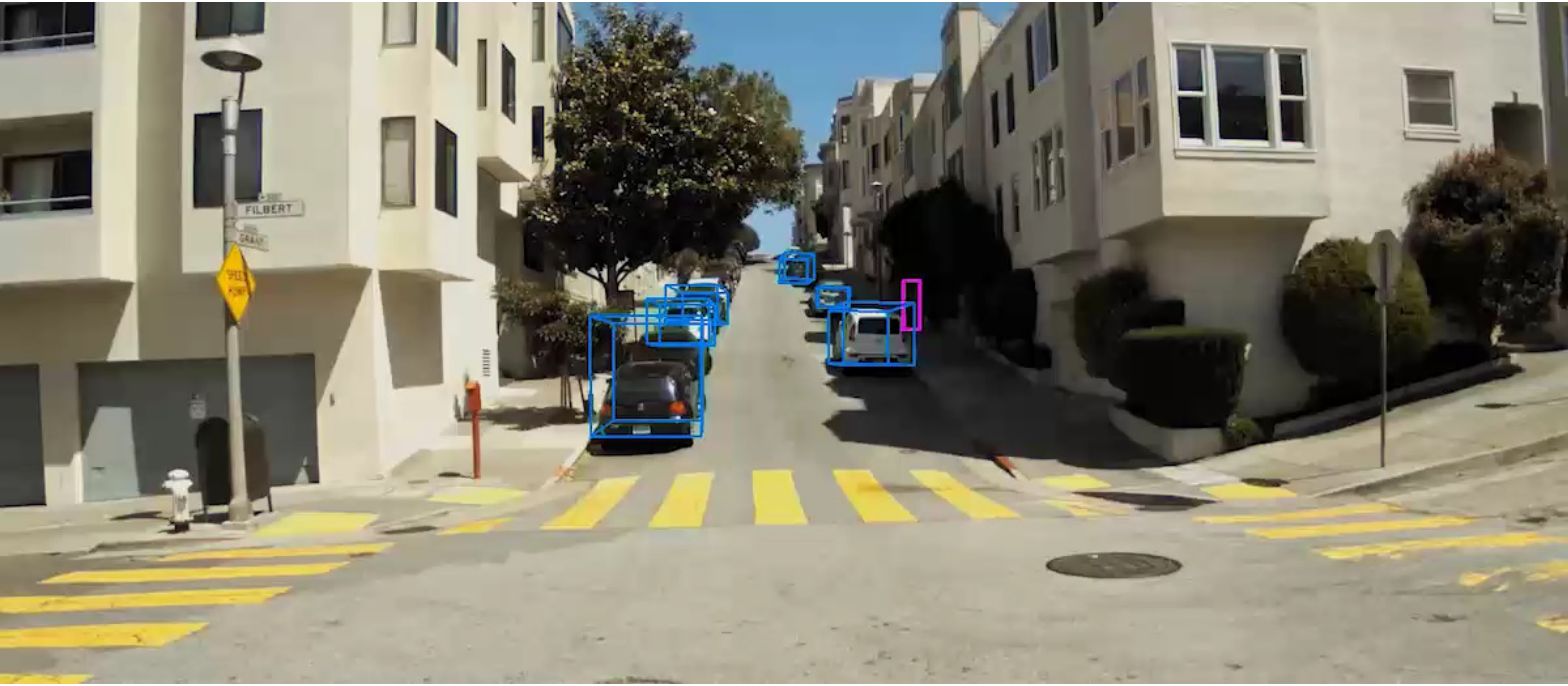
2D and 3D Object Detection

for autonomous driving

NVIDIA AGX PEGASUS TEST DRIVE

OCTOBER 2, 2018

- 80 KILOMETERS
- 4 HIGHWAY INTERCHANGES
- 10 LANE CHANGES
- 0 DISENGAGEMENTS


FILBERT
GRANT

Outline

Part I: Introduction

Part II: 2D Object Detection

- ❑ Fully Convolutional Network for Semantic Segmentation
- ❑ Faster R-CNN for Object Detection

Part III: 3D Segmentation and Detection

Slides from Olga Veksler, Raquel Urtasun, Ross Girshick, Jonathan Long, Ricardo Vilalta, Fei-fei Li, Justin Johnson, Serena Yeung

Part I: Introduction

Perception in Autonomous Driving

- Detection
- Tracking
- Semantic Segmentation
- Instance-level Segmentation

Object Detection

Task: Bounding box around the object of interest and determine its class

Dominated by deep learning

Tracking

Task: Place bounding boxes at each frame, and link them over time

Semantic Segmentation

Task: Label each pixel with a semantic category

Dominated by deep learning + graphical models

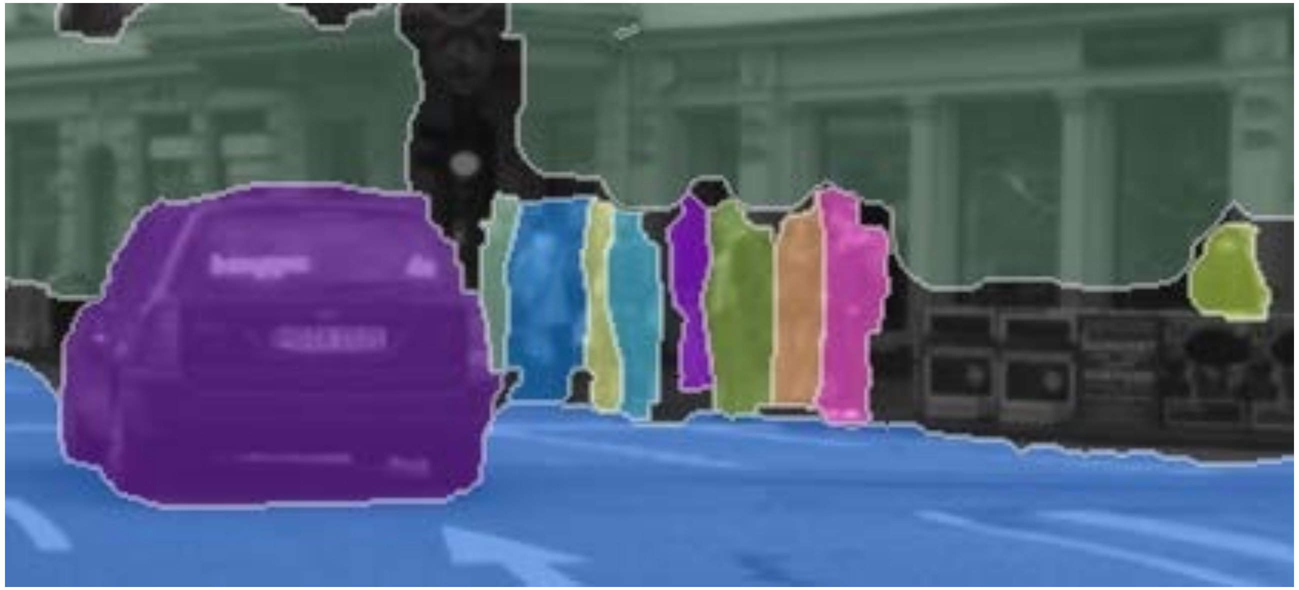
Instance-level Segmentation

Task: Label each pixel with an instance number

Difficult as labeling is agnostic to permutation of the labels

Very little work on this topic

Dominated by deep learning + graphical models



Challenges: viewpoint variation

Michelangelo 1475-1564

Challenges: illumination variation

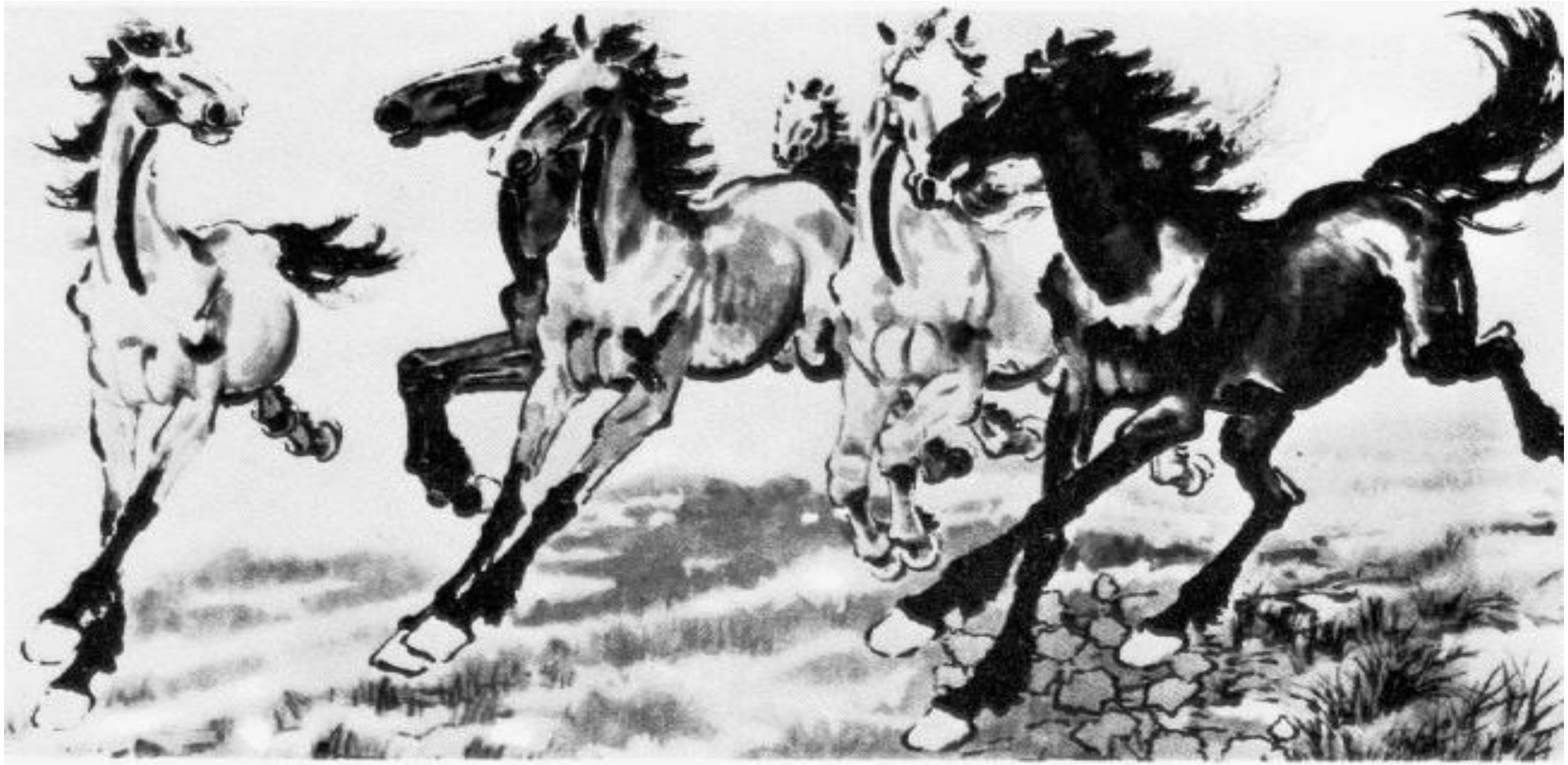
slide credit: S. Ullman

Challenges: occlusion

Magritte, 1957

Challenges: scale

Challenges: deformation



Xu, Beihong 1943

Challenges: background clutter

Klimt, 1913

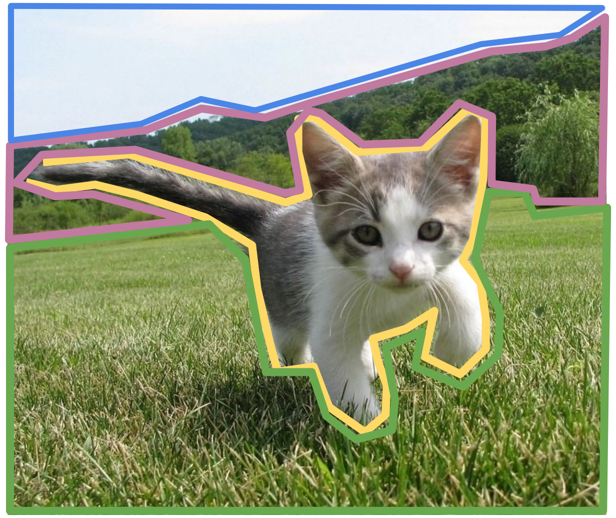
Challenges: intra-class variation






Part II: 2D Object Detection

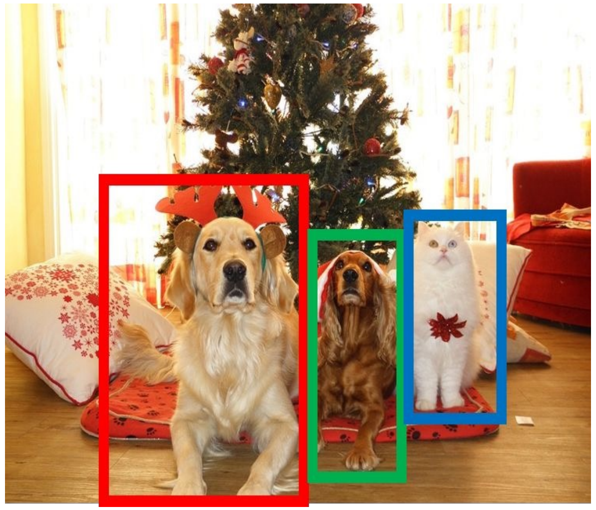
Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

Object Detection



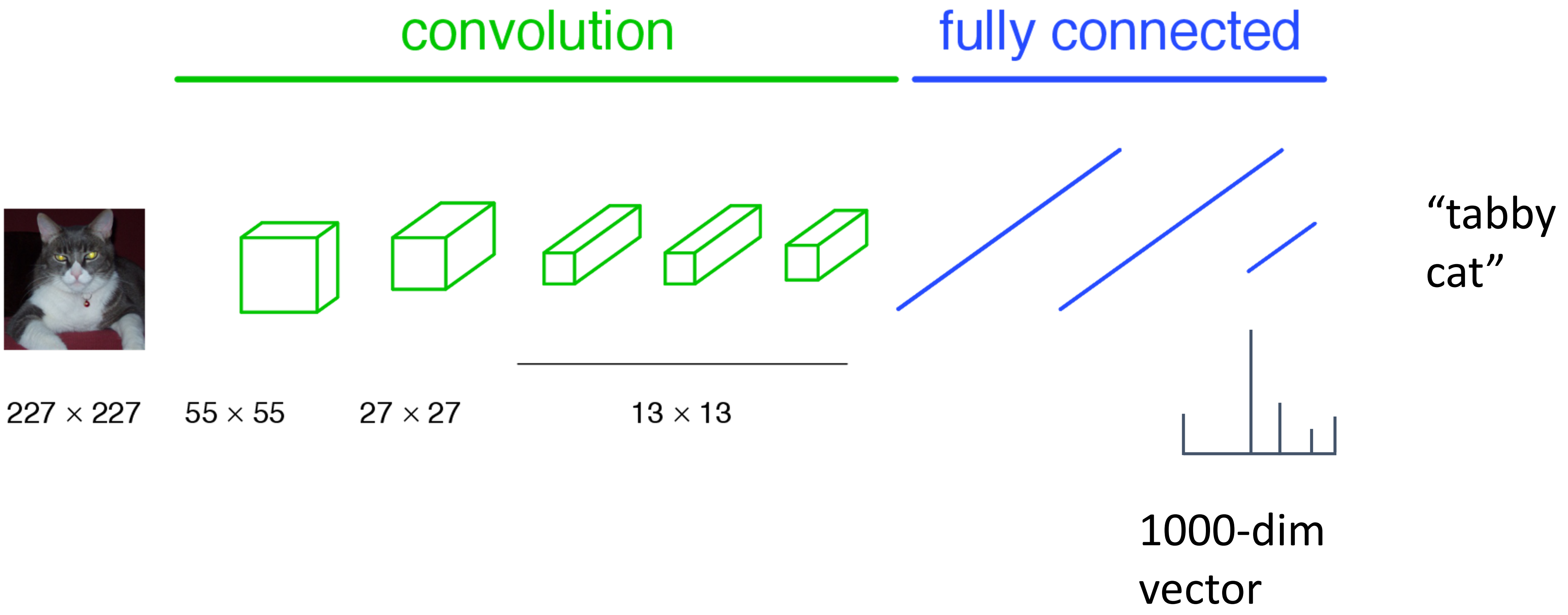
DOG, DOG, CAT

Instance Segmentation

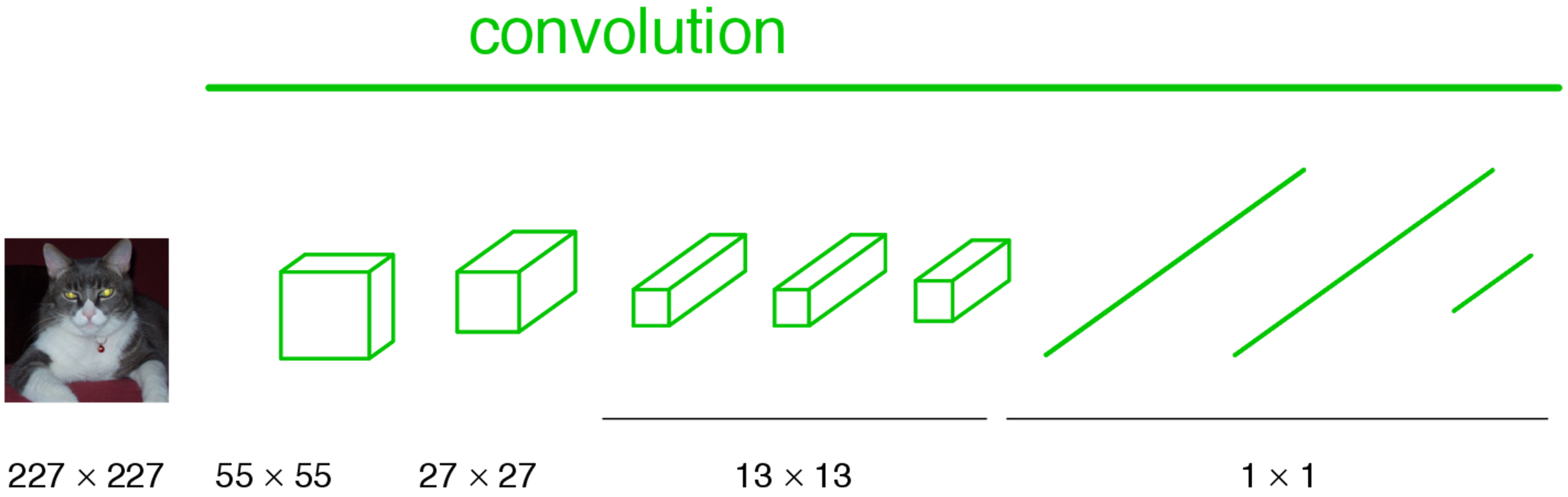
DOG, DOG, CAT

Multiple Object

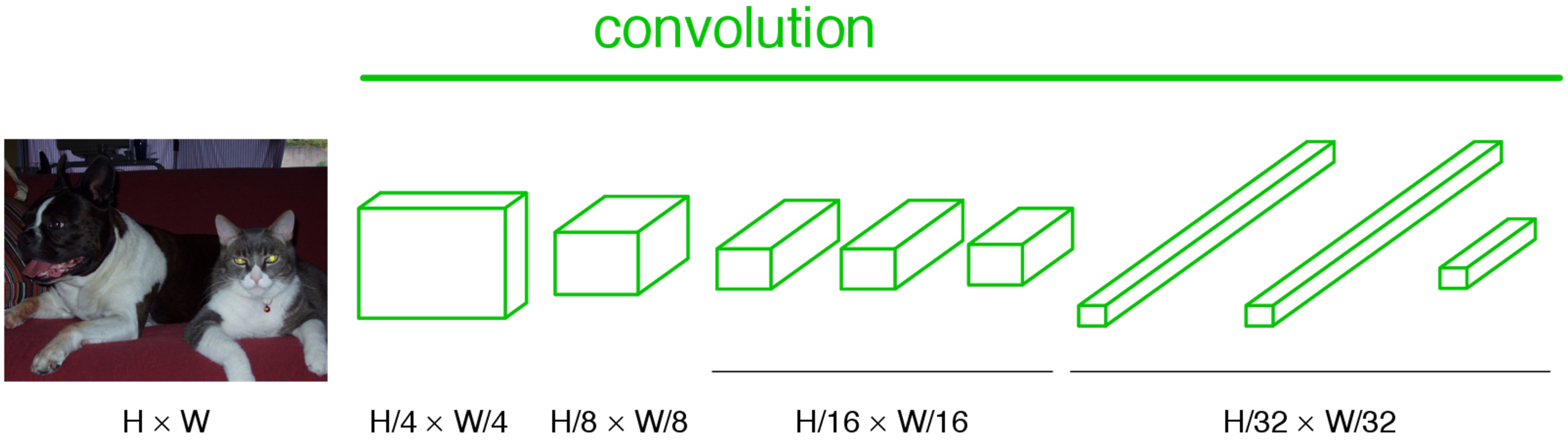
a classification network



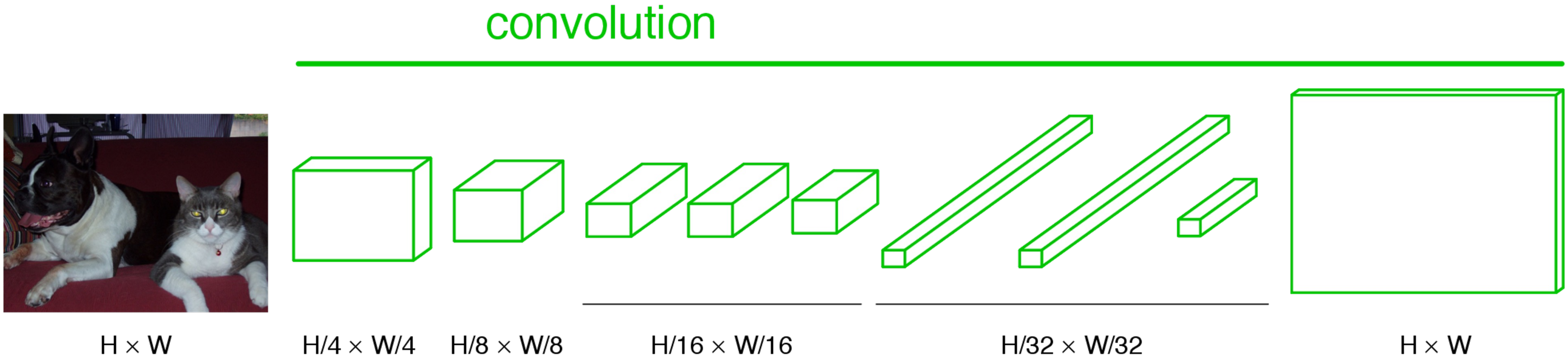
becoming fully convolutional



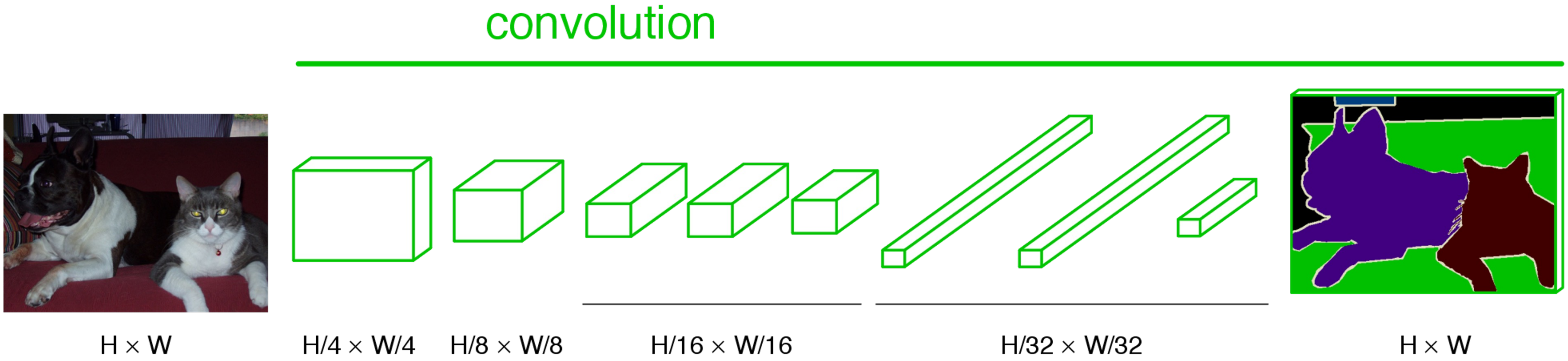
becoming fully convolutional



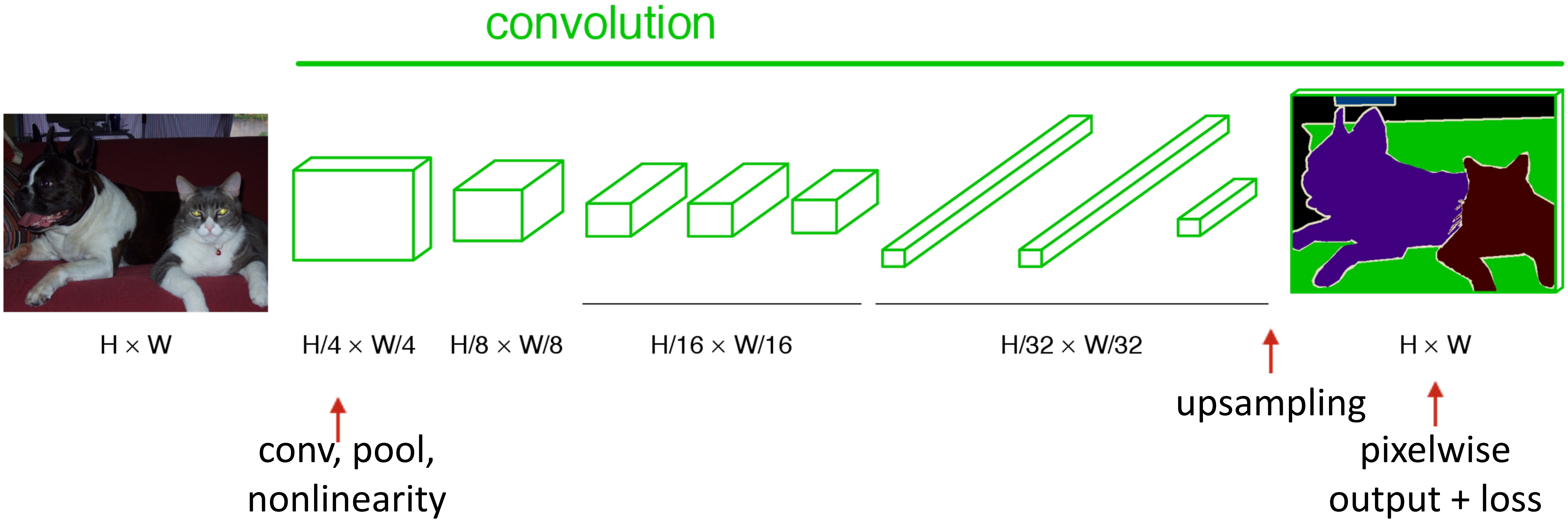
upsampling output



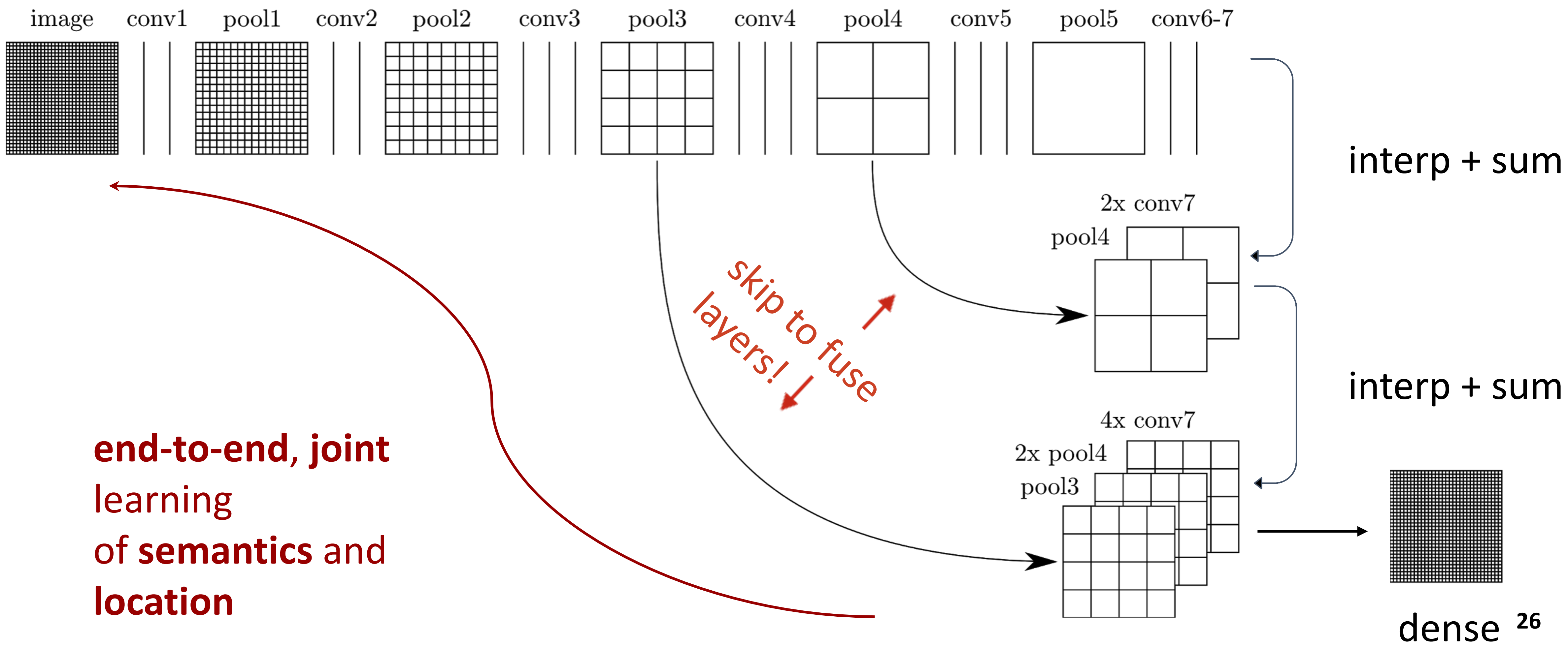
end-to-end, pixels-to-pixels network



end-to-end, pixels-to-pixels network



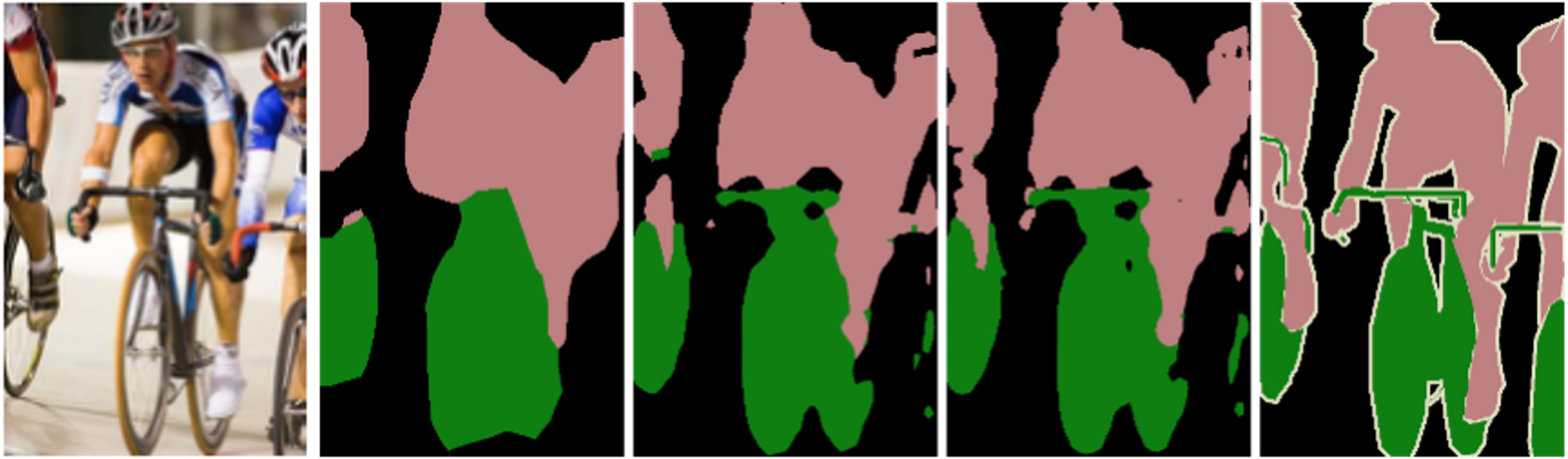
skip layers



end-to-end, **joint** learning
of **semantics** and **location**

skip layer refinement

input image | stride 32 | stride 16 | stride 8 | ground truth



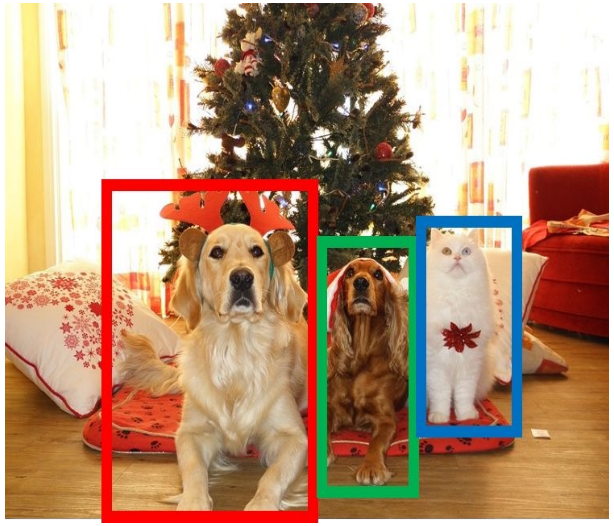
no skips | 1 skip | 2 skips

Semantic Segmentation

GRASS, CAT, TREE, SKY

No objects, just pixels

2D Object Detection



DOG, DOG, CAT

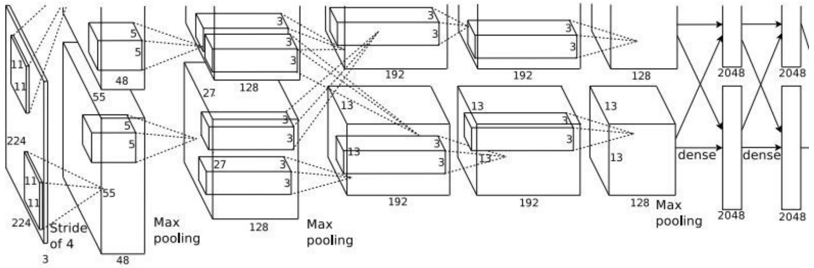
Object categories +
2D bounding boxes

3D Object Detection

Car

Object categories +
3D bounding boxes

Classification + Localization



Vector: 4096

Fully Connected: 4096 to 1000

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

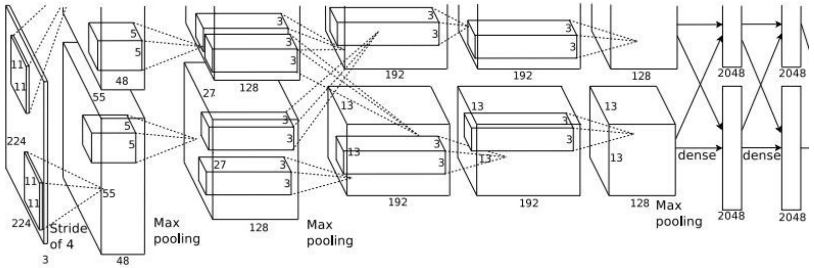
Fully Connected: 4096 to 4

Box Coordinates
(x, y, w, h)

Treat localization as a regression problem!

Classification + Localization

This image is CC0 public domain



Vector: 4096

Fully Connected: 4096 to 1000

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

Correct label:
Cat

Softmax Loss

Fully Connected: 4096 to 4

Box Coordinates
(x, y, w, h)

L2 Loss

Correct box:
(x', y', w', h')

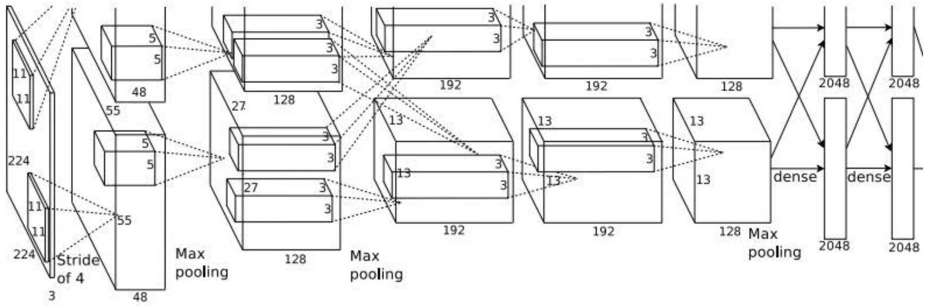
Multitask Loss

+ → **Loss**

Treat localization as a regression problem!

Object Detection as Classification: Sliding Window

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



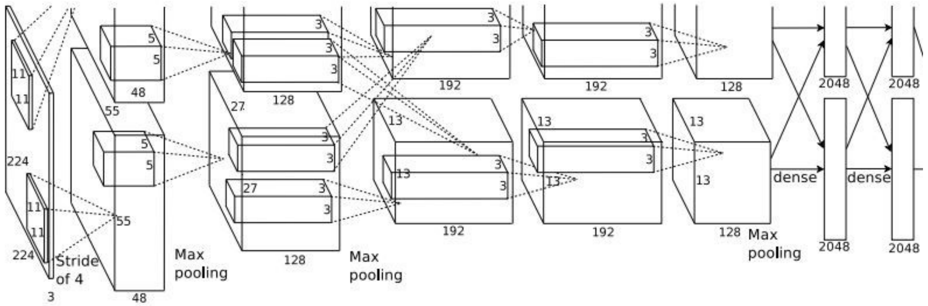
Dog? NO

Cat? NO

Background? YES

Object Detection as Classification: Sliding Window

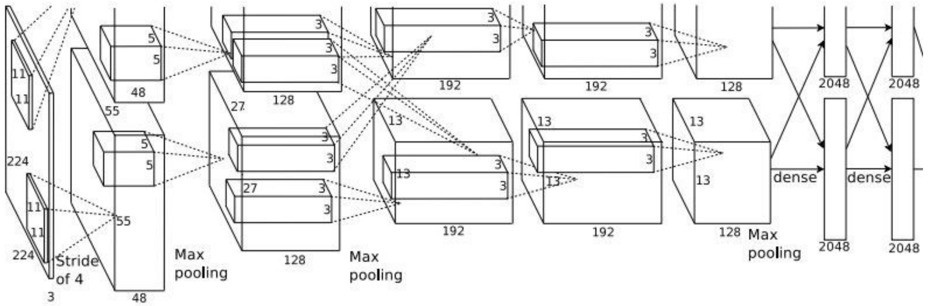
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

Object Detection as Classification: Sliding Window

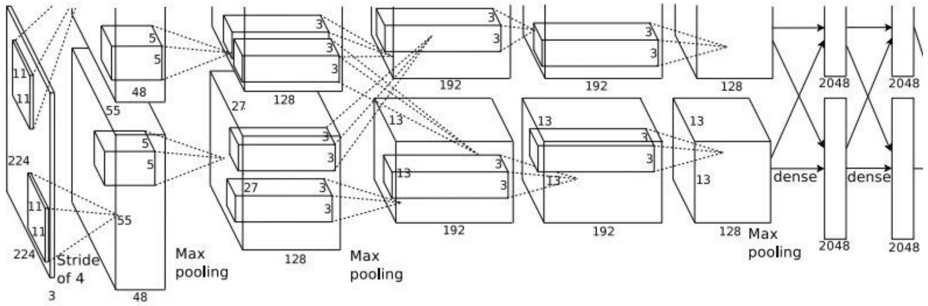
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

Object Detection as Classification: Sliding Window

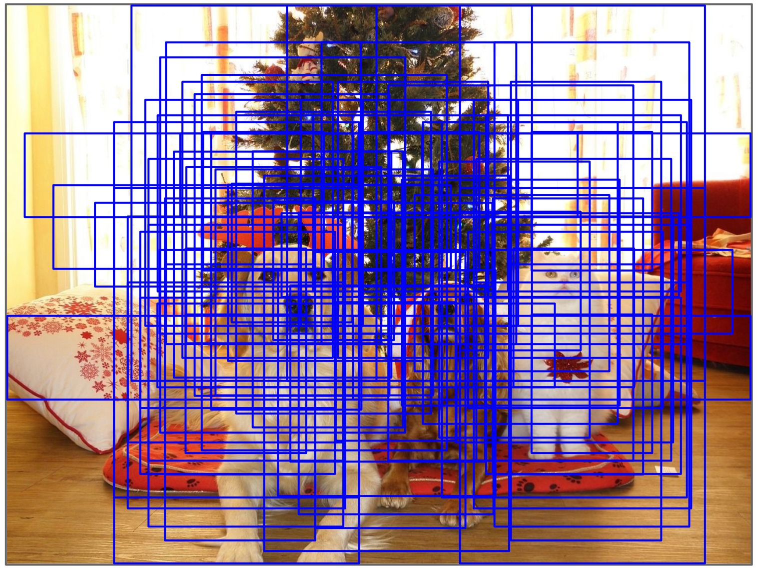
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

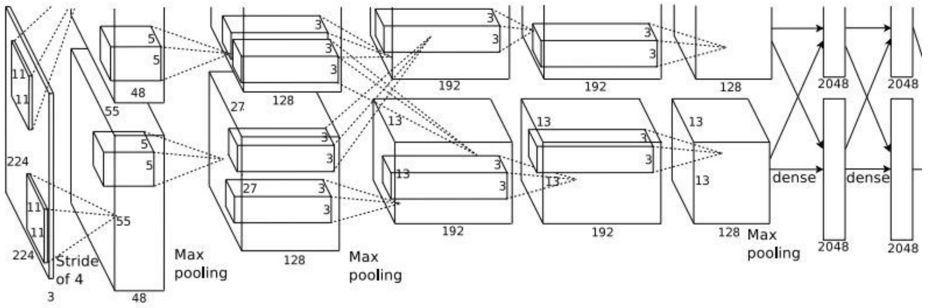


Dog? NO
Cat? YES
Background? NO

Object Detection as Classification: Sliding Window

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



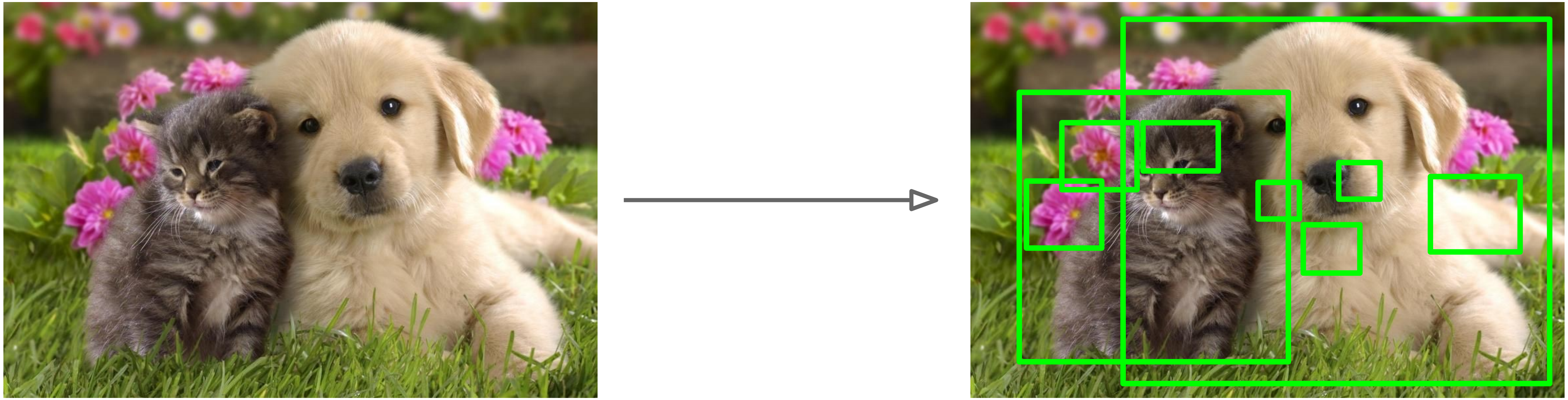


Dog? NO
Cat? YES
Background? NO

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

Region Proposals / Selective Search

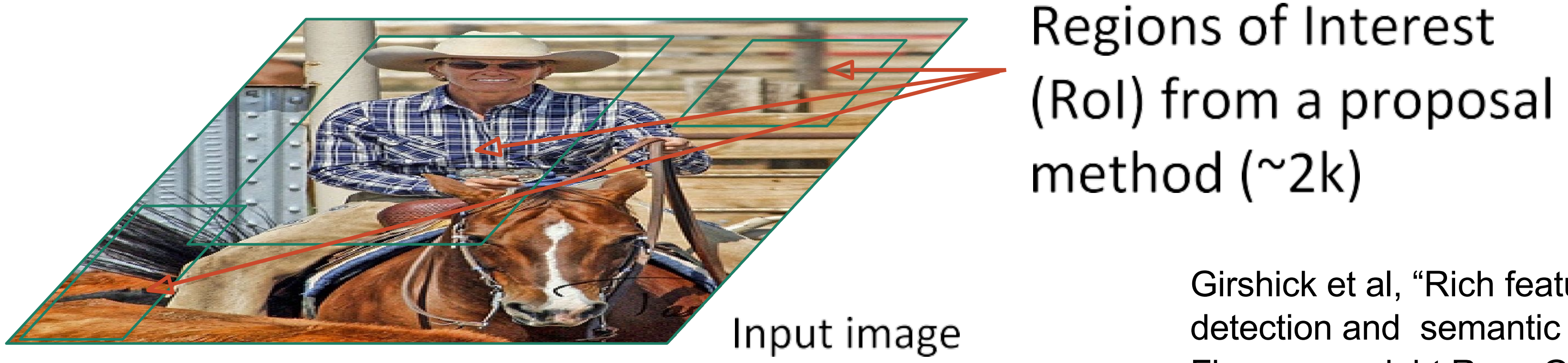
- Find “blobby” image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 2000 region proposals in a few seconds on CPU



Alexe et al, “Measuring the objectness of image windows”, TPAMI 2012
Uijlings et al, “Selective Search for Object Recognition”, IJCV 2013
Cheng et al, “BING: Binarized normed gradients for objectness estimation at 300fps”, CVPR 2014
Zitnick and Dollar, “Edge boxes: Locating object proposals from edges”, ECCV 2014

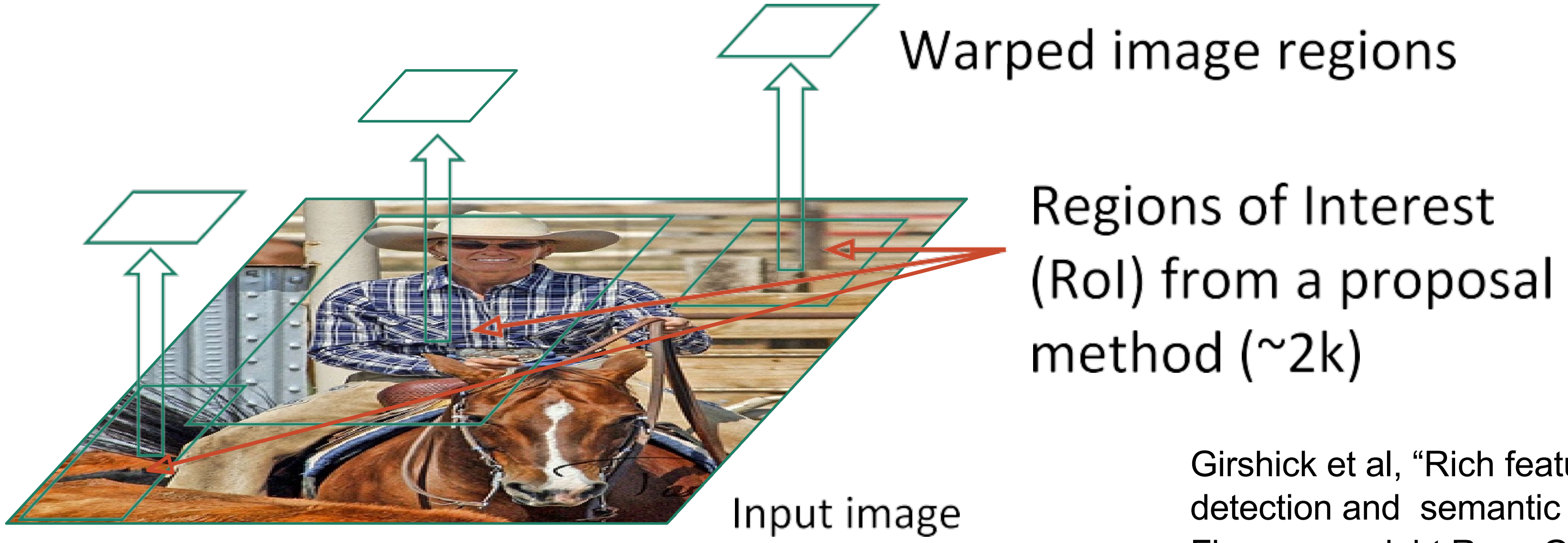
Regions from selective search

R-CNN



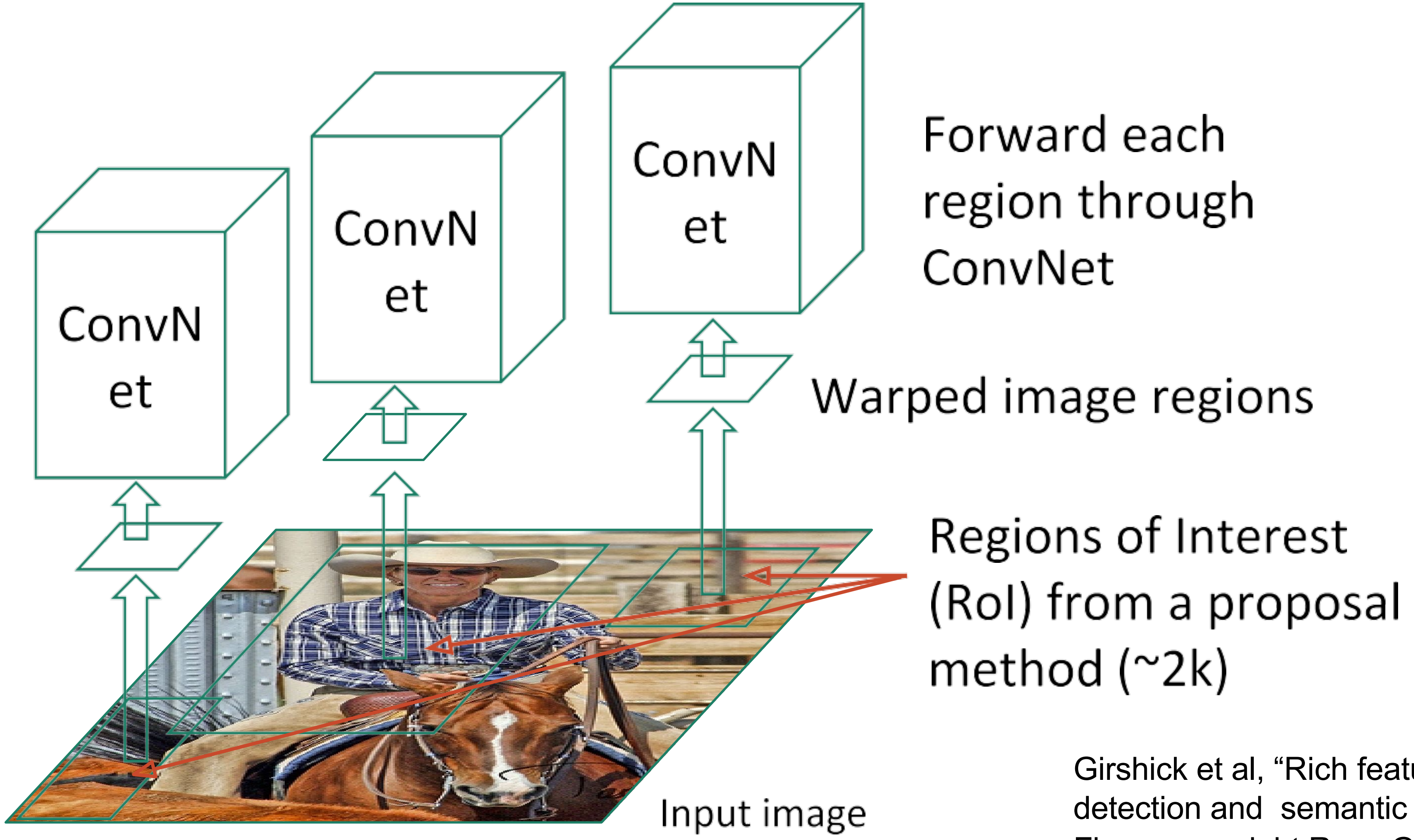
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.

R-CNN



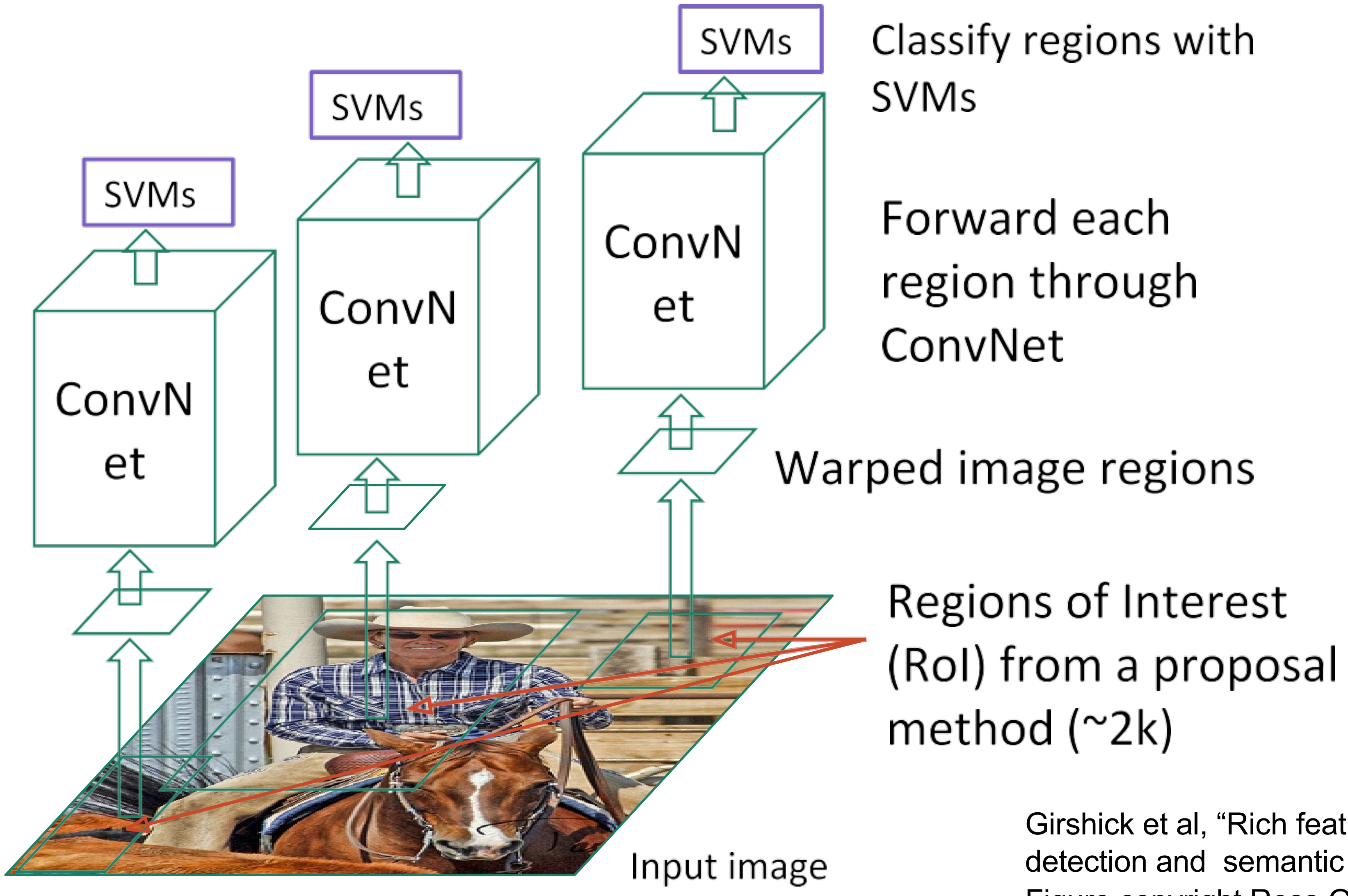
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.

R-CNN



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

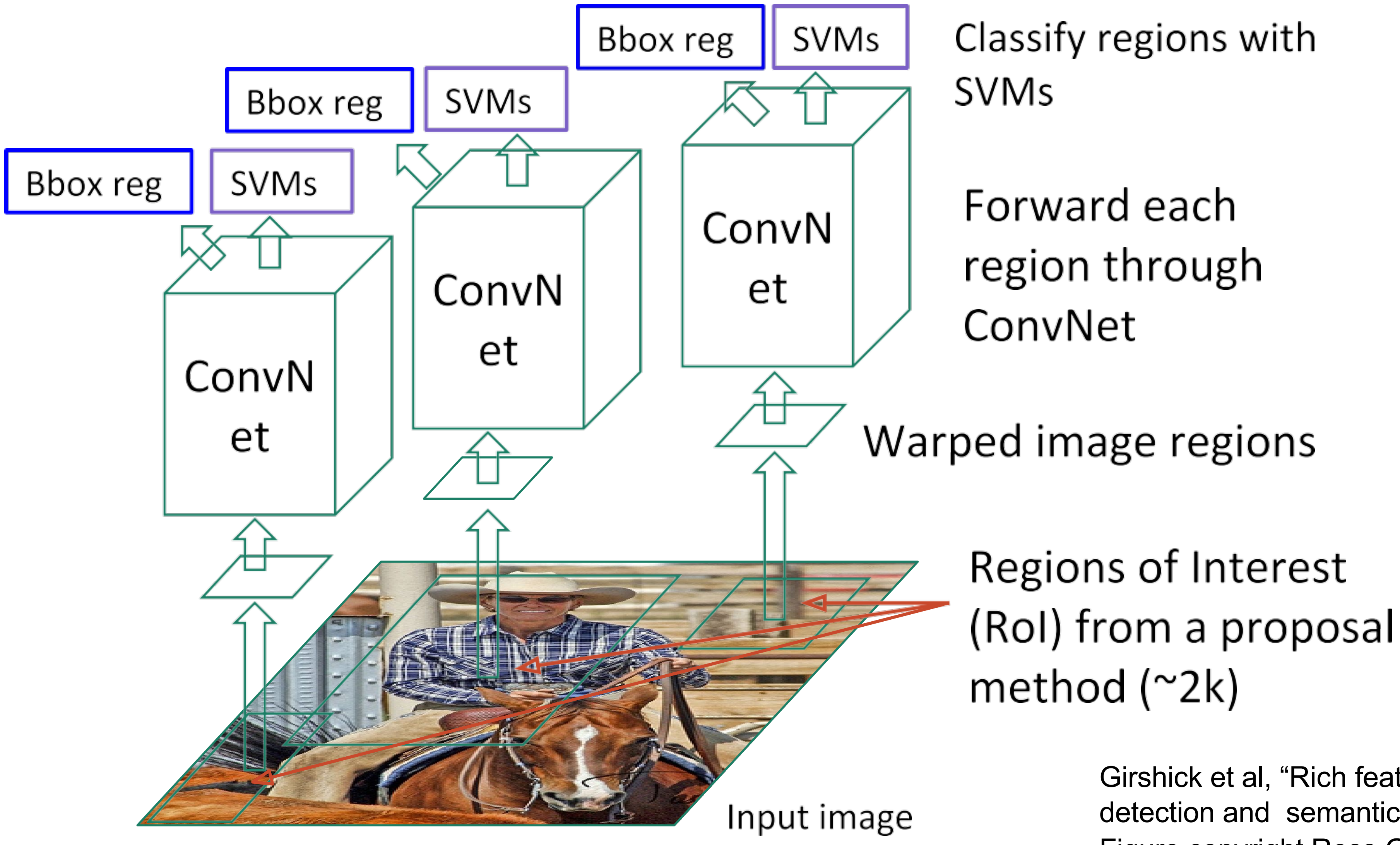
R-CNN



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

Linear Regression for bounding box offsets



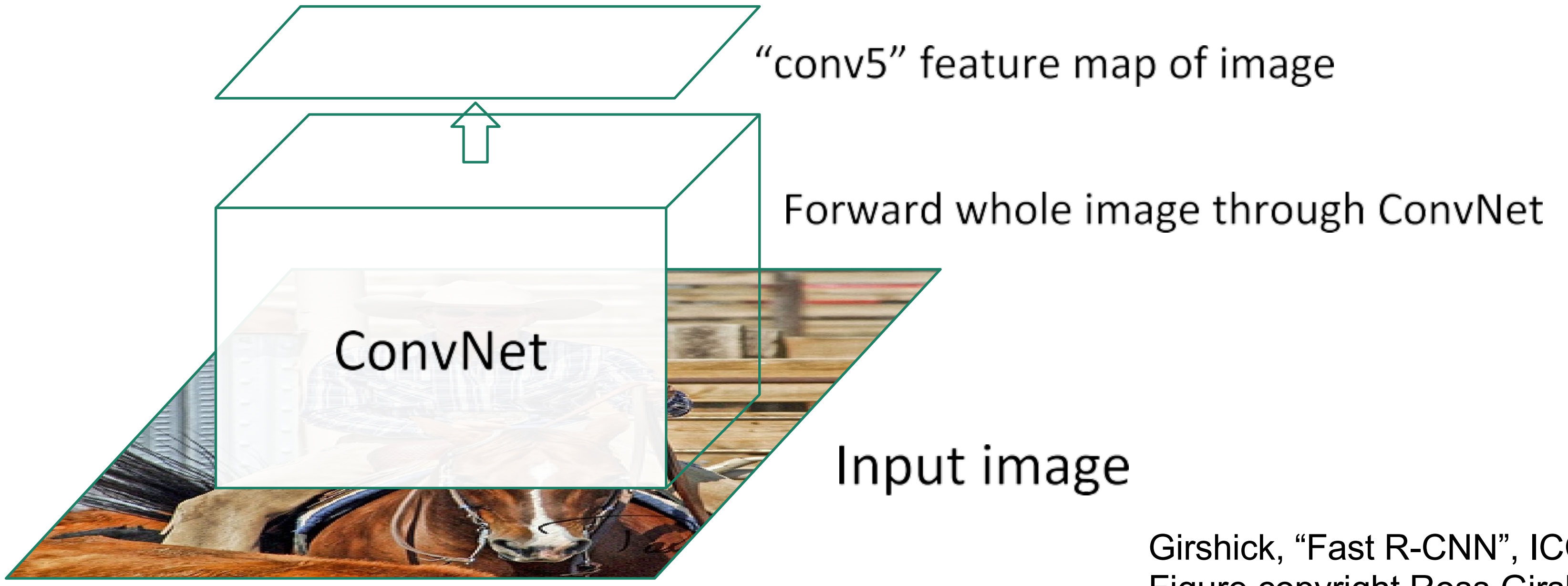
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Fast R-CNN

Input image

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

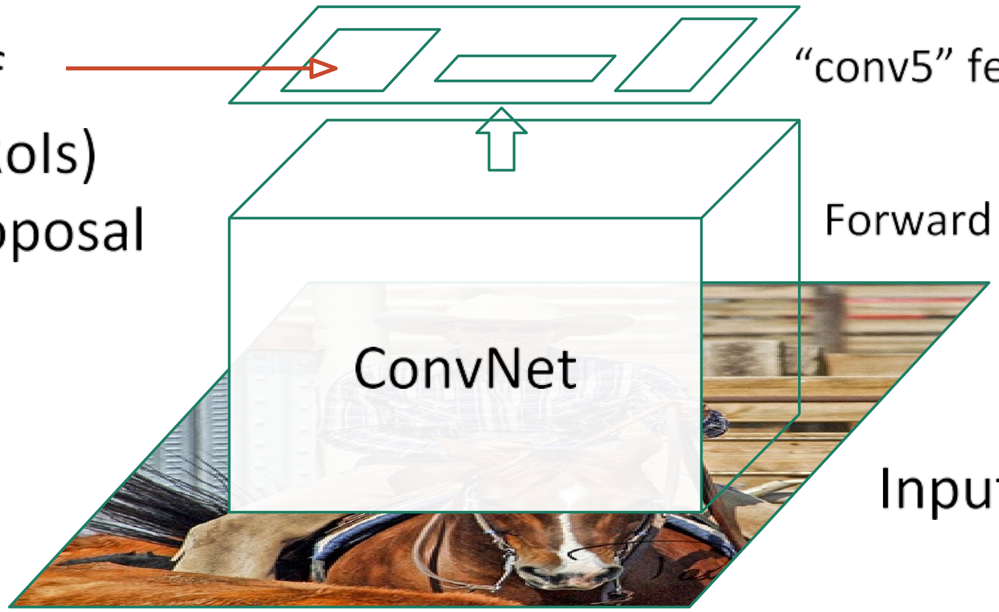
Fast R-CNN



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

Fast R-CNN

Regions of Interest (RoIs) from a proposal method



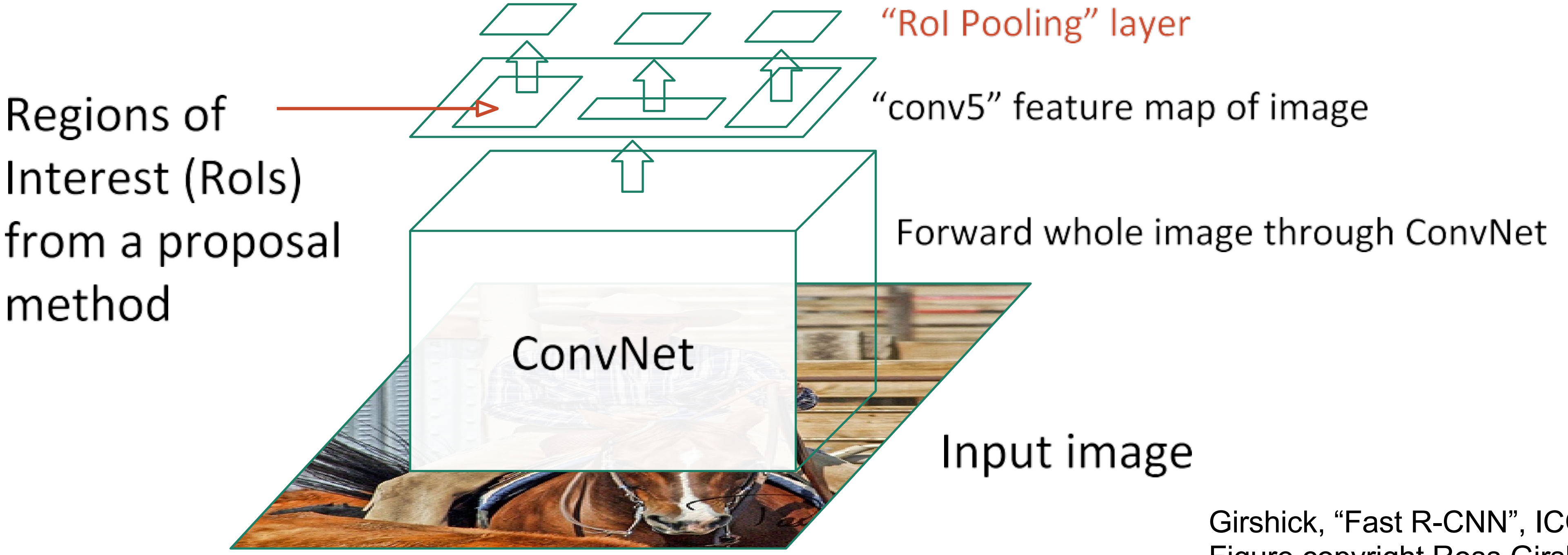
"conv5" feature map of image

Forward whole image through ConvNet

Input image

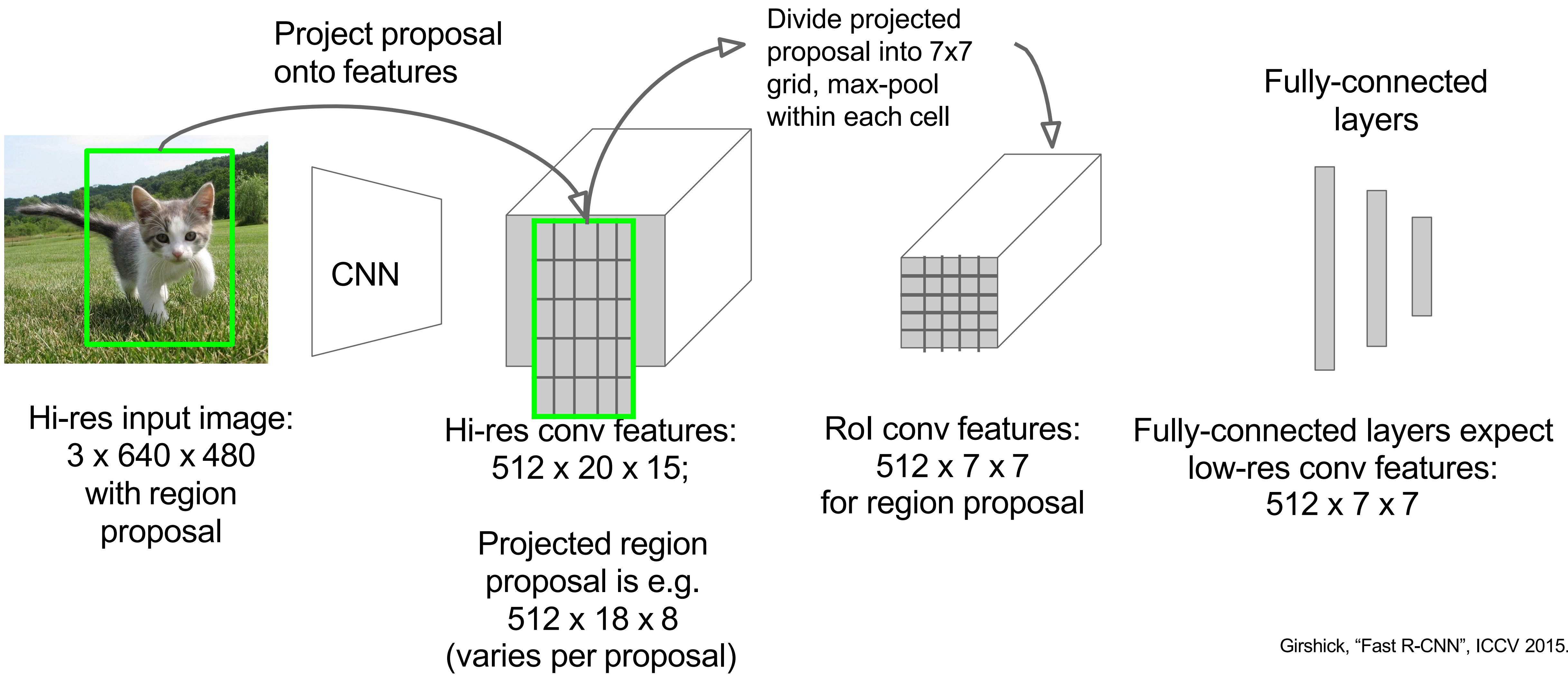
Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Fast R-CNN



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

Fast R-CNN: RoI Pooling



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

ROI pooling

0.88	0.44	0.14	0.16	0.37	0.77	0.96	0.27
0.19	0.45	0.57	0.16	0.63	0.29	0.71	0.70
0.66	0.26	0.82	0.64	0.54	0.73	0.59	0.26
0.85	0.34	0.76	0.84	0.29	0.75	0.62	0.25
0.32	0.74	0.21	0.39	0.34	0.03	0.33	0.48
0.20	0.14	0.16	0.13	0.73	0.65	0.96	0.32
0.19	0.69	0.09	0.86	0.88	0.07	0.01	0.48
0.83	0.24	0.97	0.04	0.24	0.35	0.50	0.91

1. 8x8 conv feature map

0.88	0.44	0.14	0.16	0.37	0.77	0.96	0.27
0.19	0.45	0.57	0.16	0.63	0.29	0.71	0.70
0.66	0.26	0.82	0.64	0.54	0.73	0.59	0.26
0.85	0.34	0.76	0.84	0.29	0.75	0.62	0.25
0.32	0.74	0.21	0.39	0.34	0.03	0.33	0.48
0.20	0.14	0.16	0.13	0.73	0.65	0.96	0.32
0.19	0.69	0.09	0.86	0.88	0.07	0.01	0.48
0.83	0.24	0.97	0.04	0.24	0.35	0.50	0.91

2. Region of Interest (ROI)

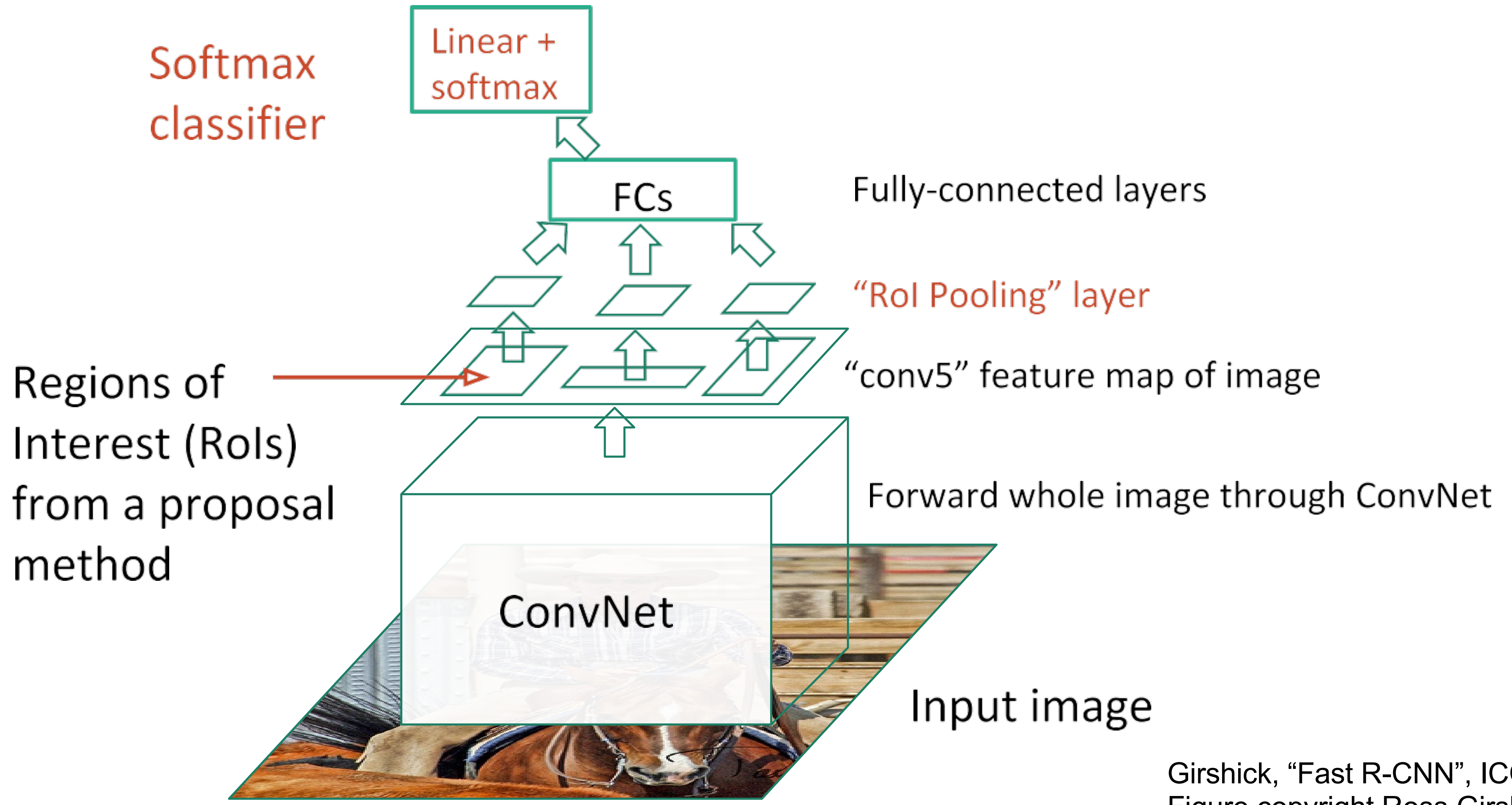
0.88	0.44	0.14	0.16	0.37	0.77	0.96	0.27
0.19	0.45	0.57	0.16	0.63	0.29	0.71	0.70
0.66	0.26	0.82	0.64	0.54	0.73	0.59	0.26
0.85	0.34	0.76	0.84	0.29	0.75	0.62	0.25
0.32	0.74	0.21	0.39	0.34	0.03	0.33	0.48
0.20	0.14	0.16	0.13	0.73	0.65	0.96	0.32
0.19	0.69	0.09	0.86	0.88	0.07	0.01	0.48
0.83	0.24	0.97	0.04	0.24	0.35	0.50	0.91

3. 2x2 intended pooling output

0.85	0.84
0.97	0.96

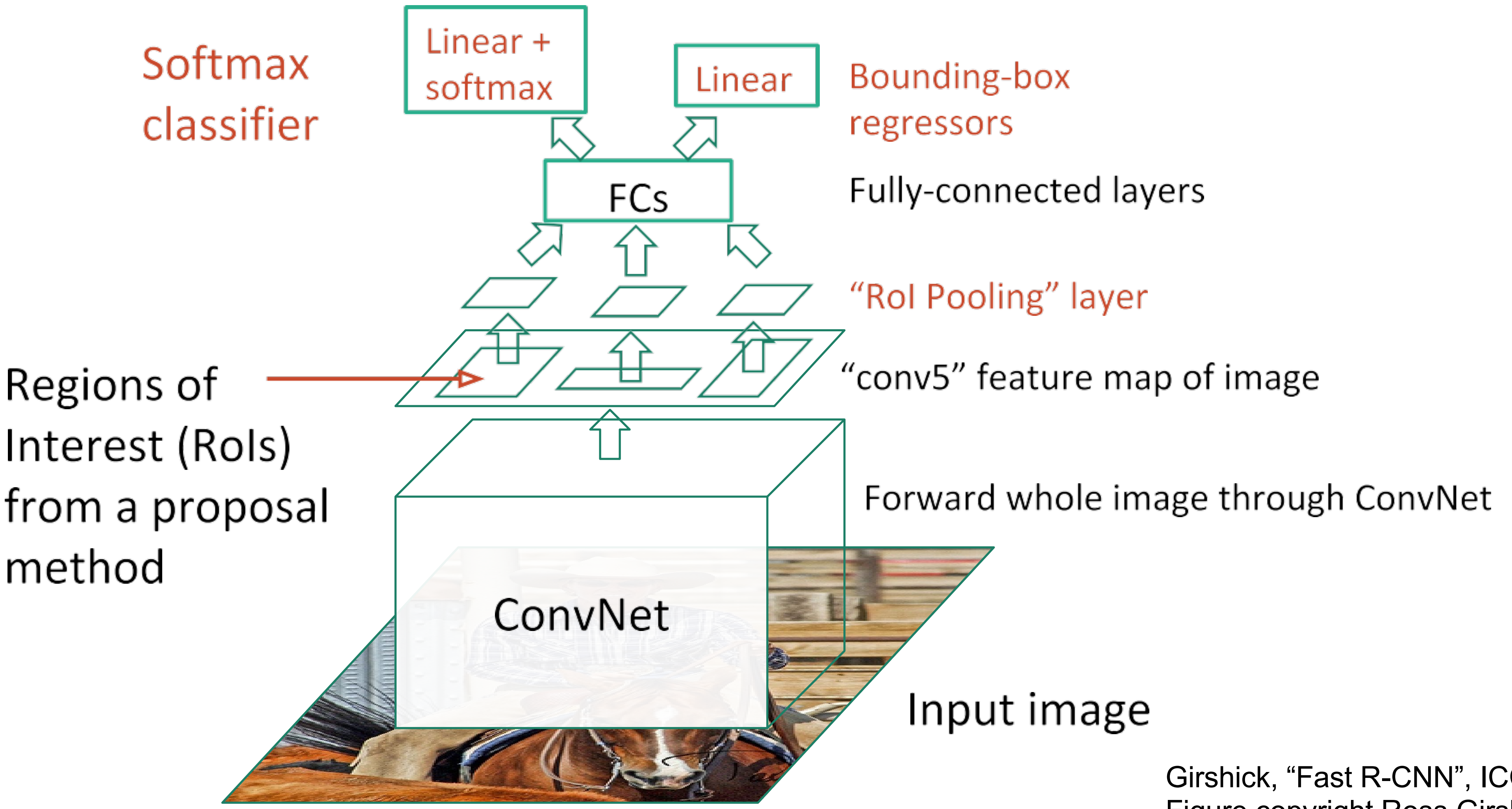
4. Output

Fast R-CNN



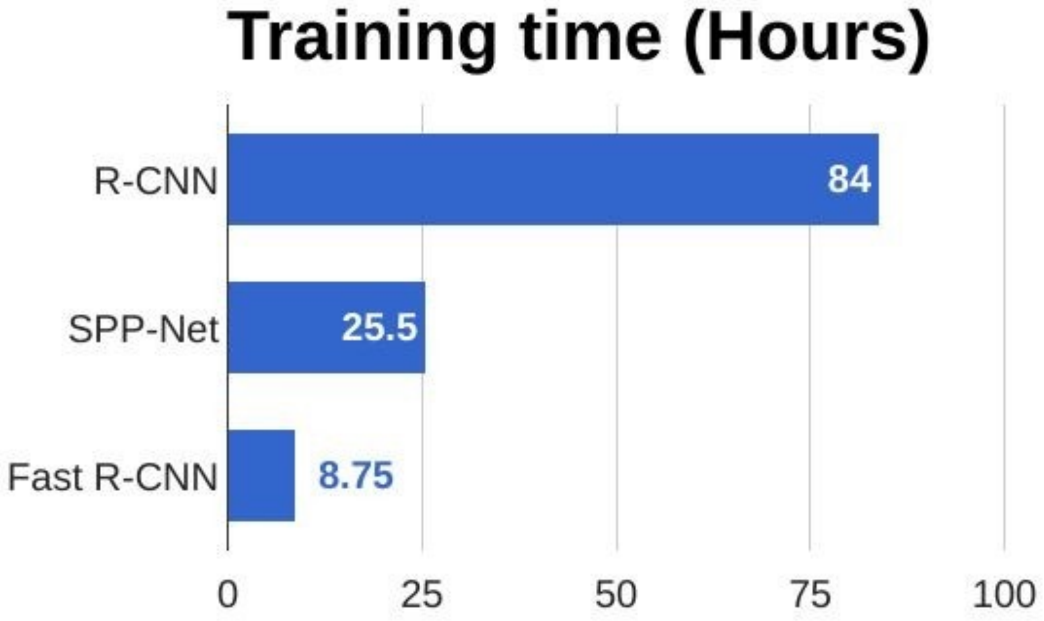
Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source.
Reproduced with permission.

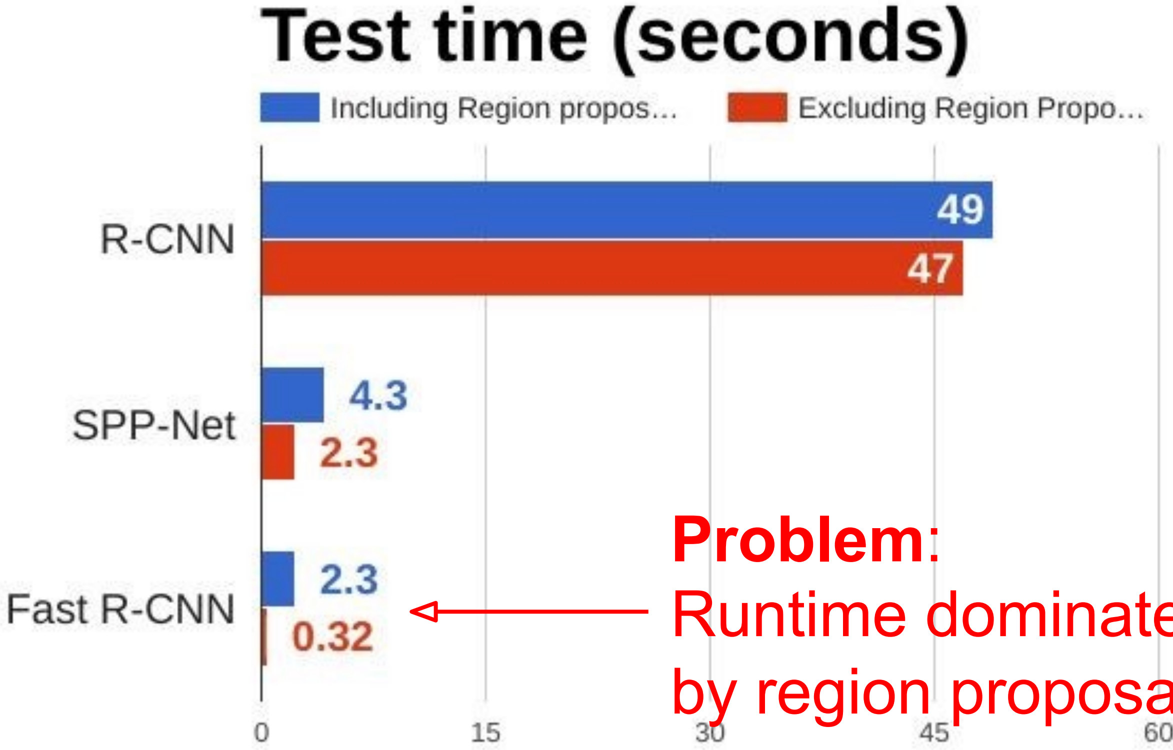
Fast R-CNN



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

R-CNN vs SPP vs Fast R-CNN





Problem:
Runtime dominated by region proposals!

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

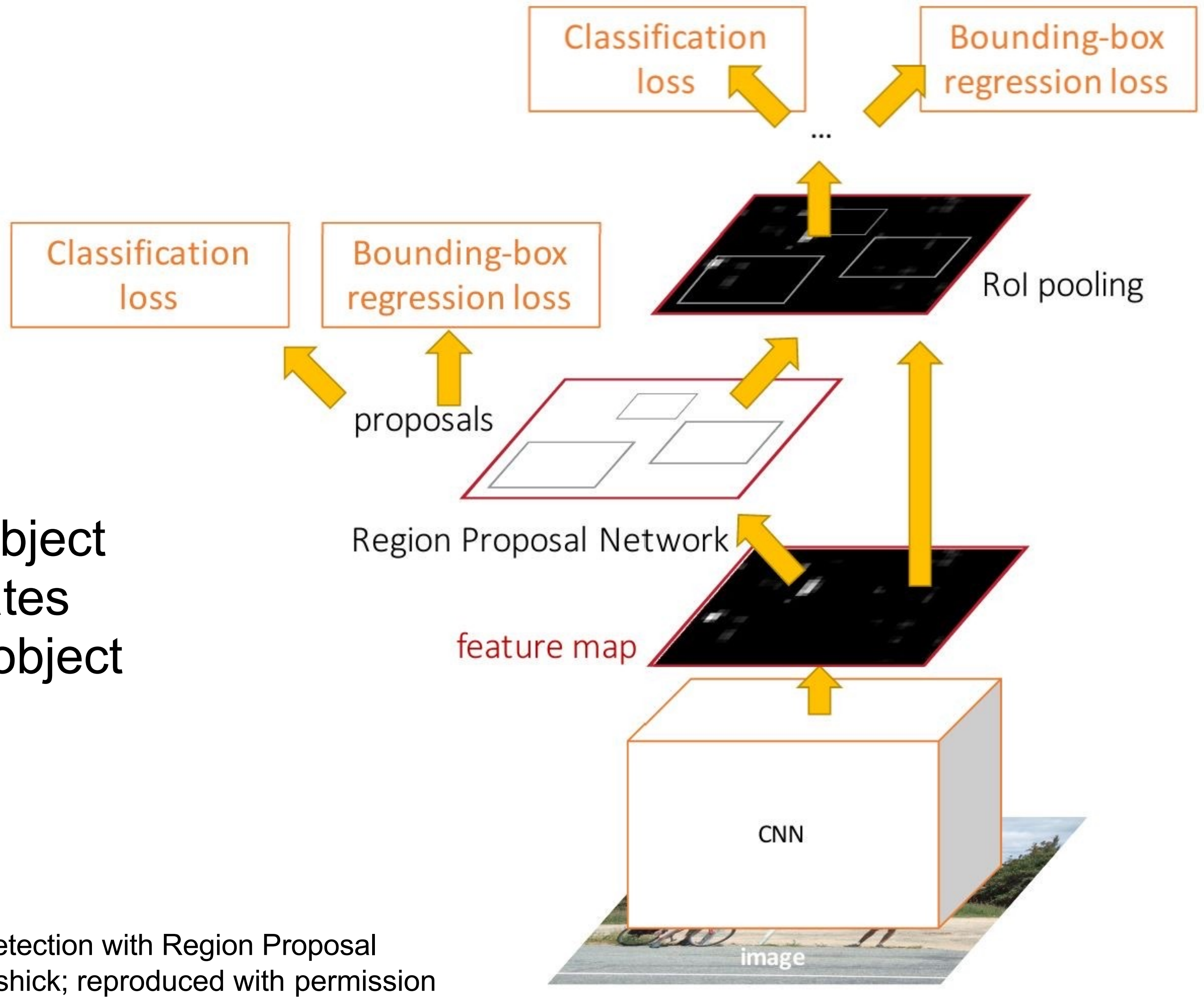
Faster R-CNN:

Make CNN do proposals!

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

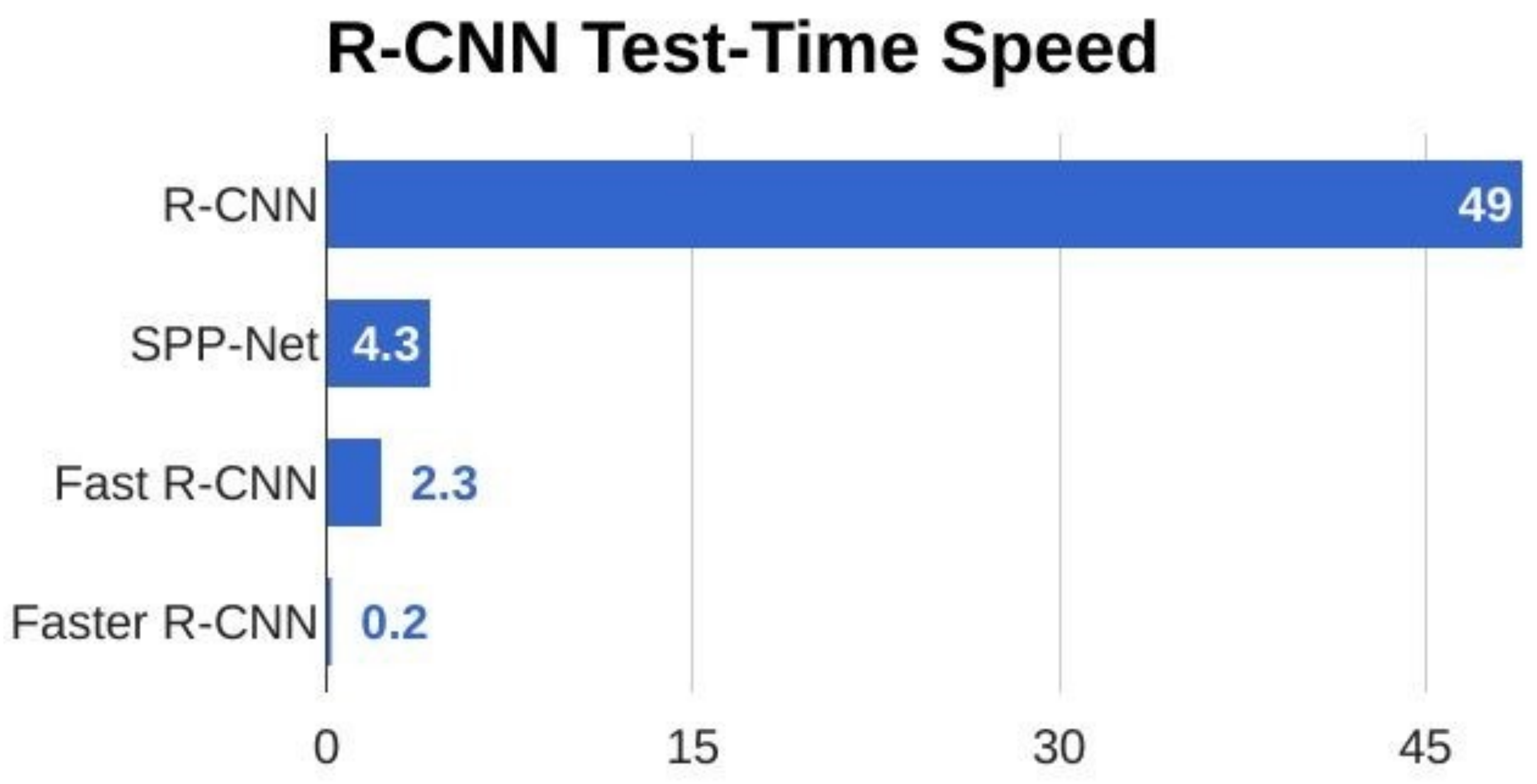
Jointly train with 4 losses:

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



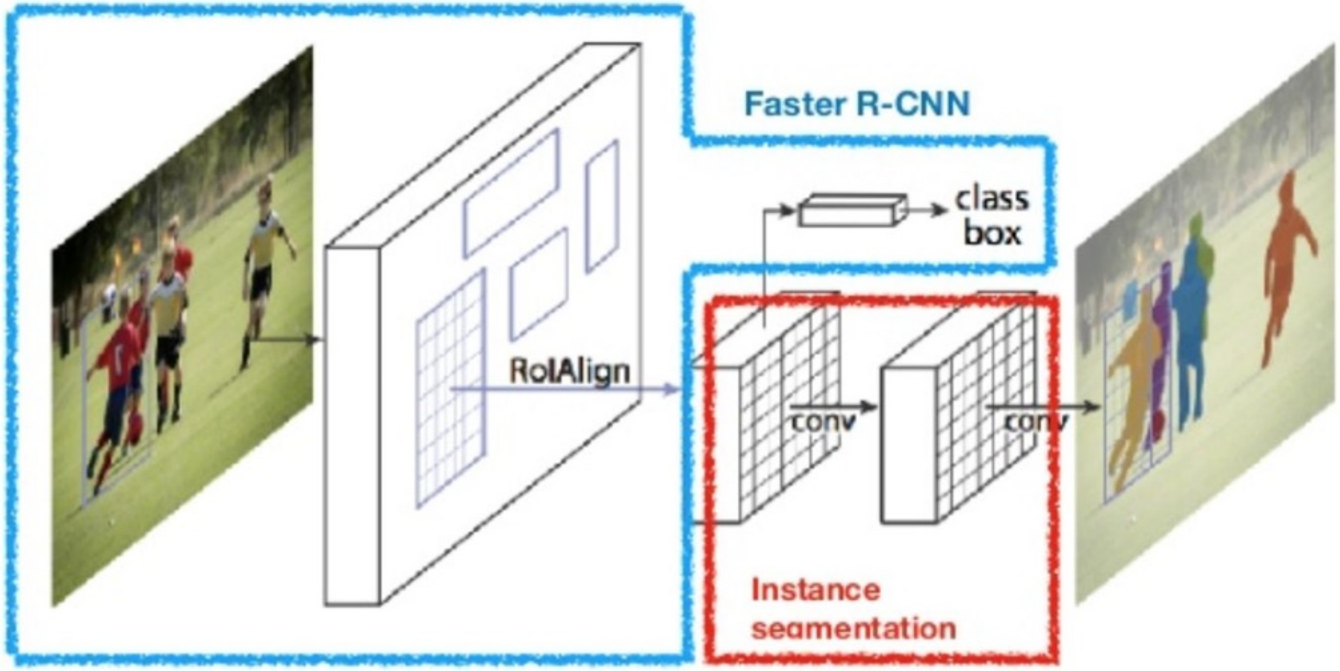
Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015 Figure copyright 2015, Ross Girshick; reproduced with permission

Faster R-CNN: Make CNN do proposals!



Instance Segmentation

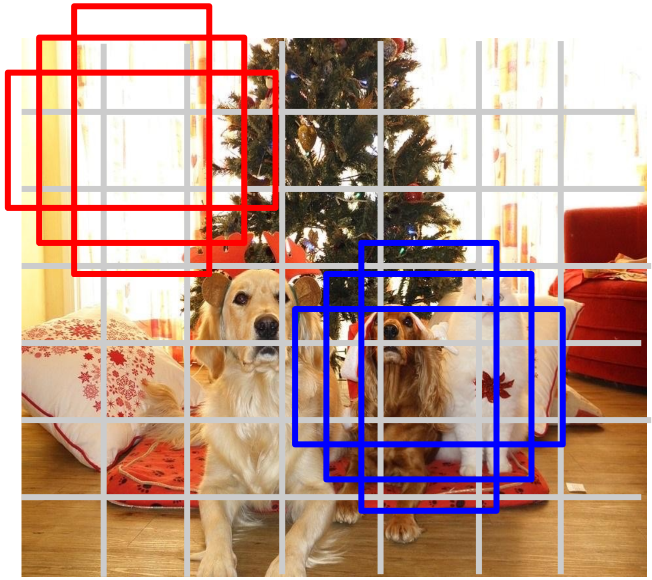
- Mask R-CNN



Detection without Proposals: YOLO / SSD

Go from input image to tensor of scores with one big convolutional network!

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)

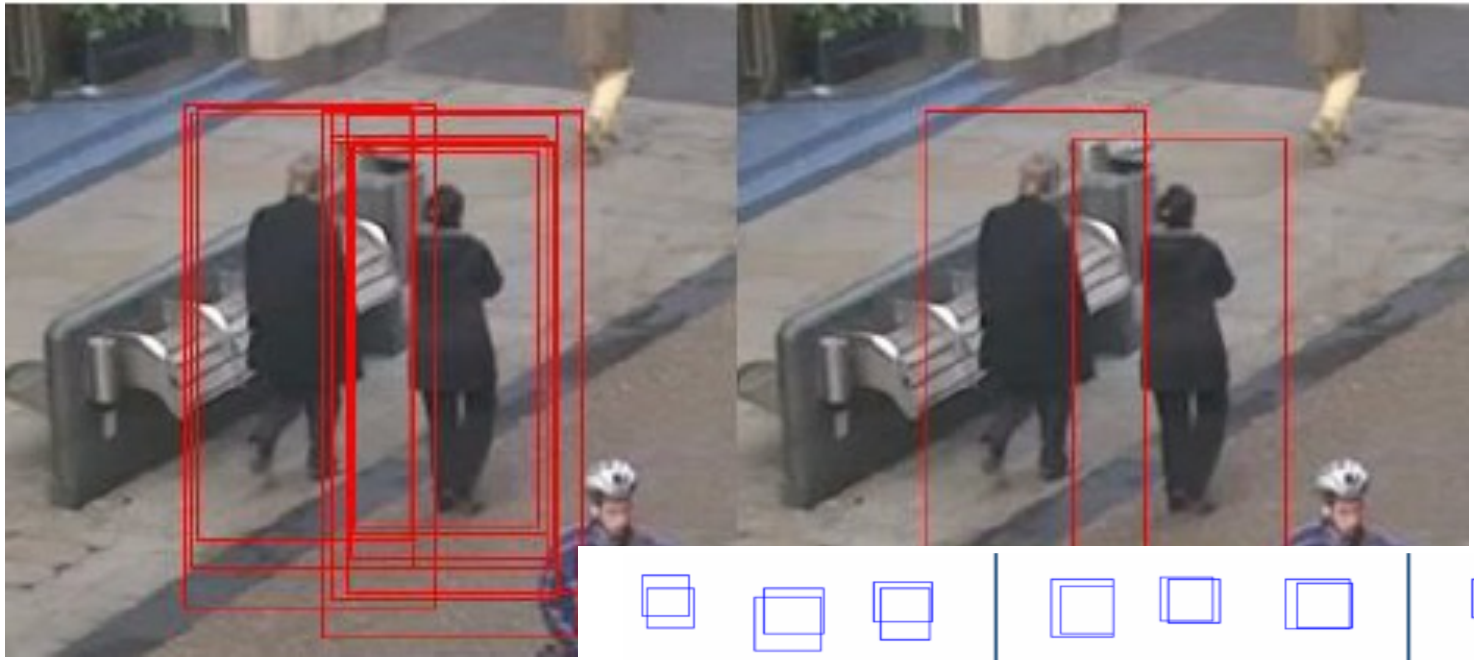
Output:
7 x 7 x (5 * B + C)

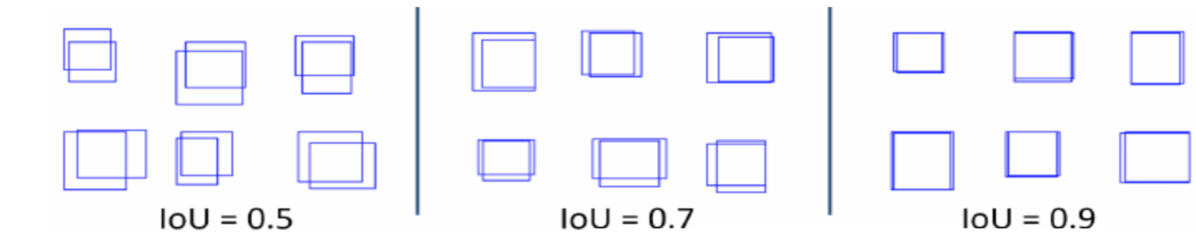
Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016

Many detections above threshold.

Non-maximum suppression
(to improve precision)

Non-maximum suppression (to improve precision)





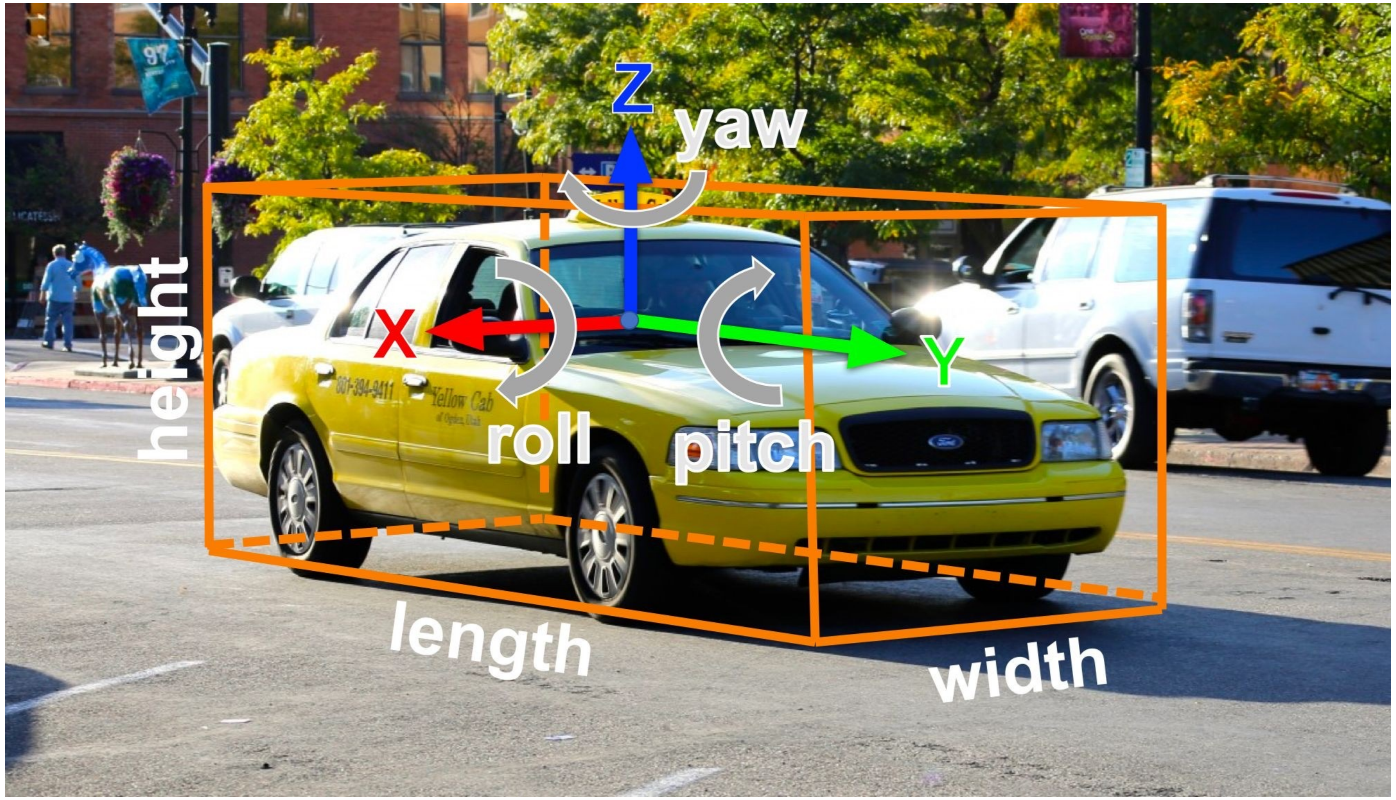
$$IoU(A,B) = \frac{A \cap B}{A \cup B}$$

Sample heuristics:

- Remove all boxes, if confidence < T1
- For each object class, pick box with highest confidence, remove all boxes with IoU > T2, stop when all boxes are considered.

Part III: 3D Segmentation & Detection

3D Object Detection



2D Object Detection:
2D bounding box
(x, y, w, h)

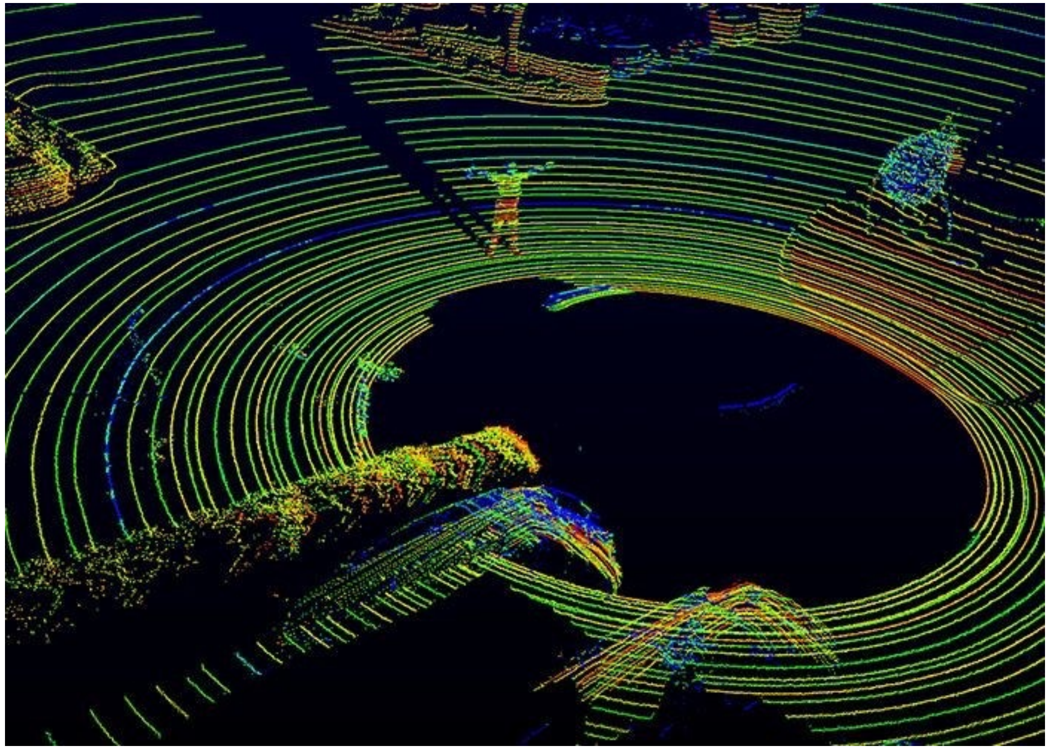
3D Object Detection:
3D oriented bounding box
(x, y, z, w, h, l, r, p, y)

Simplified bbox: no roll & pitch

Much harder problem than 2D object detection!

LiDAR

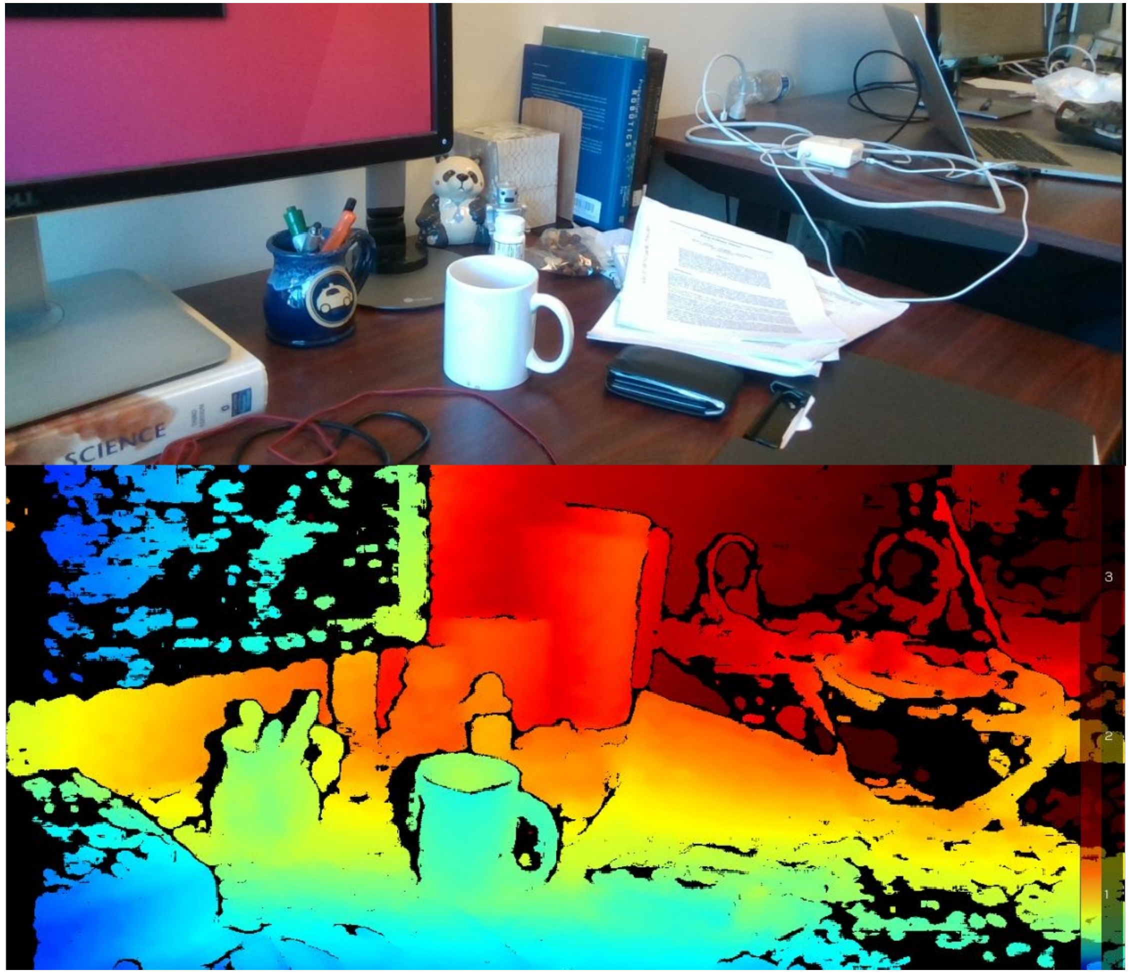
Velodyne (HDL-64e)



3D Point Cloud

RGB-Depth Camera

Kinect (Xbox One)



RGB

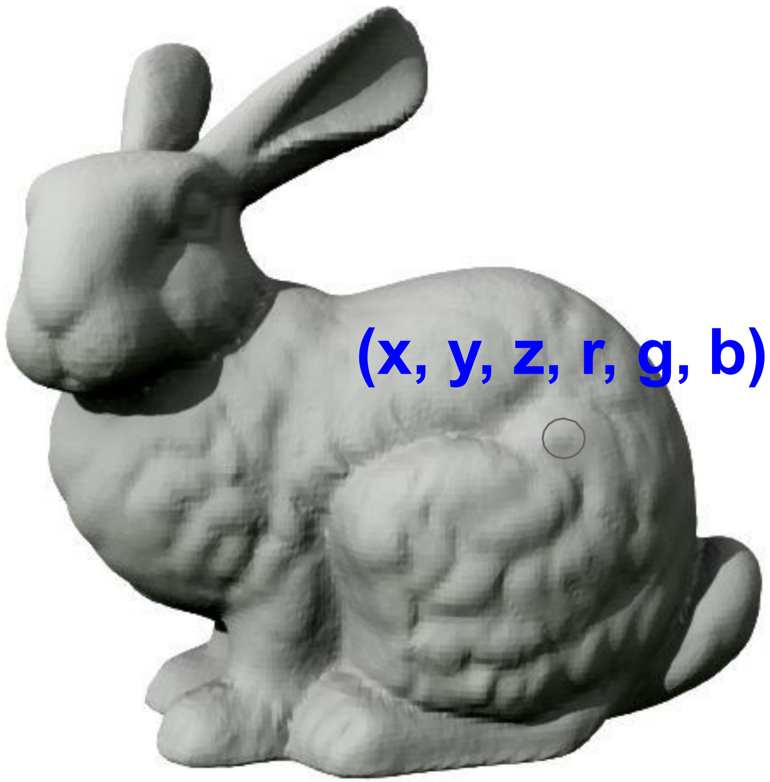
Depth

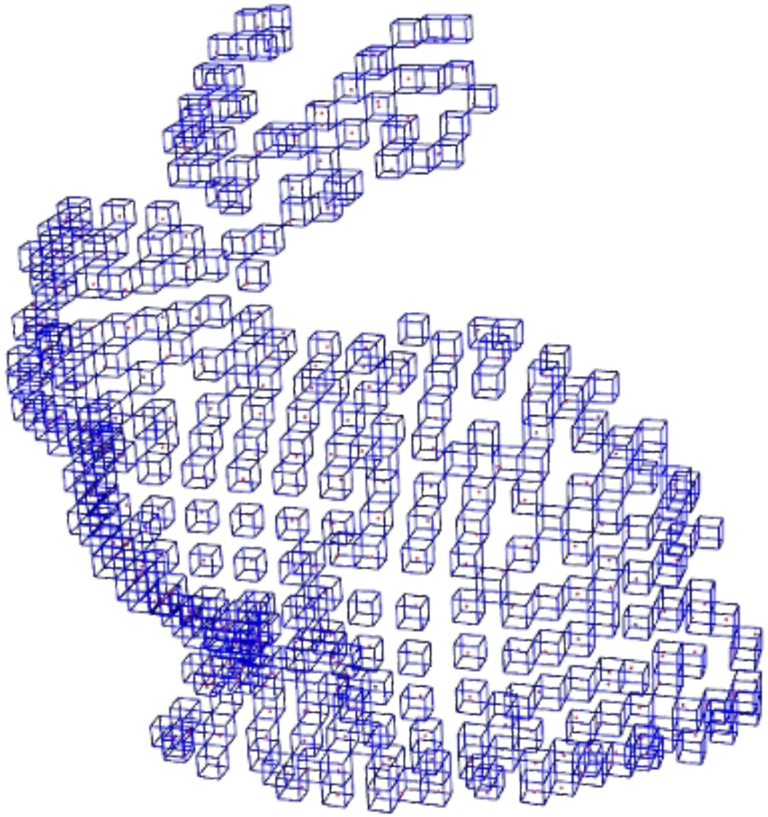
RGB-Depth Camera

Registered RGB + depth point cloud

How to feed point cloud to neural networks?

Point Cloud Voxelization





1. Capture RGB-D point cloud of a scene.

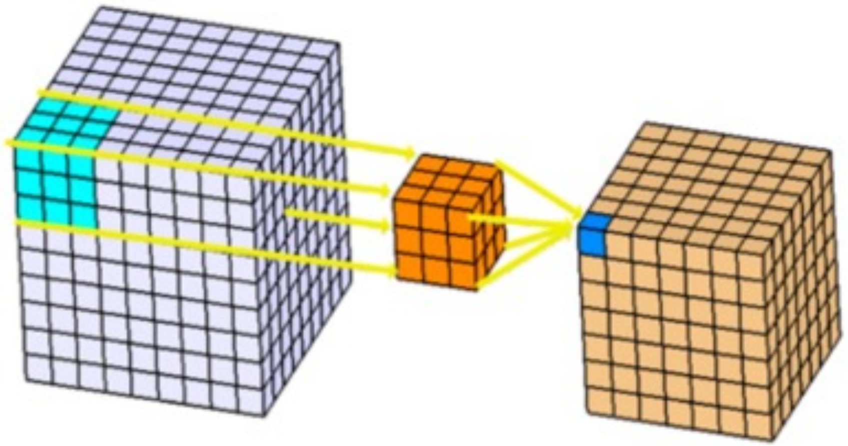
2. Partition the 3D space into a regular 3D grid.

3. For each grid cell that has a point fall into it, fill the cell with the RGB value of that point.

A bit like "3D image"

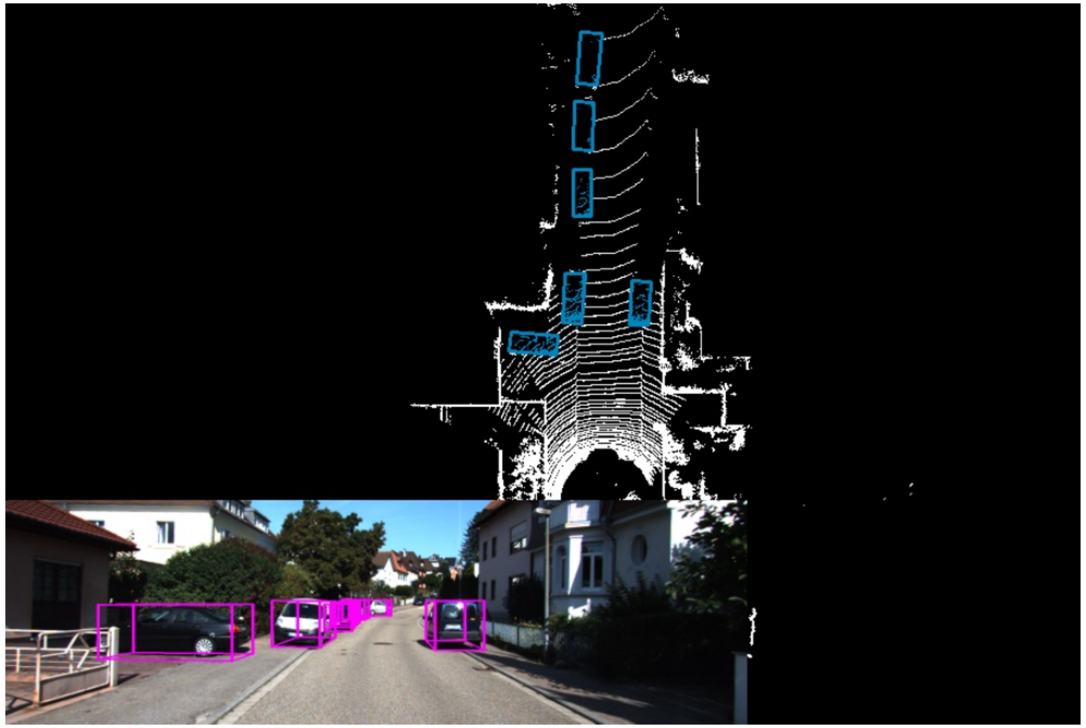
3D Convolution

3-dimensional filter that moves 3-directions (x,y,z)



https://www.kaggle.com/shivamb/3d-convolutions-understanding-and-implementation

2D views



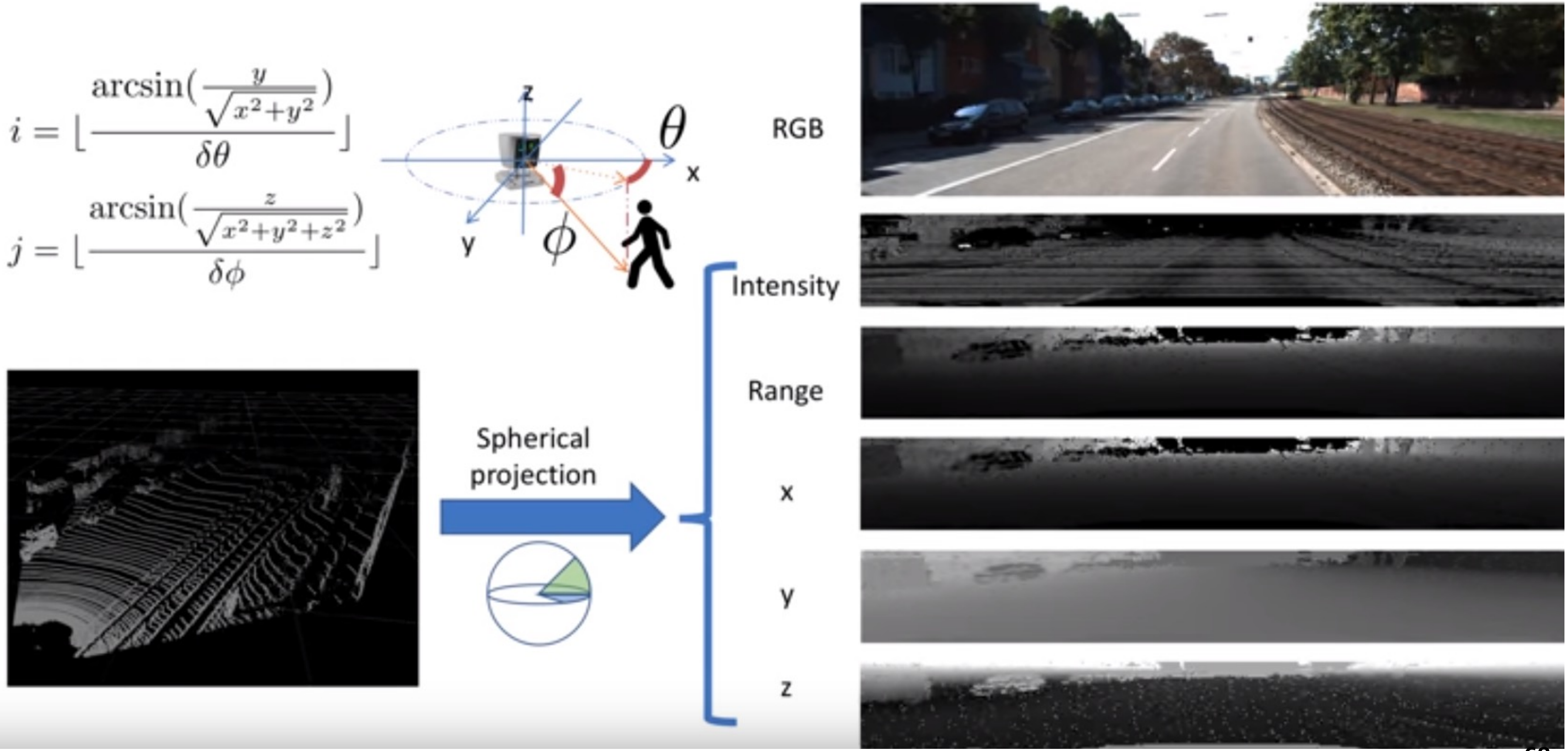
Bird's Eye View

Front View

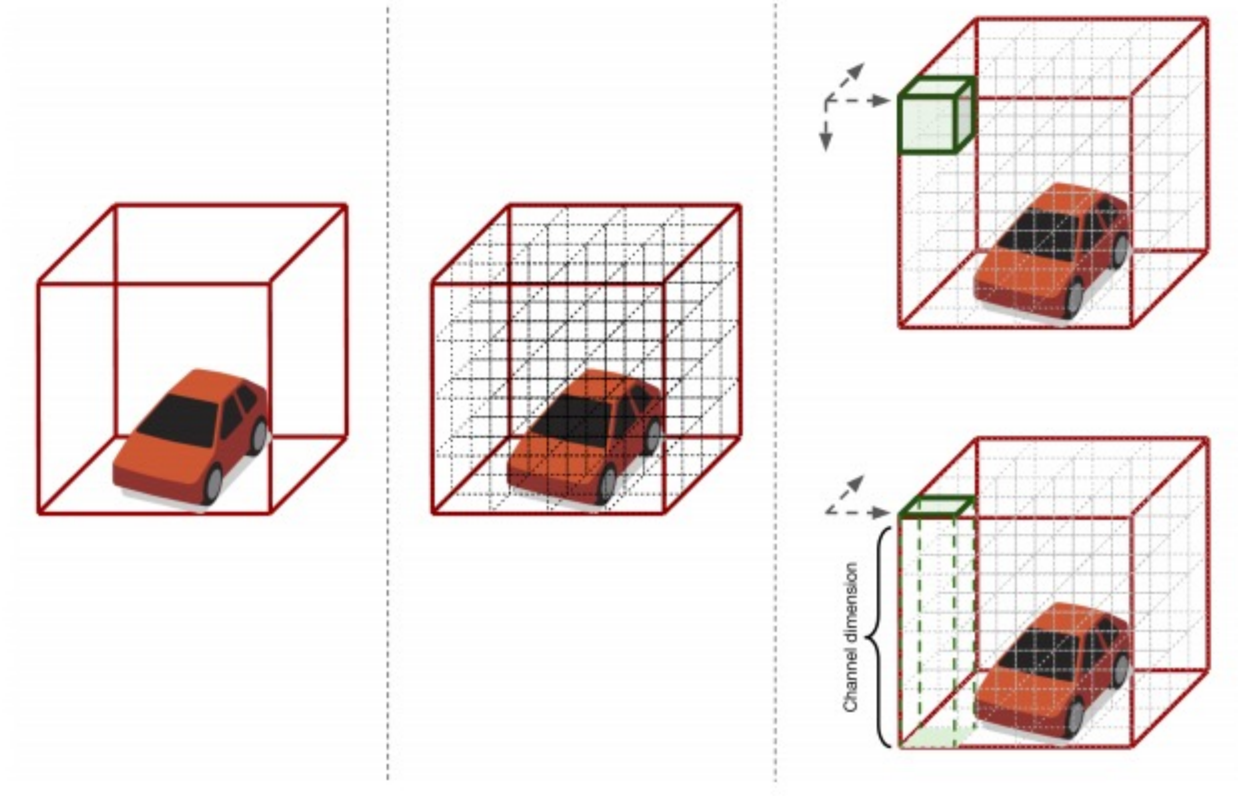
2D Convolution on Front View

$$i = \left\lfloor \frac{\arcsin\left(\frac{y}{\sqrt{x^2+y^2}}\right)}{\delta\theta} \right\rfloor$$

$$j = \left\lfloor \frac{\arcsin\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right)}{\delta\phi} \right\rfloor$$

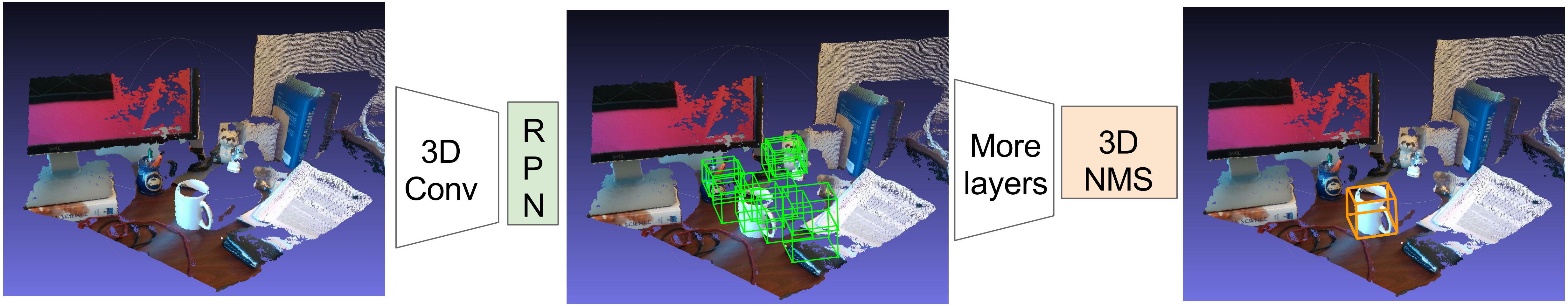


2D Convolution on Bird's Eye View



Zhang, Chris, Wenjie Luo, and Raquel Urtasun. "Efficient Convolutions for Real-Time Semantic Segmentation of 3D Point Clouds." *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018.

3D Object Detection: RGB-Depth Camera



Voxelized RGB-D point cloud

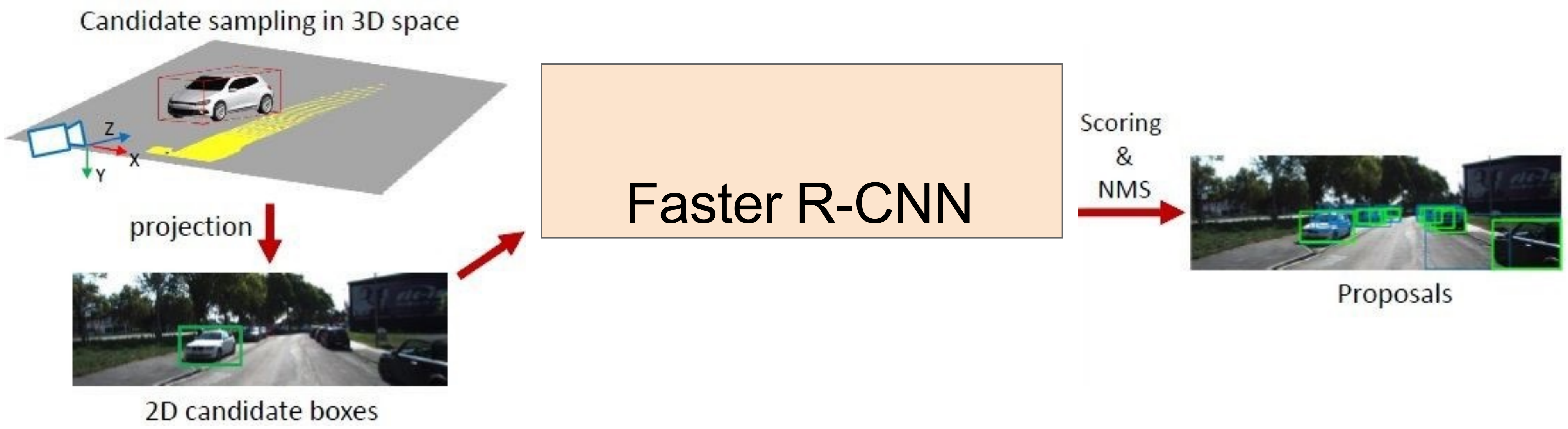
3D region proposals

Object categories +
3D bounding boxes

"Faster RCNN in 3D"

S. Song, and J. Xiao. Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images. CVPR 2016

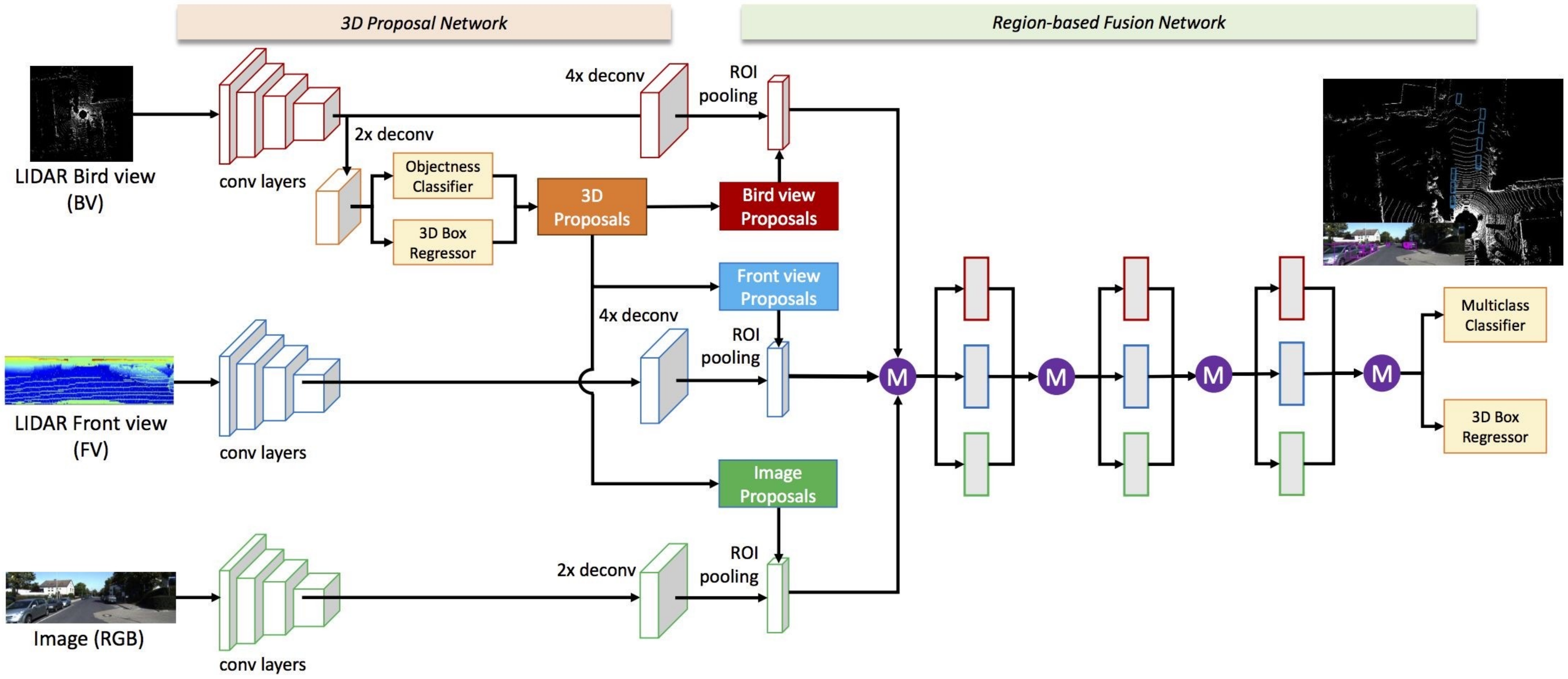
3D Object Detection: Monocular Camera



- 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.

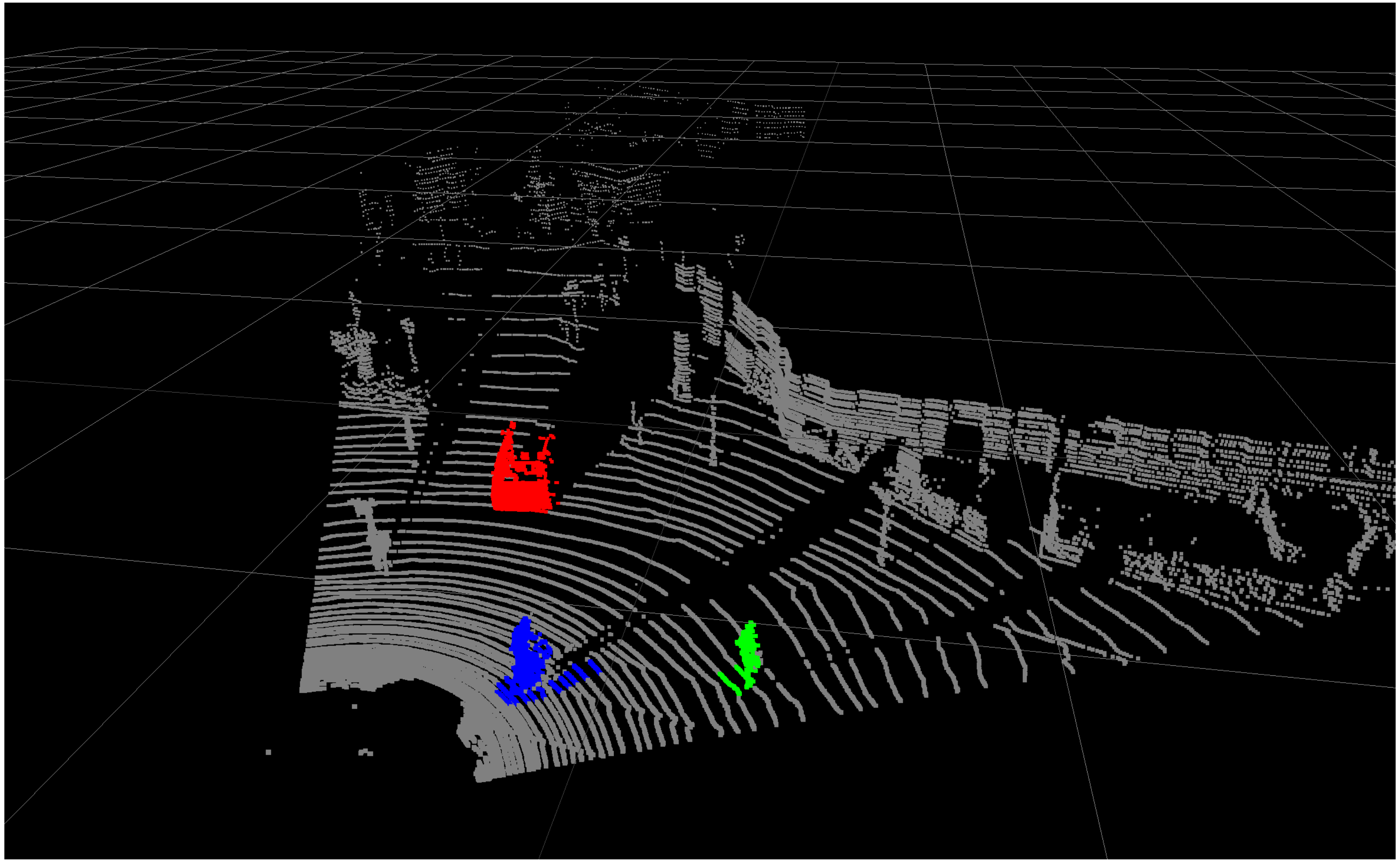
3D Object Detection: Camera + LiDAR



- Combine 3D proposals from multiple views & sensors
- regress 3D box parameters + class score

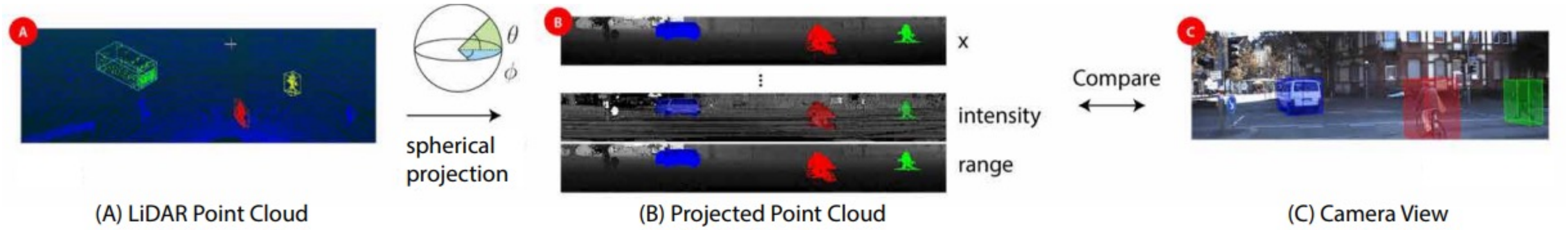
Chen, Xiaozhi, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. "Multi-view 3d object detection network for autonomous driving." *CVPR* 2017

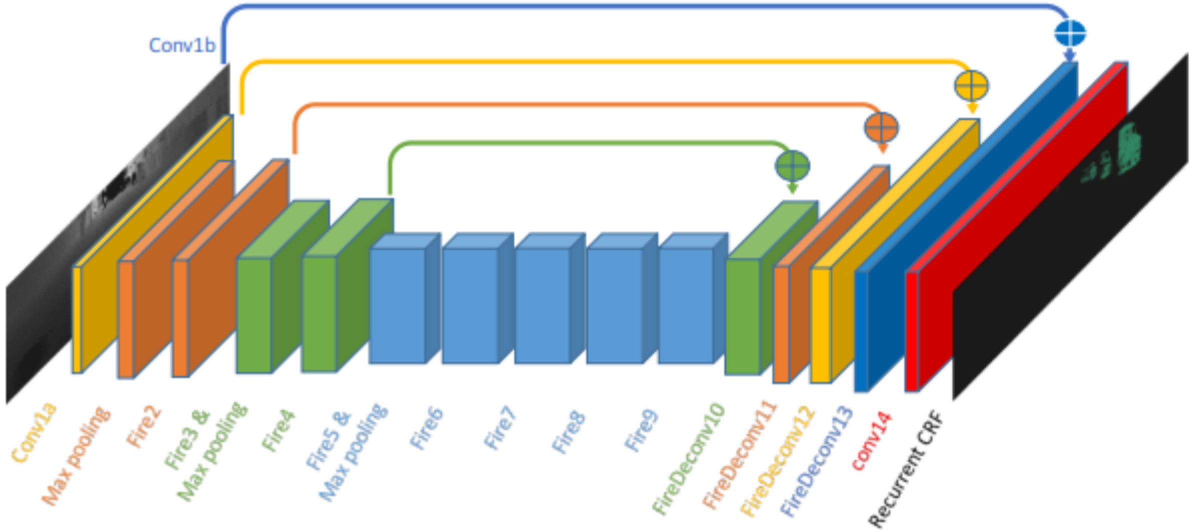
3D Segmentation



Wu, Bichen, et al. "Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud." *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018.

3D Segmentation





Wu, Bichen, et al. "Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud." *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018.

3D Segmentation and Detection

- ❑ 3D convolution
- ❑ 2D convolution on projected view
- ❑ Neural network for irregular data
  - ❑ Point cloud
  - ❑ Graph neural networks

Qi, Charles R., et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation." *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE* 1.2 (2017)

Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." *arXiv preprint arXiv:1609.02907* (2016).