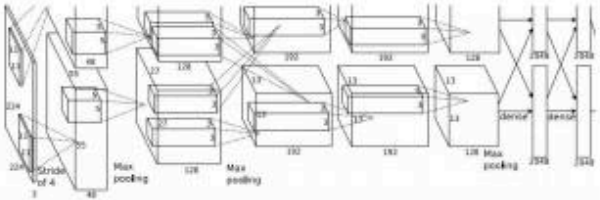
Segmentation and Detection

By: Ismail Elezi
ismail.elezi@gmail.com

Image Classification - review



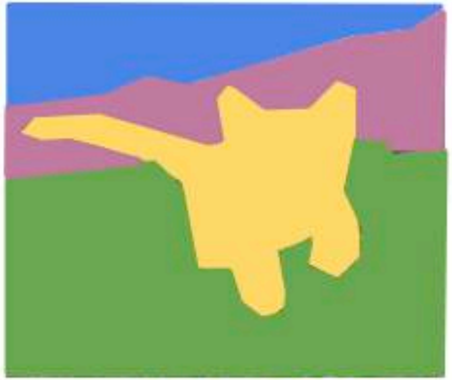
Vector: 4096

Fully-Connected: 4096 to 1000

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

Other Computer Vision Tasks

Semantic Segmentation



GRASS, CAT, TREE, SKY

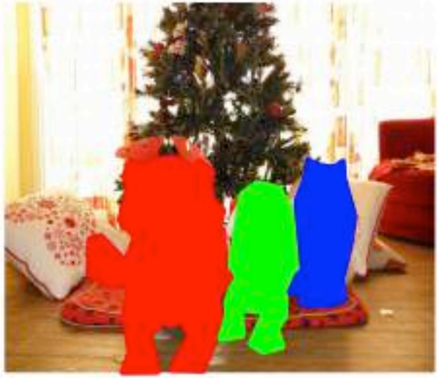
Classification + Localization

CAT

Object Detection

DOG, DOG, CAT

Instance Segmentation



DOG, DOG, CAT

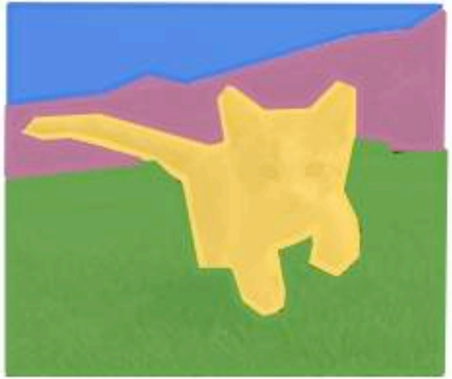
No objects, just pixels

Single Object

Multiple Object

Semantic Segmentation

Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

2D Object Detection

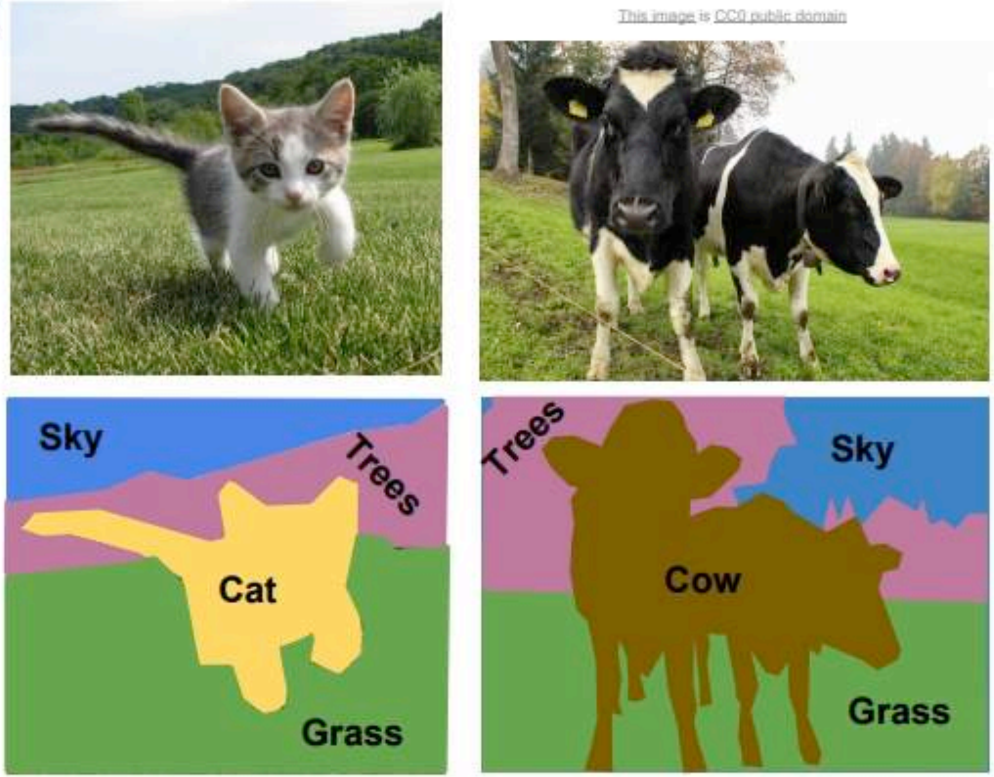
DOG, DOG, CAT

Object categories + 2D bounding boxes

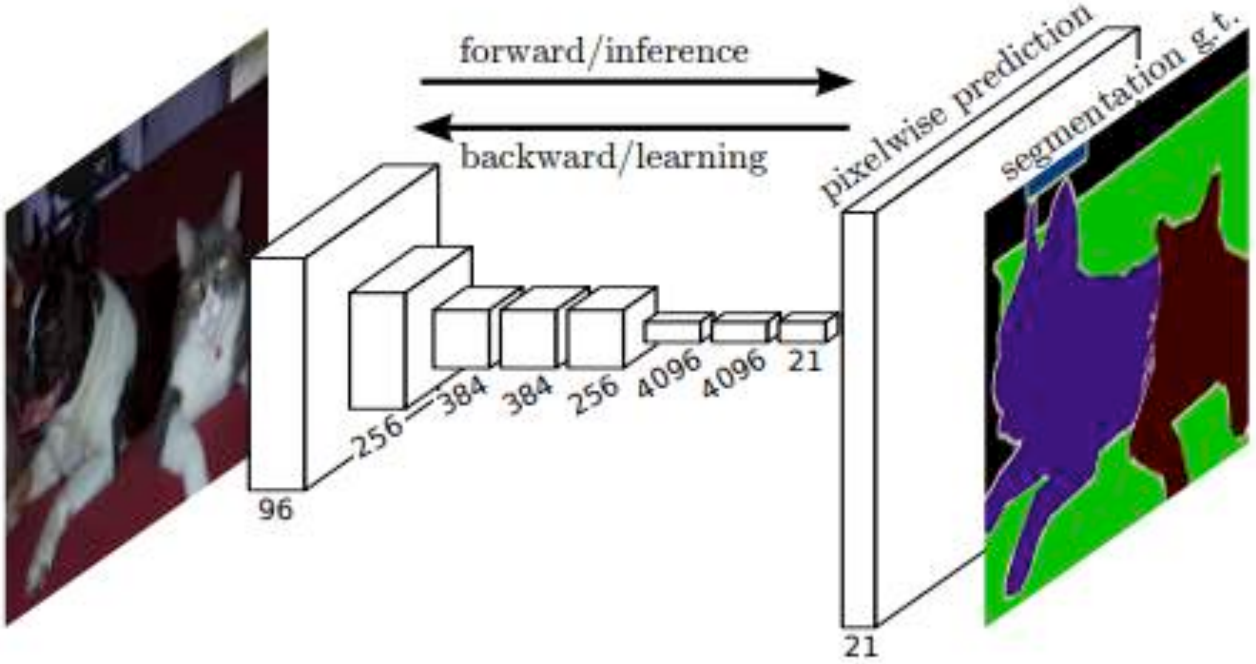
Semantic Segmentation

Label each pixel in the image with a category label

Don't differentiate instances, only care about pixels

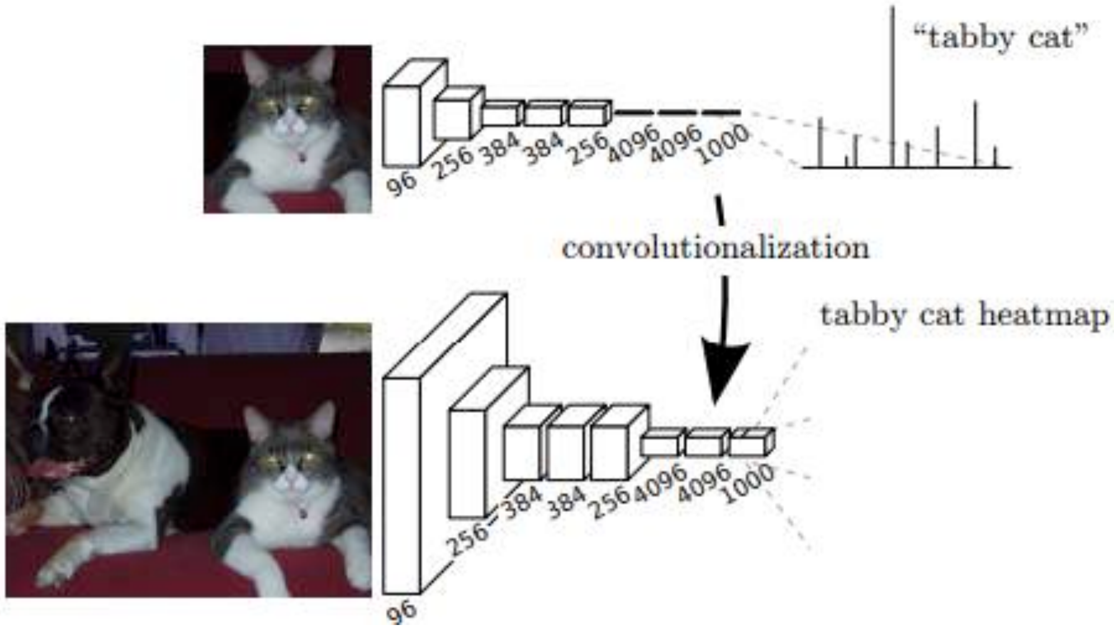


FCN for Semantic Segmentation



Long, Shelhamer, Darrell - Fully Convolutional Networks for Semantic Segmentation, CVPR 2015, PAMI 2016

“Convolutionalization”



“Convolutionalization”

Yann LeCun

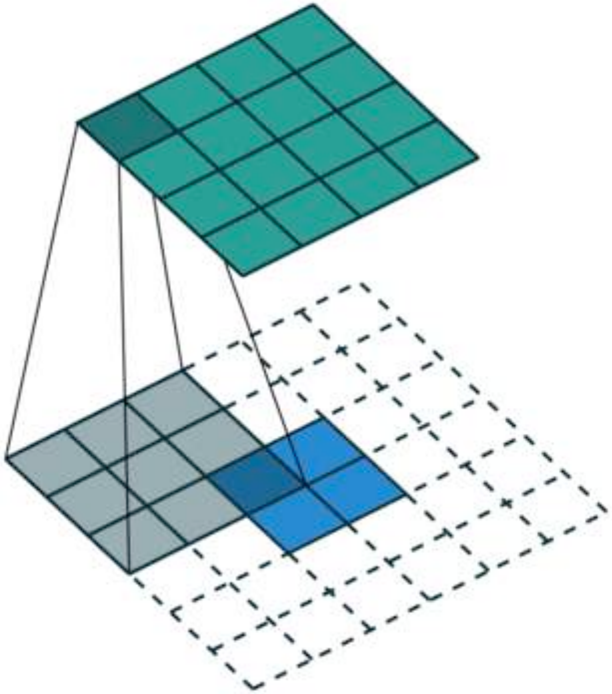
April 6, 2015 ·

In Convolutional Nets, there is no such thing as "fully-connected layers". There are only convolution layers with 1x1 convolution kernels and a full connection table.

It's a too-rarely-understood fact that ConvNets don't need to have a fixed-size input. You can train them on inputs that happen to produce a single output vector (with no spatial extent), and then apply them to larger images. Instead of a single output vector, you then get a spatial map of output vectors. Each vector sees input windows at different locations on the input.

In that scenario, the "fully connected layers" really act as 1x1 convolutions.

Digression: Transposed Convolutional Layers



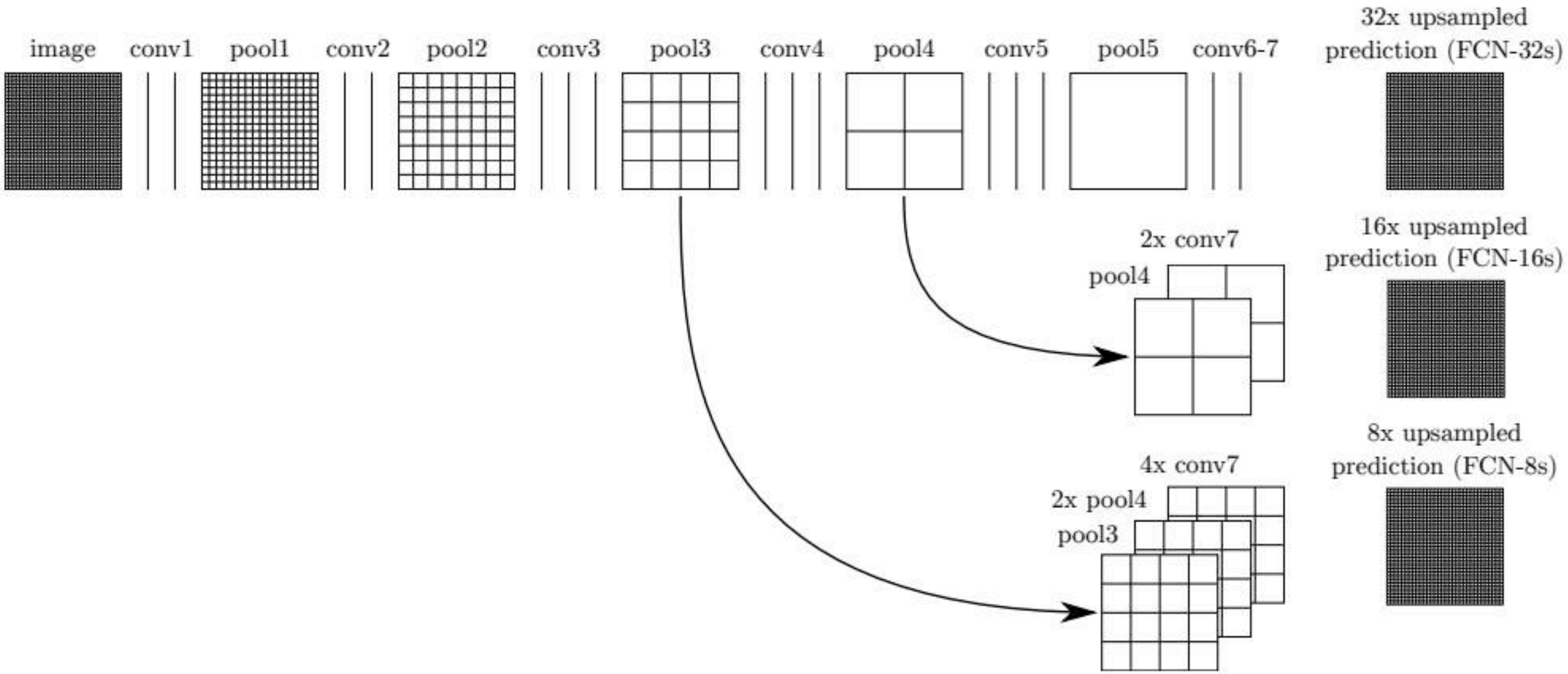
Known also as: upconvolutional layers, fractionally strided convolutions and (wrongly) deconvolutional layers.

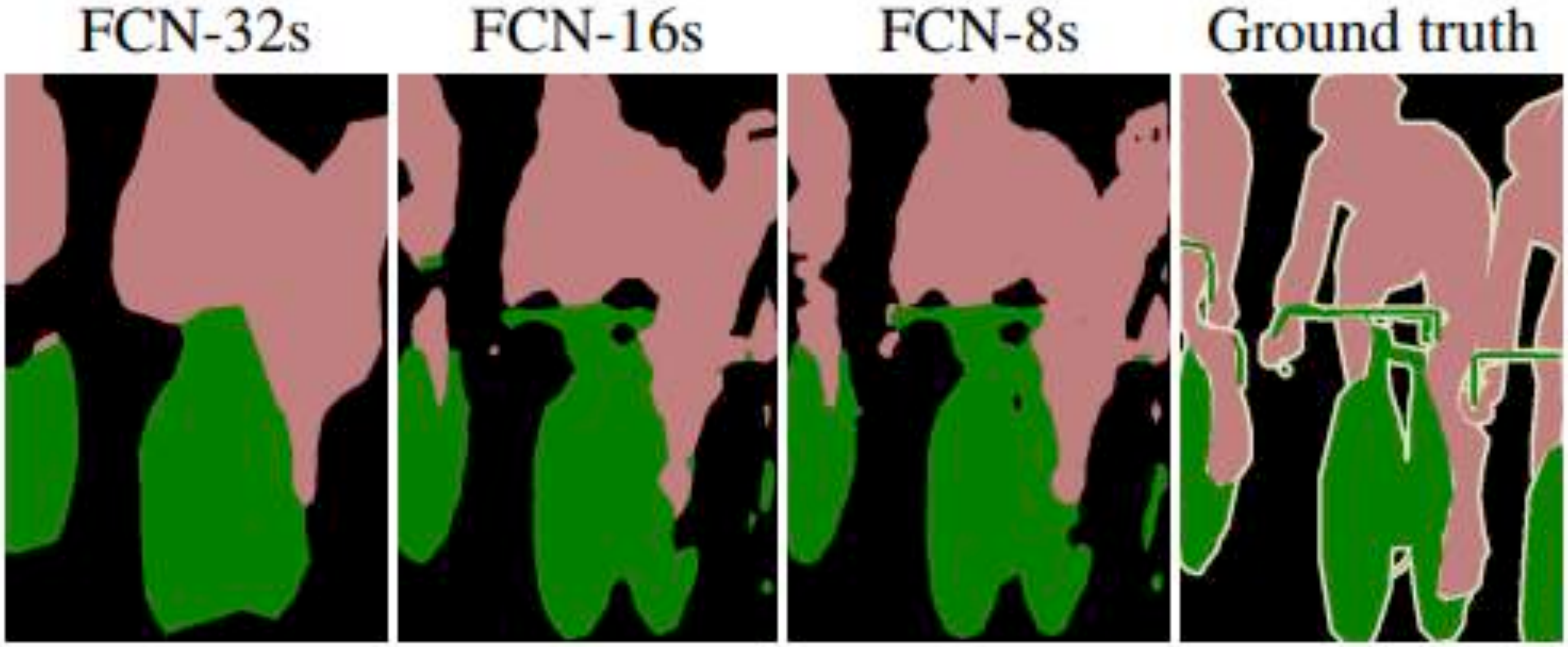
class`torch.nn.Conv2d`(*in_channels*, *out_channels*, *kernel_size*, *stride=1*, *padding=0*, *dilation=1*, *groups=1*, *bias=True*)

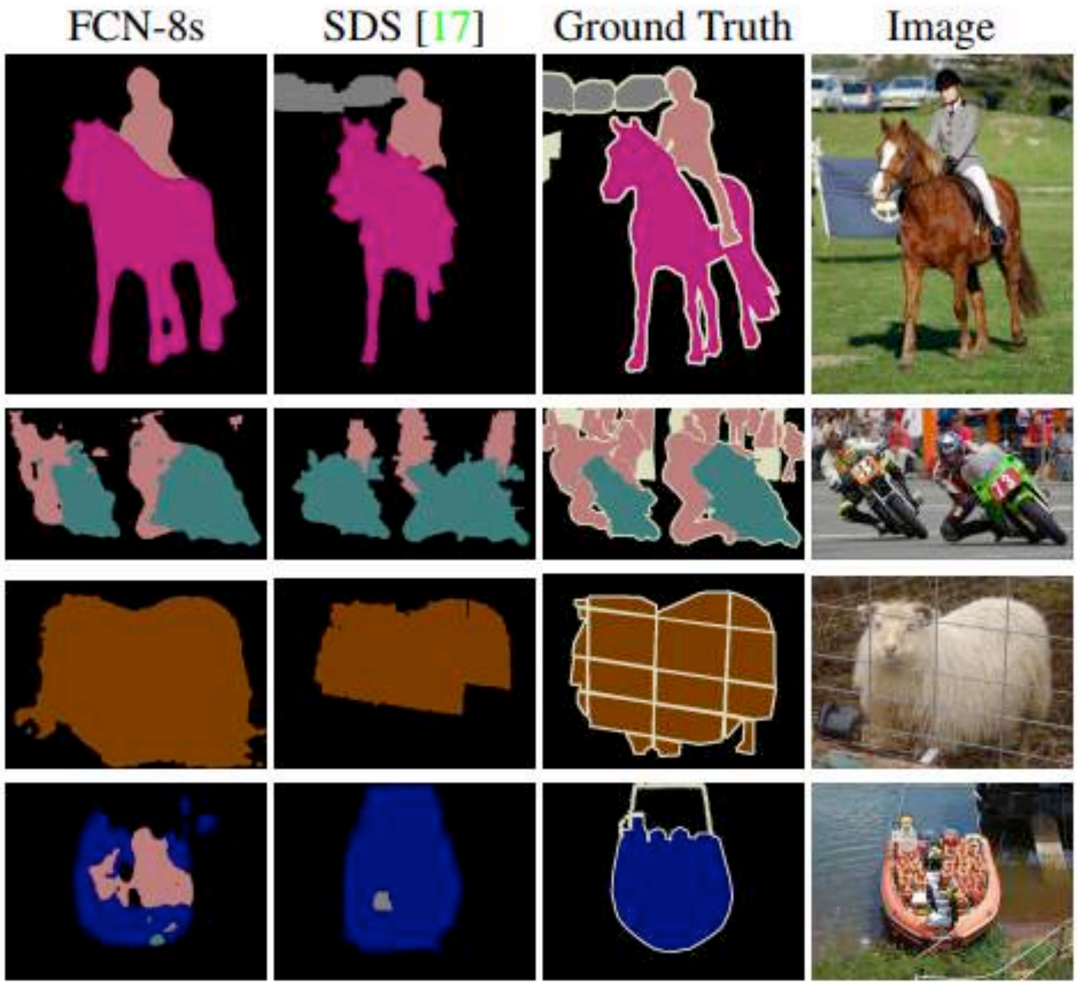
↓

class`torch.nn.ConvTranspose2d`(*in_channels*, *out_channels*, *kernel_size*, *stride=1*, *padding=0*, *output_padding=0*, *groups=1*, *bias=True*, *dilation=1*)

The Architecture




FCN-32s
FCN-16s
FCN-8s
Ground truth


FCN-8s
SDS [17]
Ground Truth
Image

DeepLab

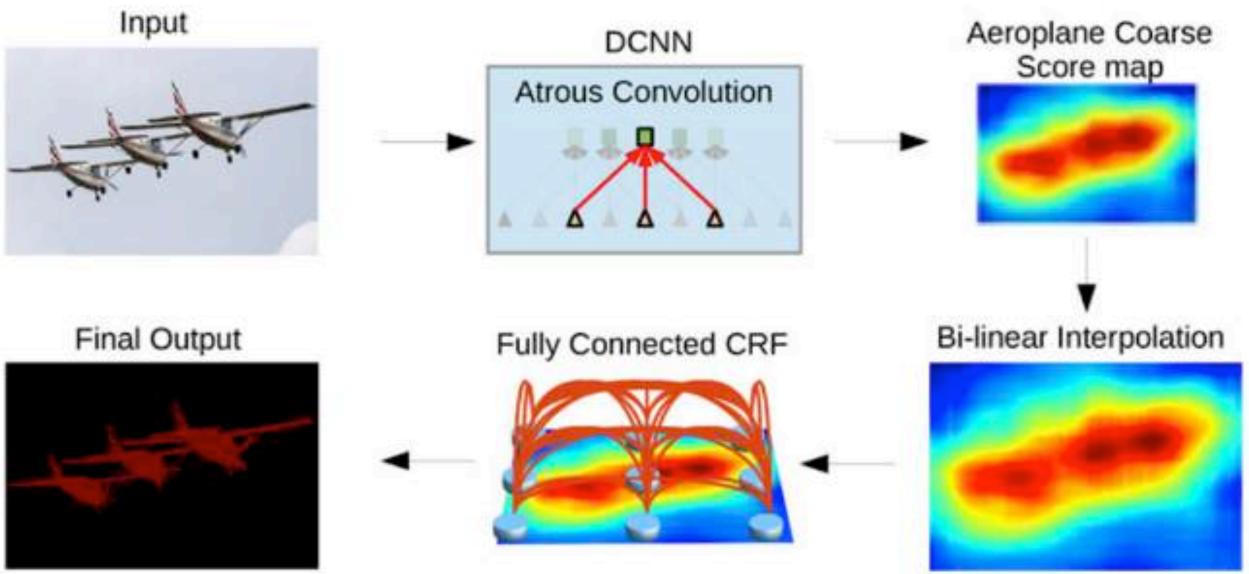
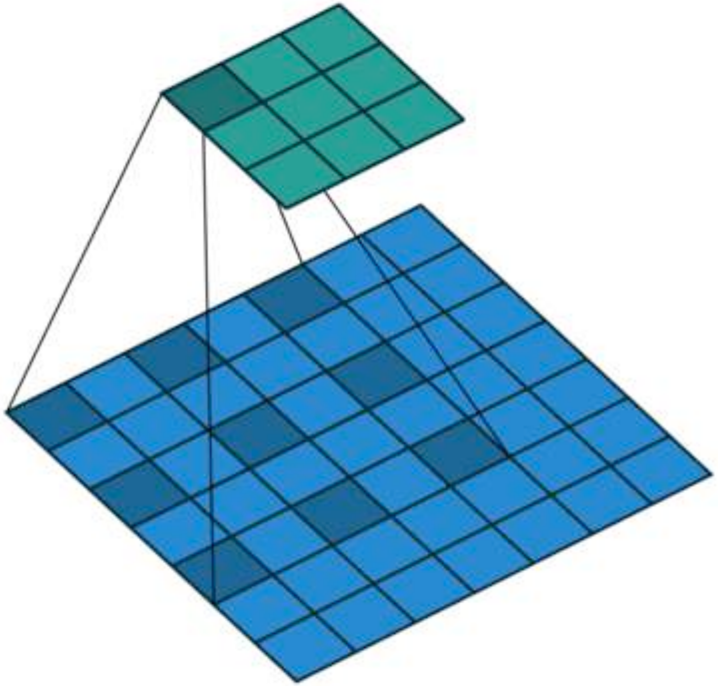


Fig. 1. Model illustration. A deep convolutional neural network such as VGG-16 or ResNet-101 is employed in a fully convolutional fashion, using atrous convolution to reduce the degree of signal downsampling (from 32x down 8x). A bilinear interpolation stage enlarges the feature maps to the original image resolution. A fully connected CRF is then applied to refine the segmentation result and better capture the object boundaries.

Chen, Papandreou, Kokkinos, Murphy, Yuille - "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." IEEE transactions on pattern analysis and machine intelligence (PAMI), 2018

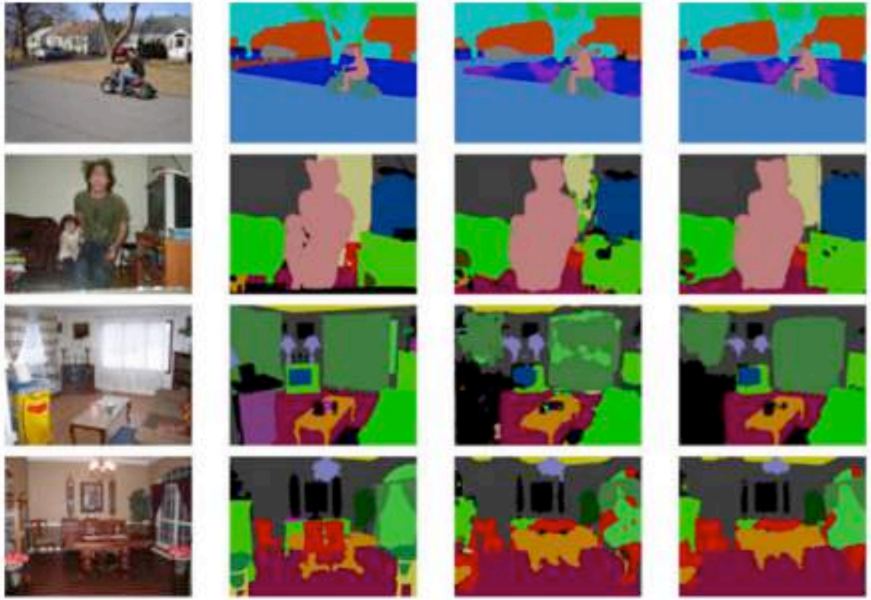
Digression: Dilated (Atrous) Convolutions



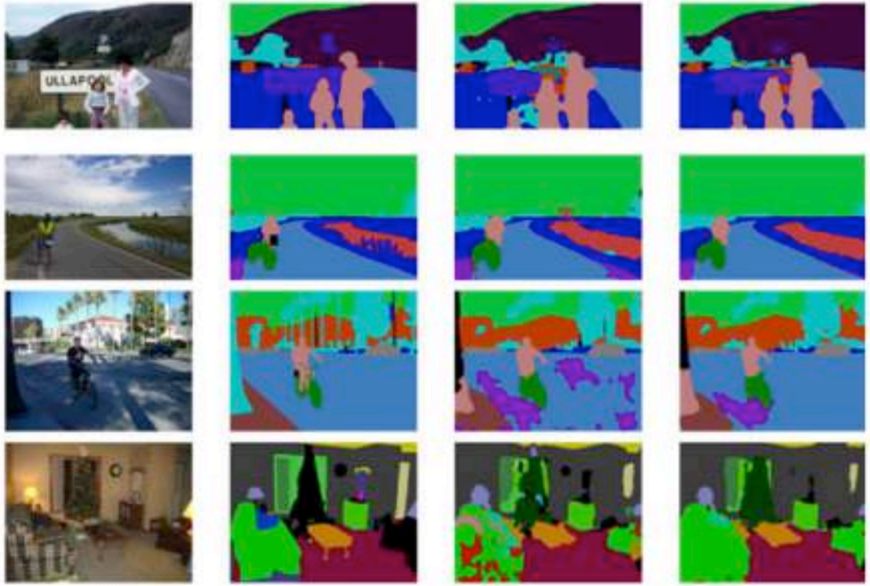
class`torch.nn.Conv2d`(*in_channels*, *out_channels*, *kernel_size*, *stride=1*, *padding=0*, *dilation=2*, *groups=1*, *bias=True*)

class`torch.nn.ConvTranspose2d`(*in_channels*, *out_channels*, *kernel_size*, *stride=1*, *padding=0*, *output_padding=0*, *groups=1*, *bias=True*, *dilation=2*)

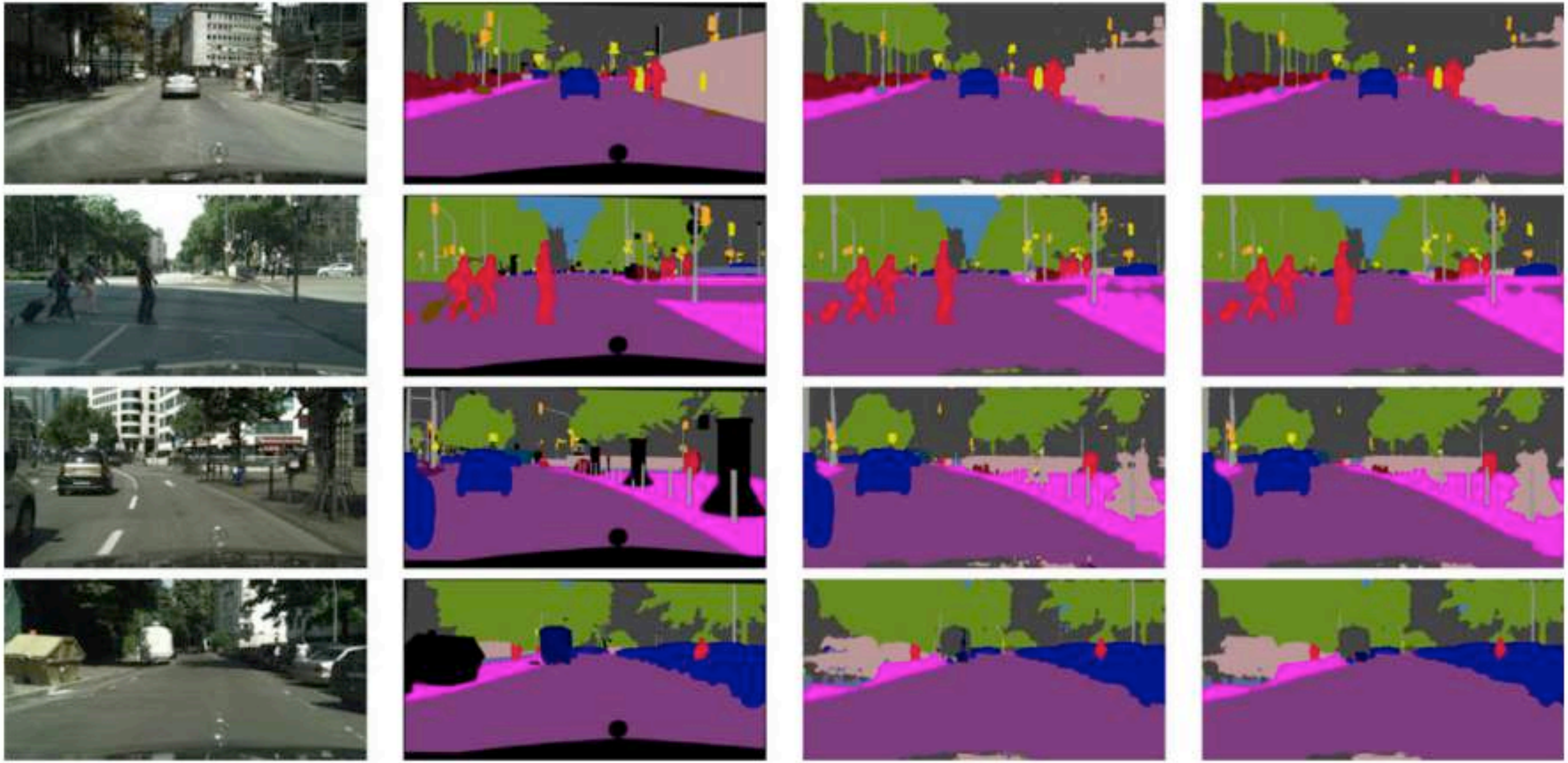
(a) Image (b) Before CRF (c) After CRF (a) Image (b) Before CRF (c) After CRF (a) Image (b) Before CRF (c) After CRF



(a) Image (b) G.T. (c) Before CRF (d) After CRF



(a) Image (b) G.T. (c) Before CRf (d) After CRF



(a) Image (b) G.T. (c) Before CRF (d) After CRF

Other important Segmentation Models

- U-Net
- Pyramid Scene Parsing Network.
- RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation.
- Segnet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation.
- Large Kernel Matters - Improve Semantic Segmentation by Global Convolutional Network.

...and many, many others.

On a side note

- Ciresan, Giusti, Gambardella and **Schmidhuber** - Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images, NIPS 2012.
- Farabet, Couprie, Najman and **LeCun** - Learning Hierarchical Features for Scene Labelling, PAMI 2013.

Yep, people knew for a long time that you might use convolutions for image segmentation.

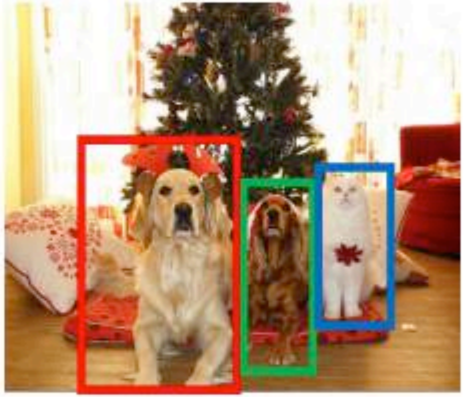
2D Object Detection

Semantic Segmentation

GRASS, CAT, TREE, SKY

No objects, just pixels

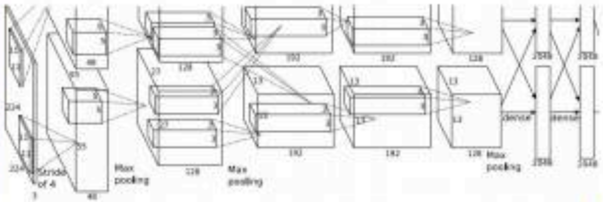
2D Object Detection



DOG, DOG, CAT

Object categories + 2D bounding boxes

Classification + Localization



Vector: 4096

Fully Connected: 4096 to 1000

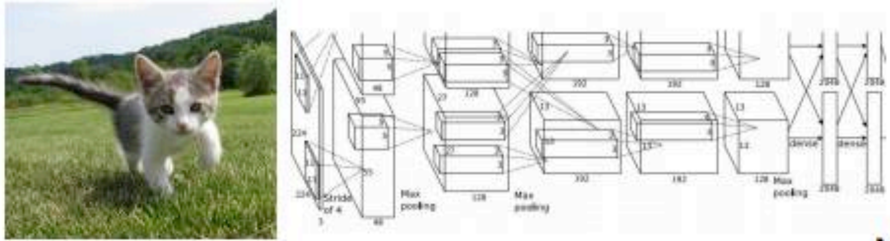
Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Fully Connected: 4096 to 4

Box Coordinates
(x, y, w, h)

Treat localization as a regression problem!

Classification + Localization



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)

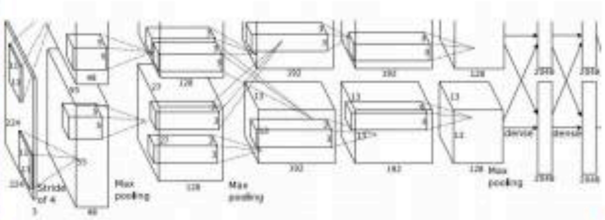
L2 Loss

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

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')

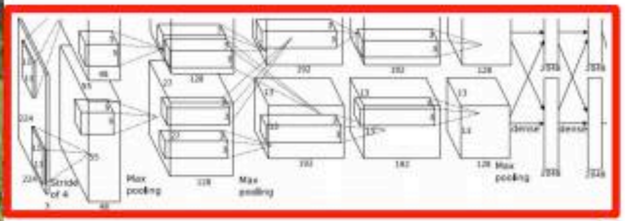
Multitask Loss

+ → Loss

Treat localization as a
regression problem!

Classification + Localization

This image is CC0 public domain



Often pretrained on ImageNet
(Transfer learning)

Vector: 4096

Fully Connected: 4096 to 1000

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

Correct label: Cat

Softmax Loss

+ → **Loss**

Fully Connected: 4096 to 4

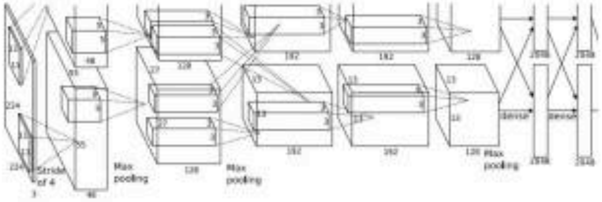
Box Coordinates (x, y, w, h)

L2 Loss

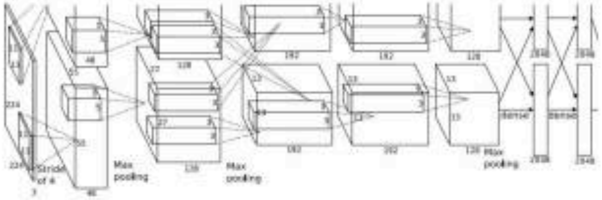
Correct box: (x', y', w', h')

Treat localization as a regression problem!

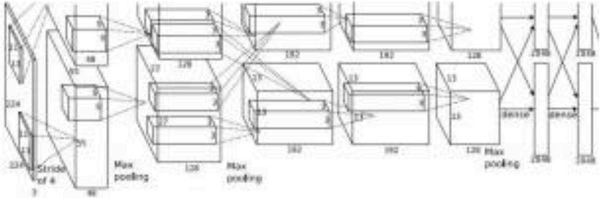
Object Detection as Regression?



CAT: (x, y, w, h)



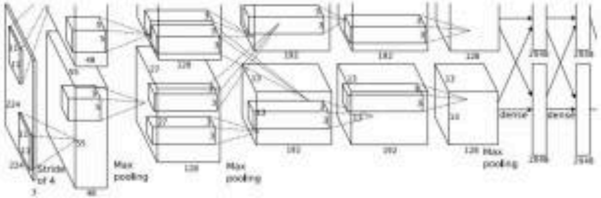
DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)



DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

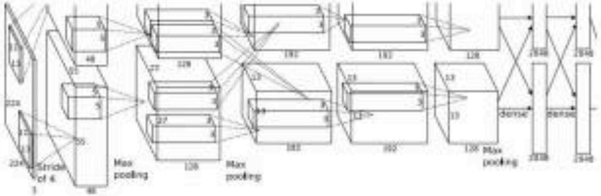
Object Detection as Regression?

Each image needs a different number of outputs!



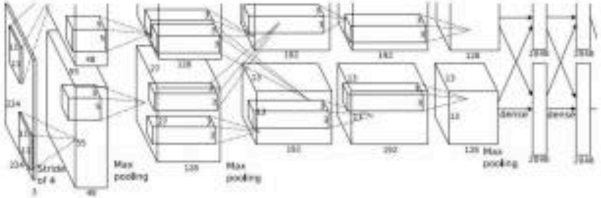
CAT: (x, y, w, h)

4 numbers



DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)

16 numbers

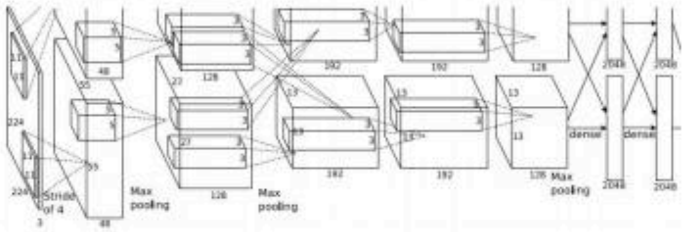


DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

Many numbers!

Object Detection as Classification: Sliding Window

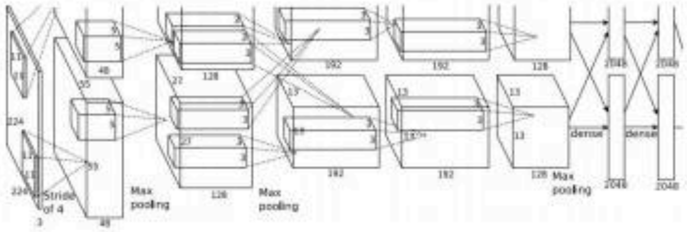
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? NO
Background? YES

Object Detection as Classification: Sliding Window

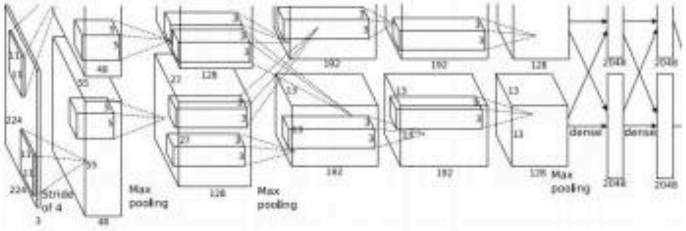
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

Object Detection as Classification: Sliding Window

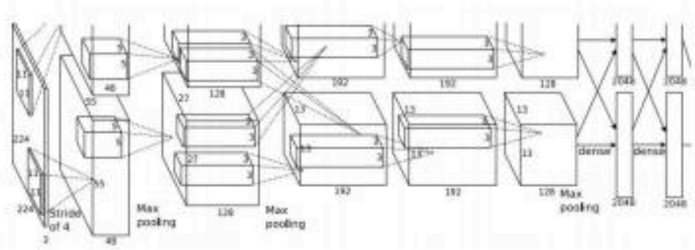
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

Object Detection as Classification: Sliding Window

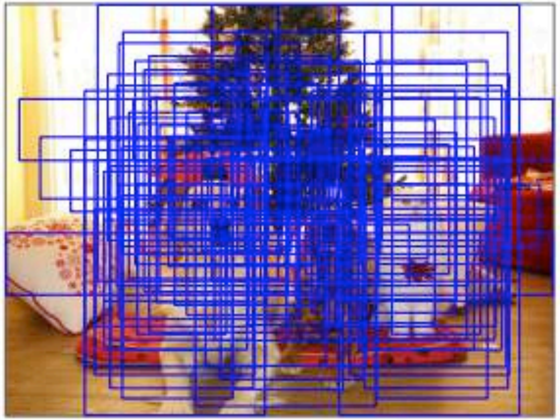
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

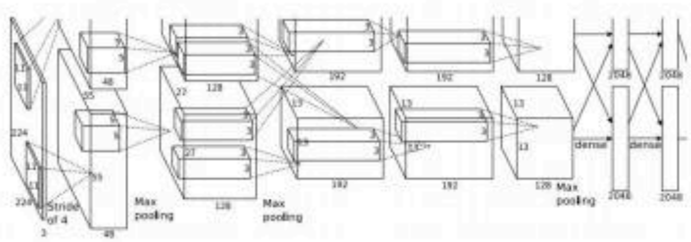


Dog? NO
Cat? YES
Background? NO

Object Detection as Classification: Sliding Window

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



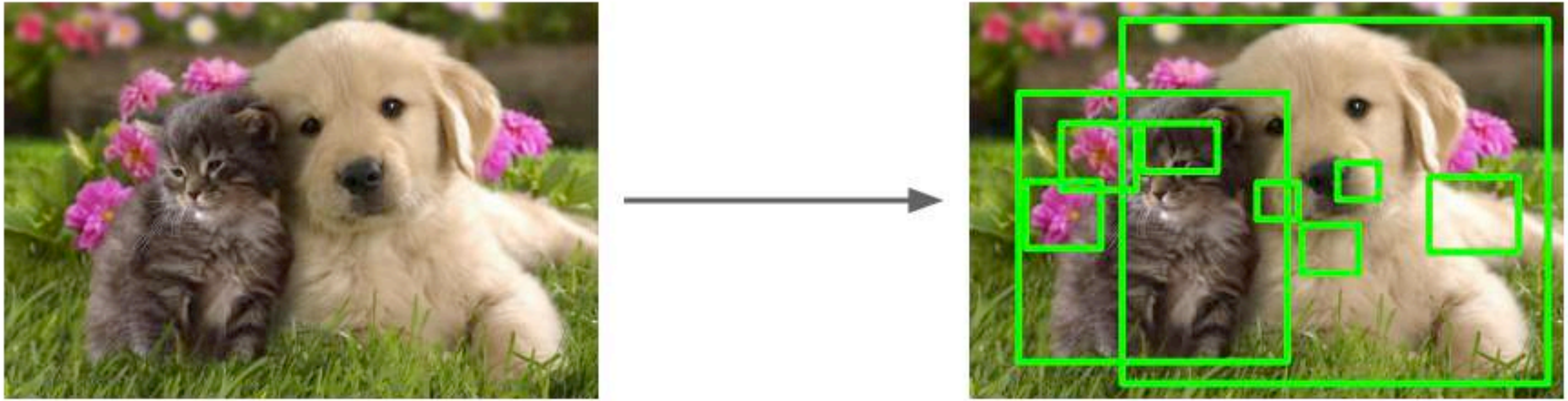


Dog? NO
Cat? YES
Background? NO

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



R-CNN

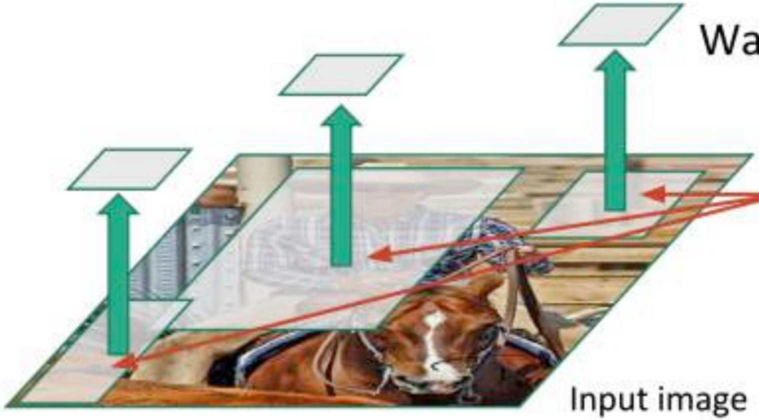
Girshick, Donahue, Darrell, Malik - Rich feature hierarchies for accurate object detection and semantic segmentation, CVPR 2014

R-CNN

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

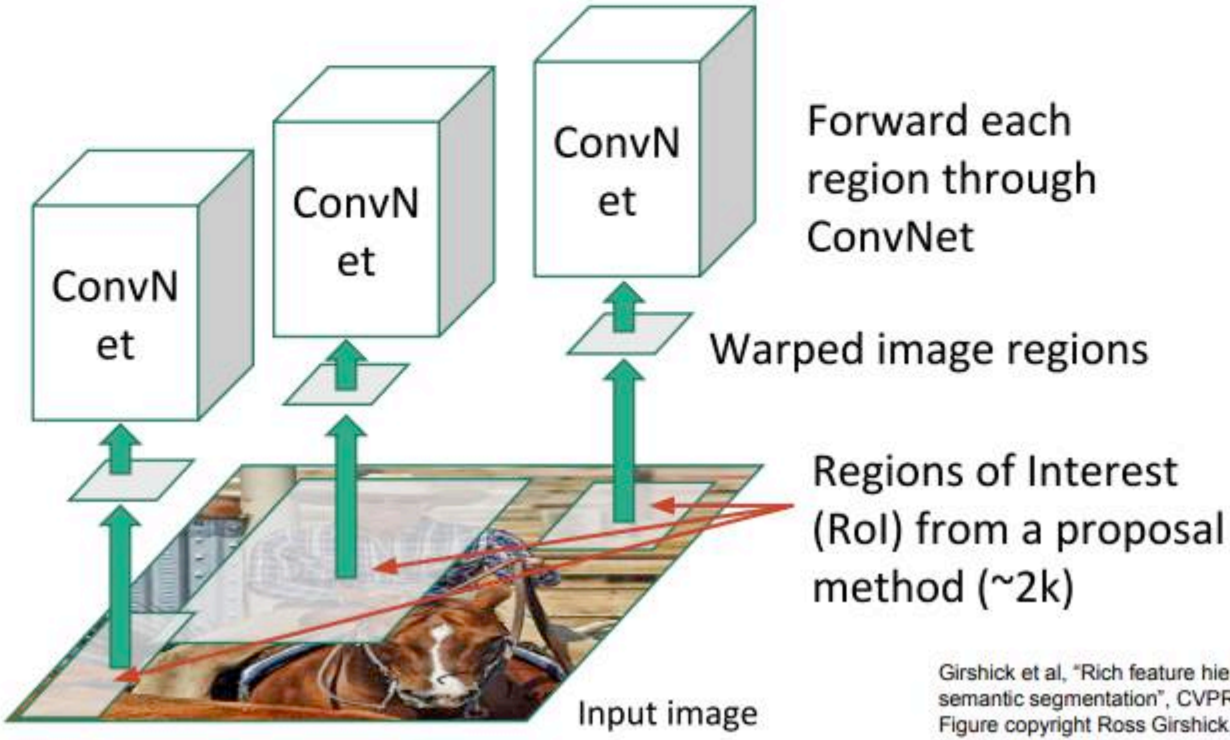
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

R-CNN



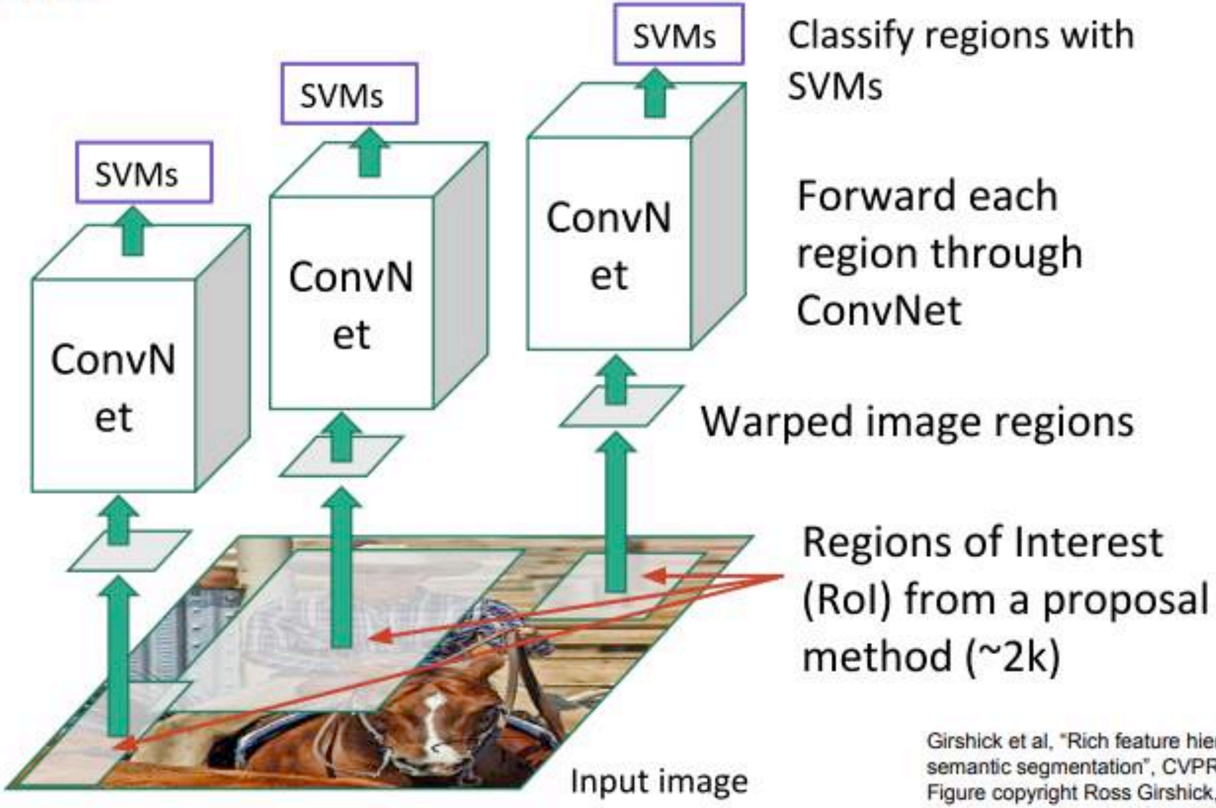
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

R-CNN



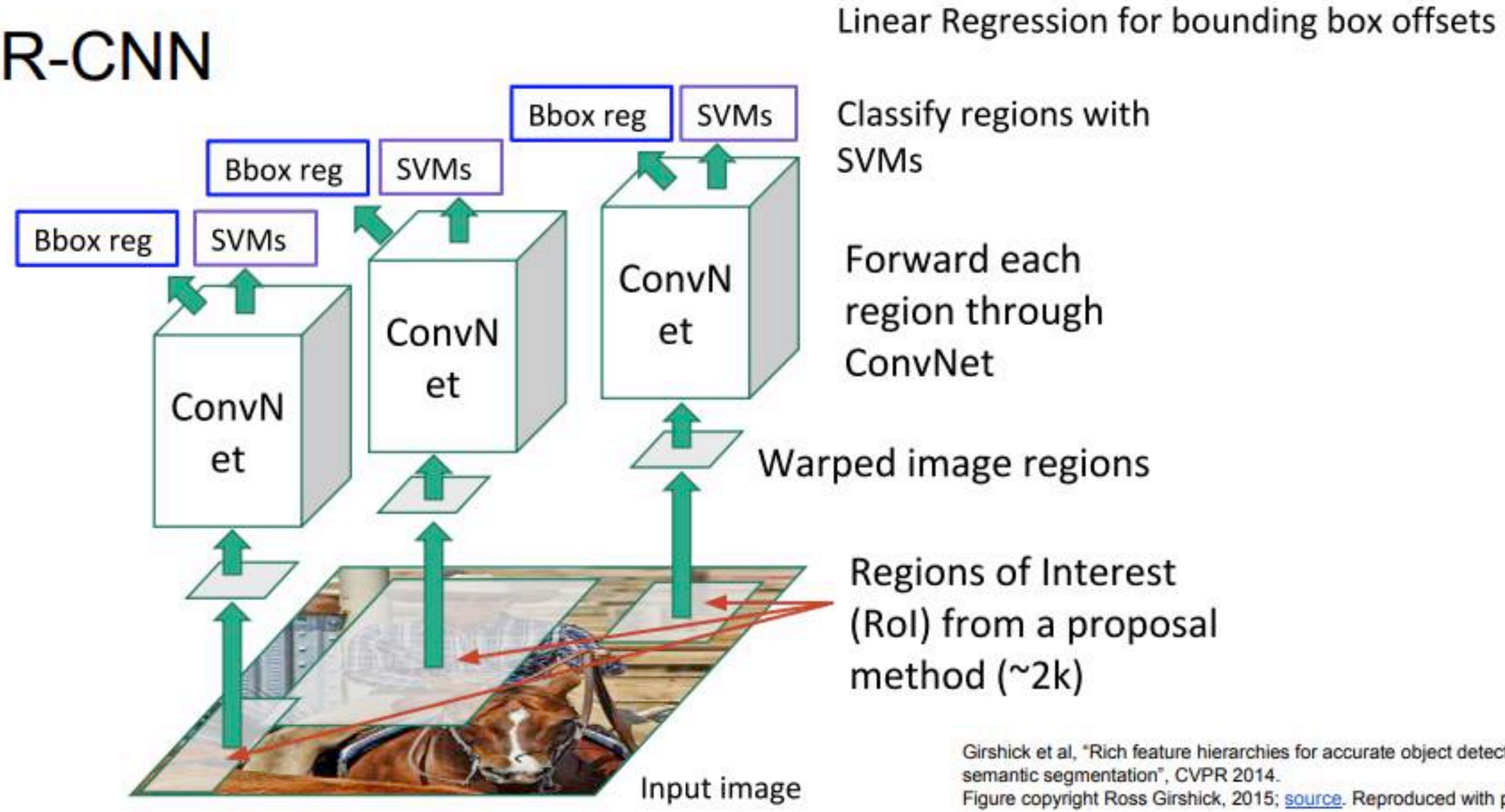
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

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.

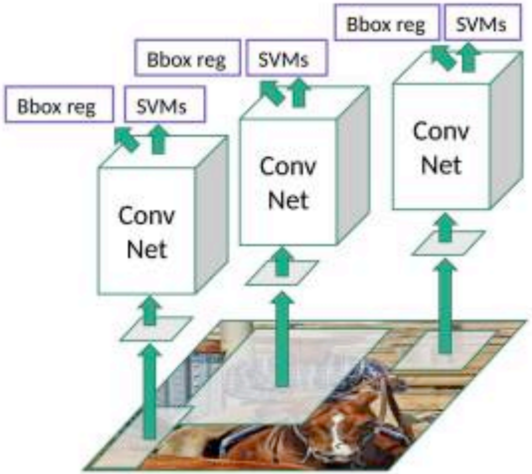
R-CNN



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

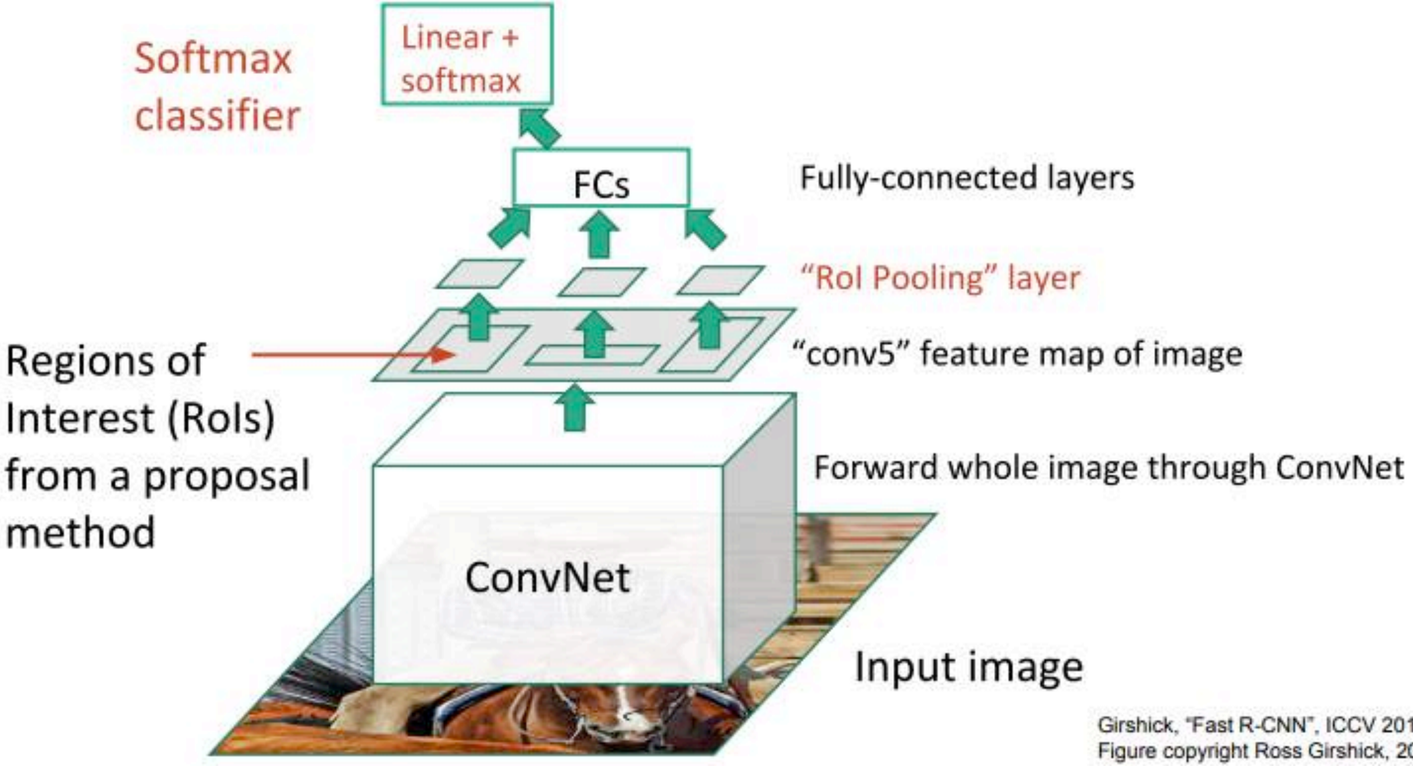
R-CNN: Problems

- Ad hoc training objectives
  - Fine-tune network with softmax classifier (log loss)
  - Train post-hoc linear SVMs (hinge loss)
  - Train post-hoc bounding-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]

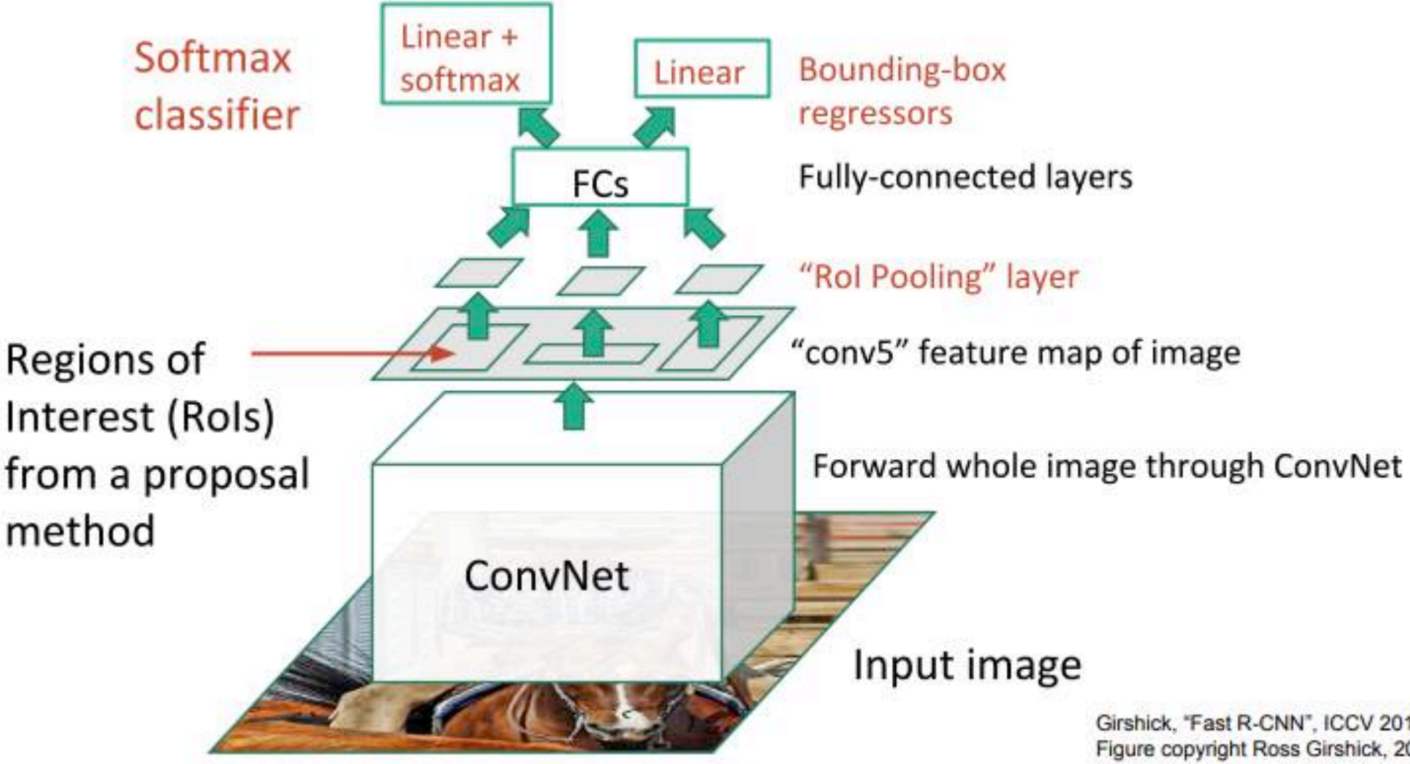


Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

Fast R-CNN

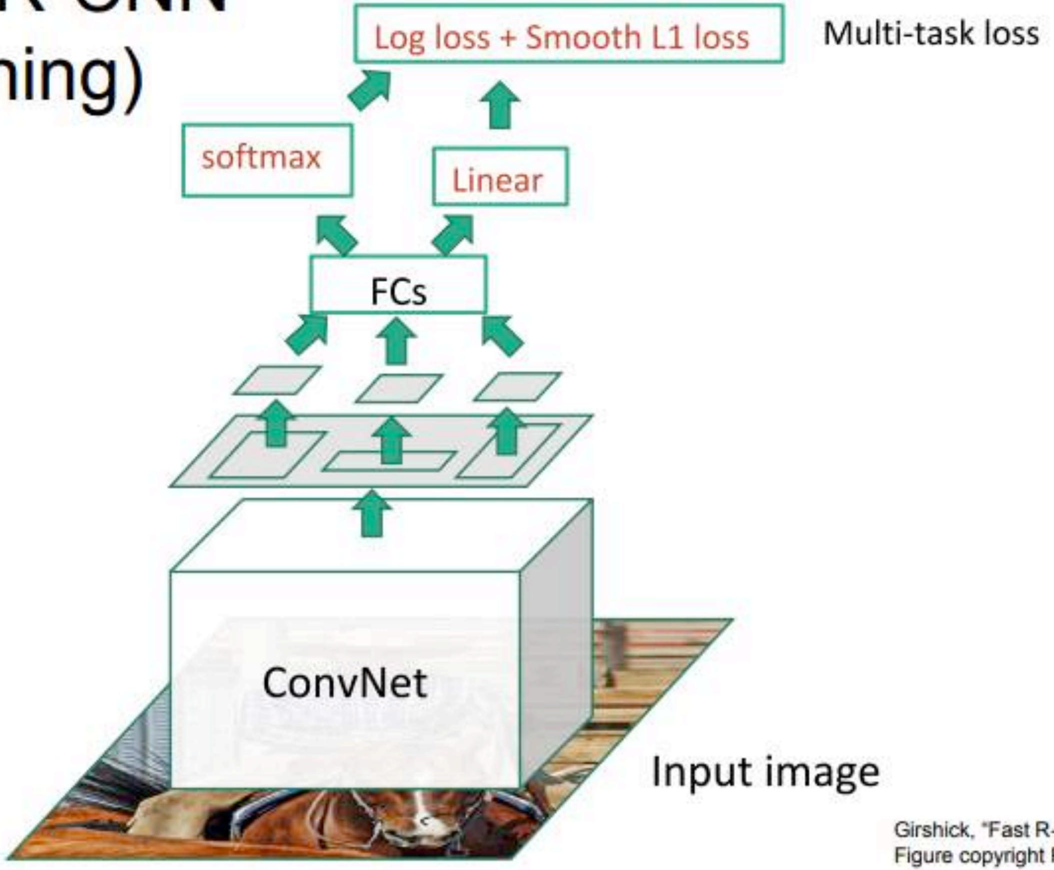

Softmax
classifier
Linear +
softmax
FCs
Fully-connected layers
"RoI Pooling" layer
Regions of
Interest (RoIs)
from a proposal
method
"conv5" feature map of image
Forward whole image through ConvNet
ConvNet
Input image

Fast R-CNN



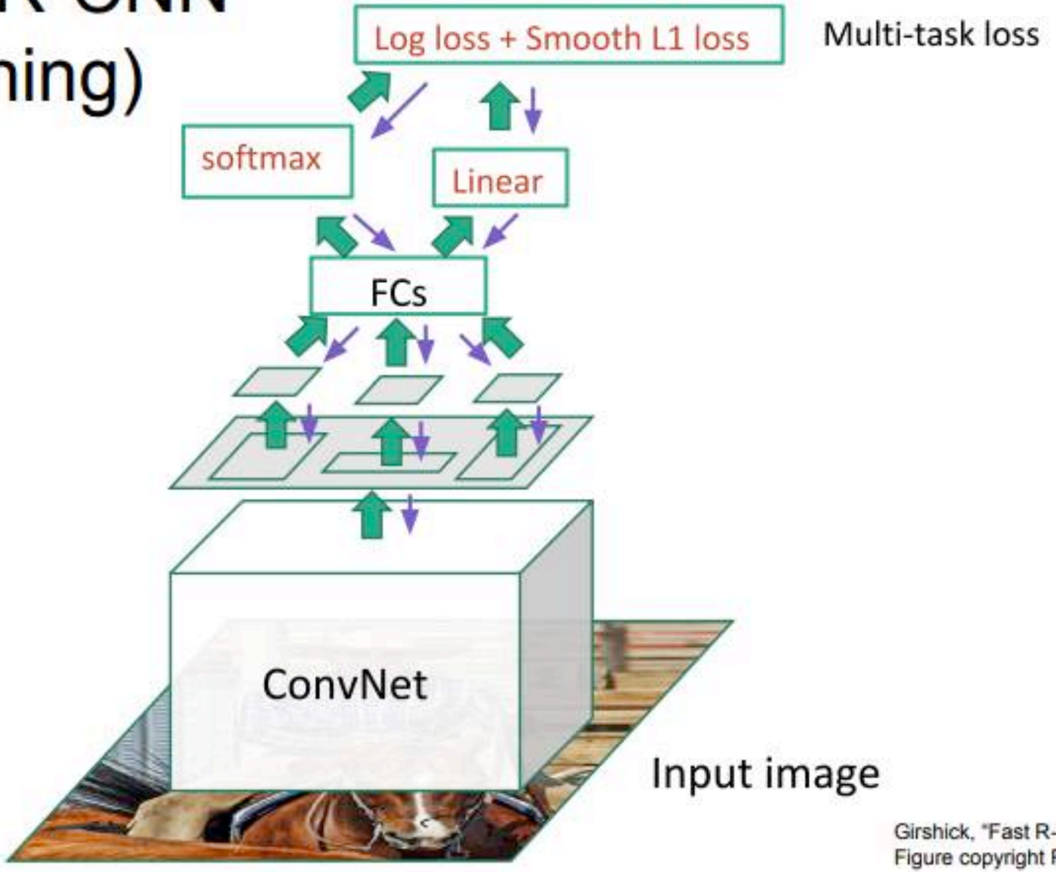
Girshick, "Fast R-CNN", ICCV 2015.

Fast R-CNN (Training)



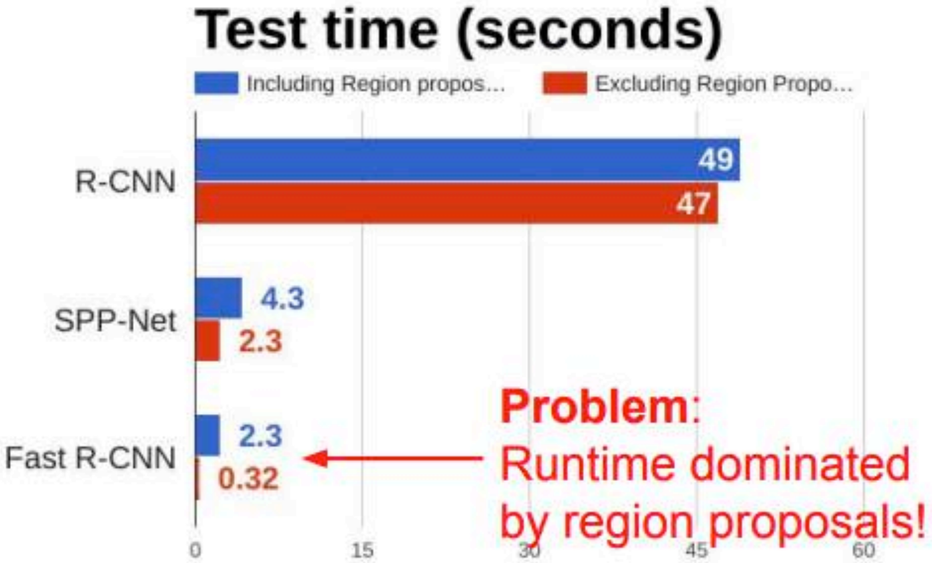
Girshick, "Fast R-CNN", ICCV 2015.

Fast R-CNN (Training)



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

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

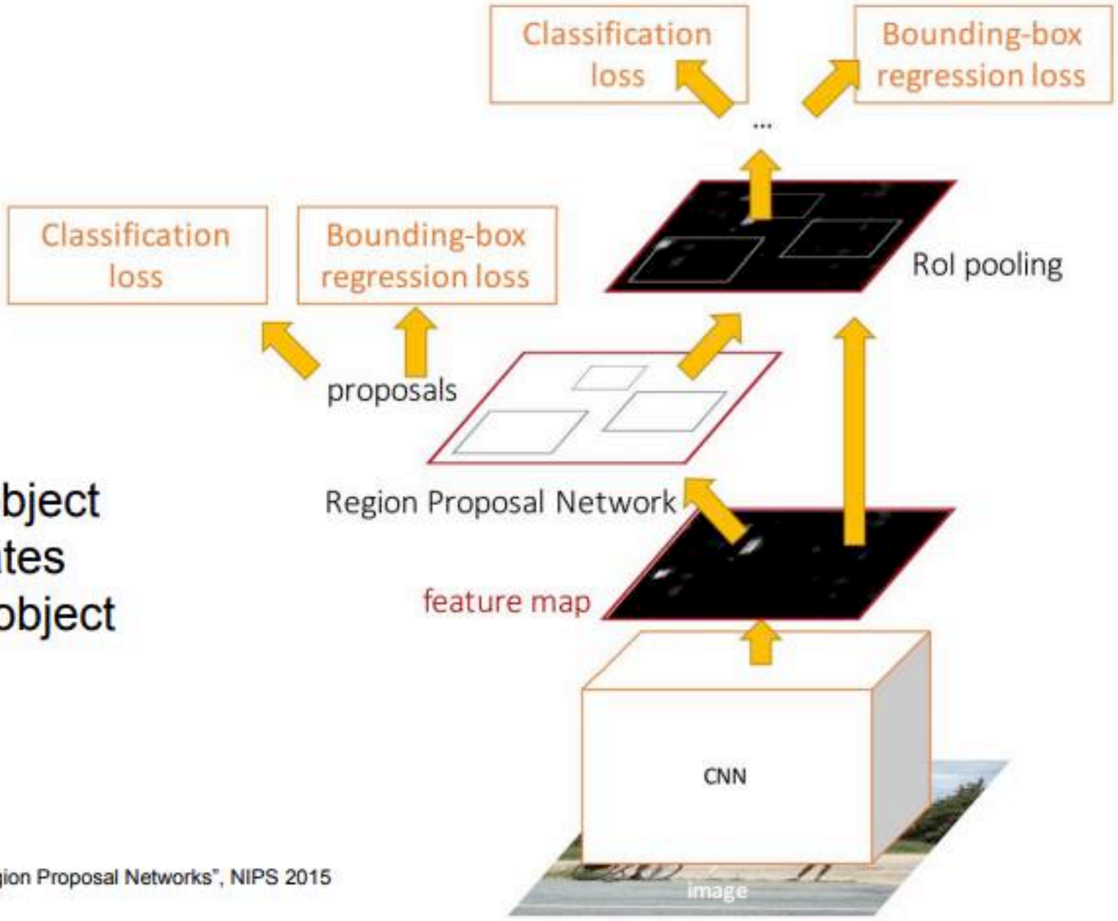
Faster R-CNN:

Make CNN do proposals!

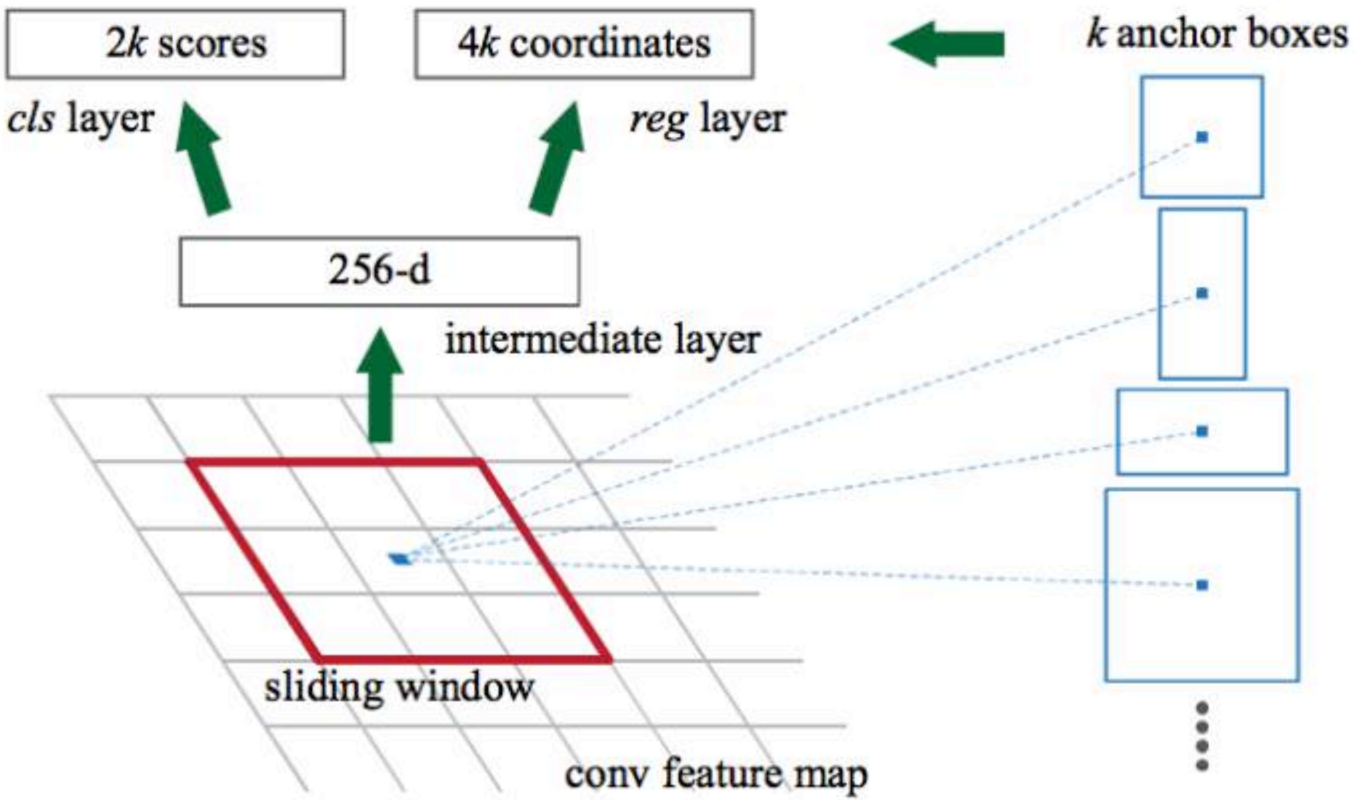
Insert **Region Proposal Network (RPN)** to predict proposals from features

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



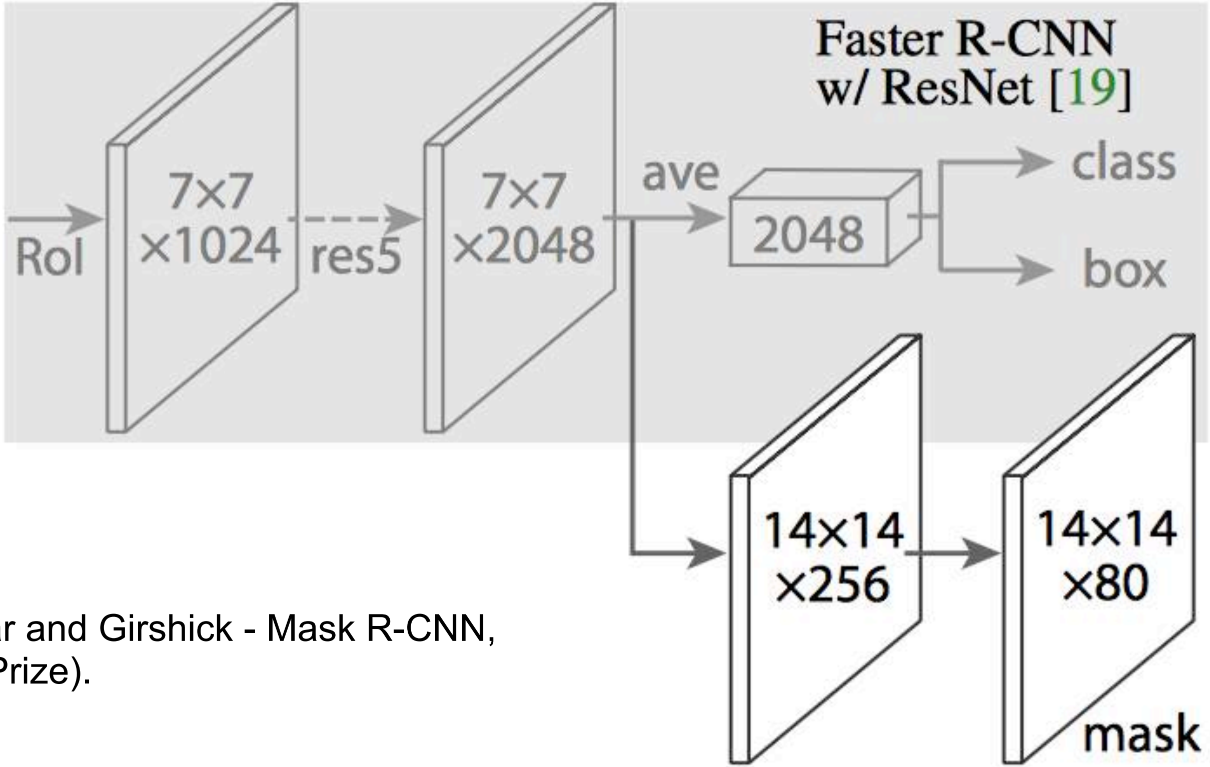
Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015



The Region Proposal Network slides a window over the features of the CNN. At each window location, the network outputs a score and a bounding box per anchor (hence 4k box coordinates where k is the number of anchors). Source: https://arxiv.org/abs/1506.01497.

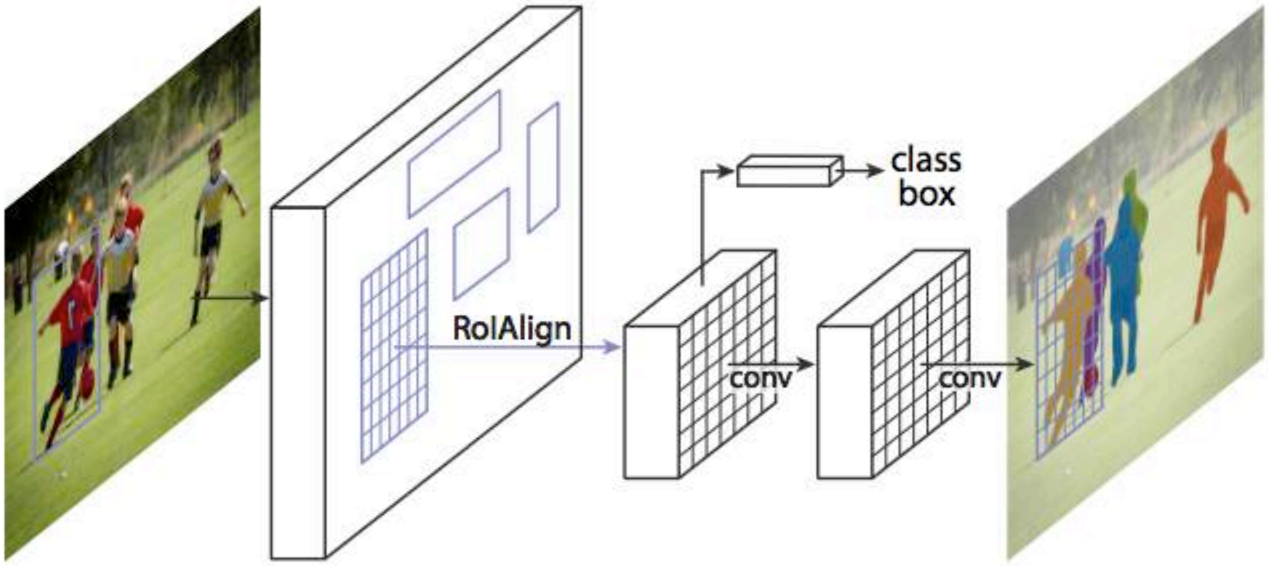
We know that the bounding boxes for people tend to be rectangular and vertical. We can use this intuition to guide our Region Proposal networks through creating an anchor of such dimensions. Image Source: http://vlm1.uta.edu/~athitsos/courses/cse6367_spring2011/assignments/assignment1/bbox0062.jpg.

Mask R-CNN

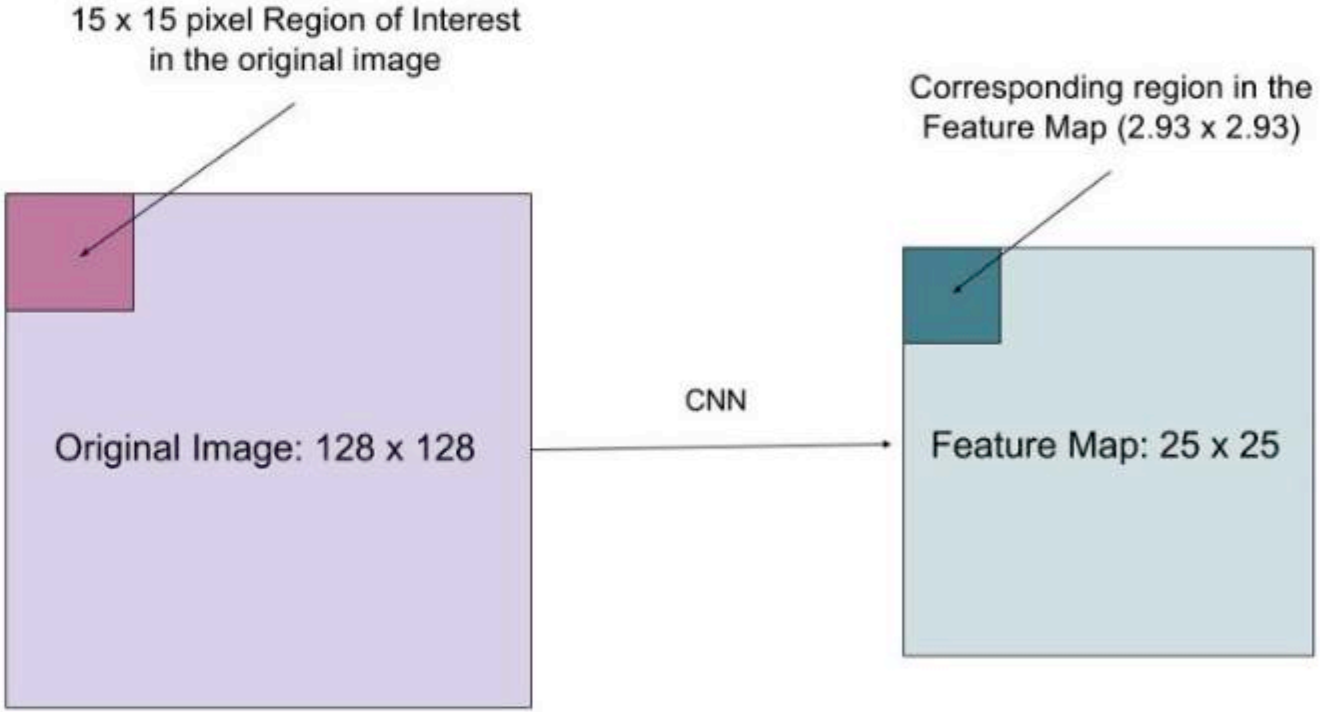


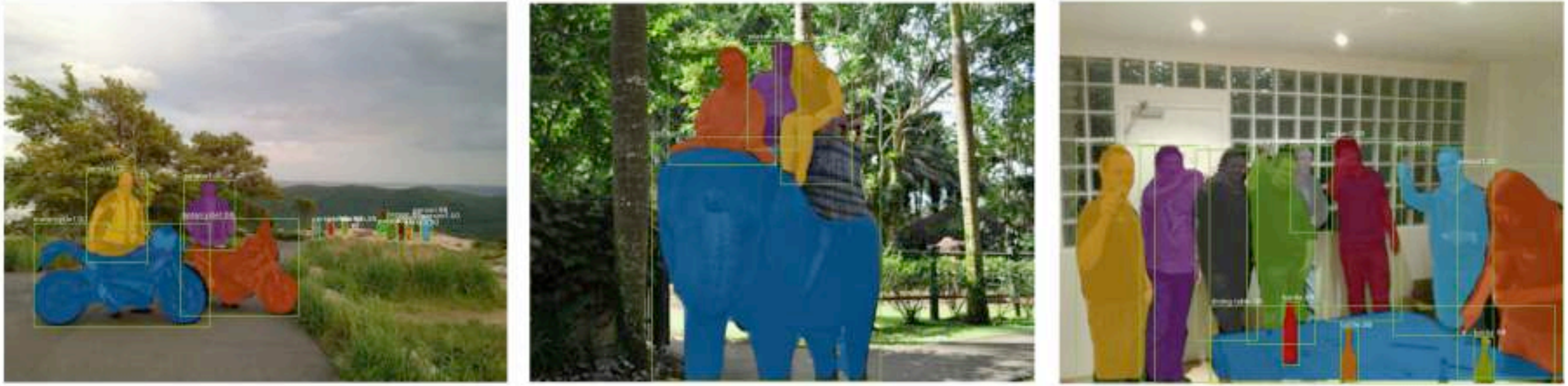
He, Gkioxari, Dollar and Girshick - Mask R-CNN, ICCV 2017 (Marr Prize).

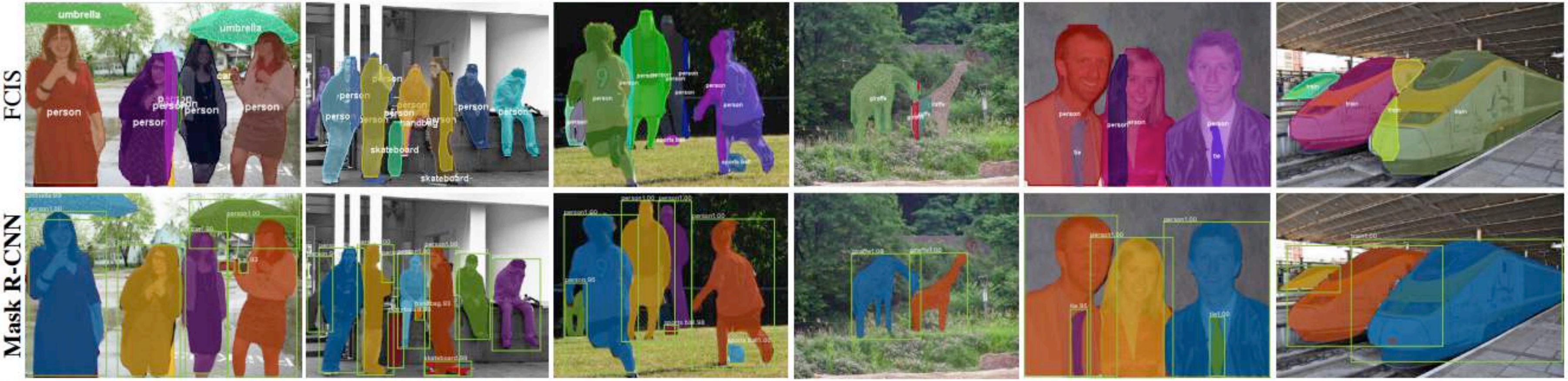
Mask R-CNN



Mask R-CNN


15 x 15 pixel Region of Interest
in the original image
Corresponding region in the
Feature Map (2.93 x 2.93)
Original Image: 128 x 128
CNN
Feature Map: 25 x 25




FCIS
Mask R-CNN
umbrella
umbrella
person
skateboard
skateboard
handbag


brewers.com

YOLO

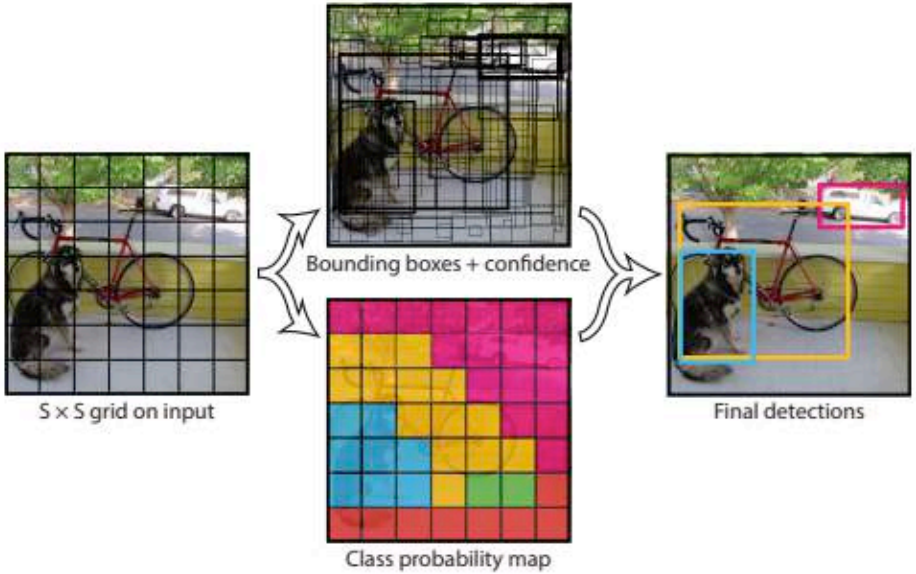
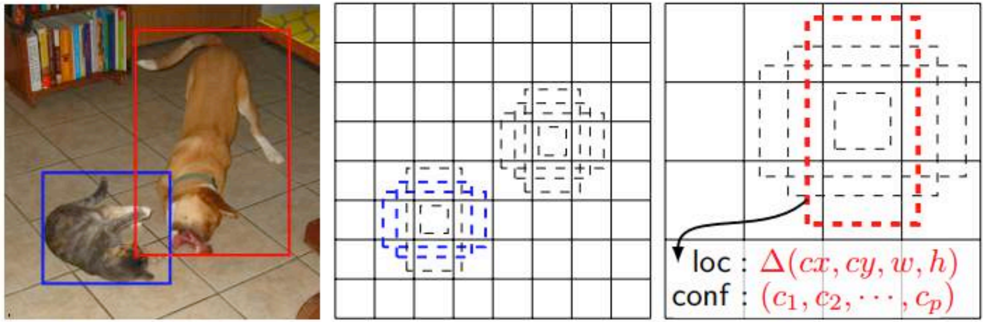


Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

Redmon, Divvala, Girshick, Farhadi - You Only Look Once, Real Time Object Detection, CVPR 2016.

SSD

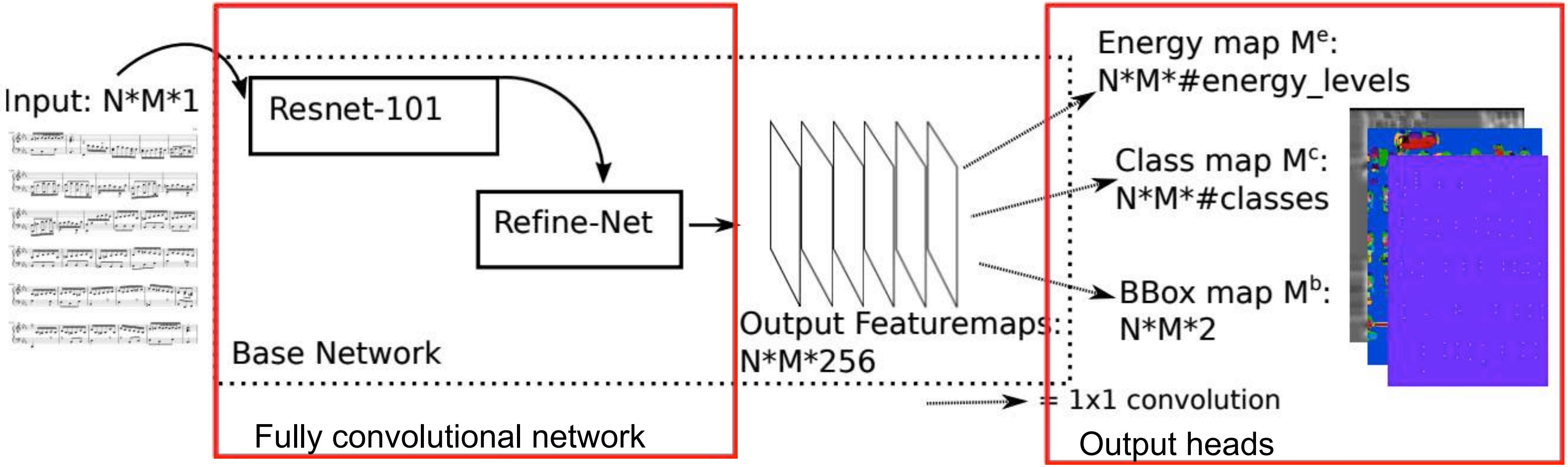
Liu, Anguelov, Erhan, Szegedy, Reed, Fu, Berg - SSD: Single Shot Box Detector, ECCV 2016.



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

Fig. 1: **SSD framework.** (a) SSD only needs an input image and ground truth boxes for each object during training. In a convolutional fashion, we evaluate a small set (e.g. 4) of default boxes of different aspect ratios at each location in several feature maps with different scales (e.g. 8×8 and 4×4 in (b) and (c)). For each default box, we predict both the shape offsets and the confidences for all object categories ($(c_1, c_2, \cdots, c_p)$). At training time, we first match these default boxes to the ground truth boxes. For example, we have matched two default boxes with the cat and one with the dog, which are treated as positives and the rest as negatives. The model loss is a weighted sum between localization loss (e.g. Smooth L1 [6]) and confidence loss (e.g. Softmax).

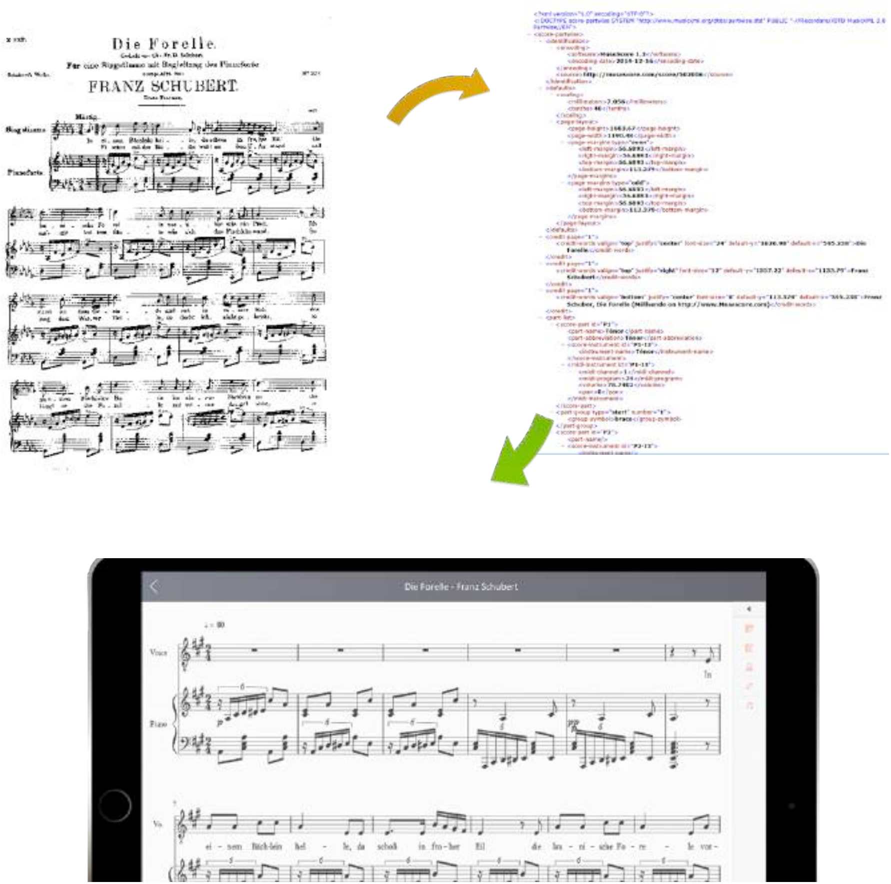
DWDNet



- The choice of FCNN is agnostic, though we use a RefineNet in our experiments.
- State-of-the-art results while having fast inference time (1-2 seconds for 2000 by 2000 images).
- Currently working on porting the solution to natural images (VOC, COCO, DOTA), and on domain adaptation (train on one dataset, do inference on some other dataset).

Tuggener, Elezi, Schmidhuber, Pelillo and Stadelmann - [DeepScores--A Dataset for Segmentation, Detection and Classification of Tiny Objects](), On ICPR 2018, Beijing, China.
Tuggener, Elezi, Schmidhuber and Stadelmann - [Deep Watershed Detector for Music Object Recognition](), On ISMIR 2018, Paris, France.
Elezi, Tuggener*, Pelillo and Stadelmann - [DeepScores and Deep Watershed Detection: current state and open issues](), On WORMS 2018, Paris, France*

DWDNet



mAP (%)	DeepScores (synthetic)	Muscima++ (handwritten)	DeepScores (scans)
Faster R-CNN	19.6	3.9	
RetinaNet	9.8	7.7	
U-NET	24.8	16.6	
DWDNet (ours)	**41.4**	**19.9**	47.3

DWDNet inference time is 1-2 orders of magnitude faster than U-NET, and around as fast as Faster R-CNN.

- Tomorrow - seminar on DWDNet and DeepScore project.

- Next Lecture (tuesday) - Generative Adversarial Networks.

Thank You!