Applications of Deep model

Many slides from Fei-fei Li's course @Standford and Xiaogang Wang's tutorial @CUHK
http://cs231n.stanford.edu/ http://www.ee.cuhk.edu.hk/~xgwang/

Outline

- Image classification and object recognition
- Image segmentation
- Object detection

Demos

- https://www.youtube.com/watch?v=VlH3OEhZnow
- https://www.youtube.com/watch?v=nus5-4cZr7c
- https://www.youtube.com/watch?v=nDqnMpE6b8s
- https://www.youtube.com/watch?v=RCM0u2tBl5E
- https://www.youtube.com/watch?v=BtYMOOrBb2E

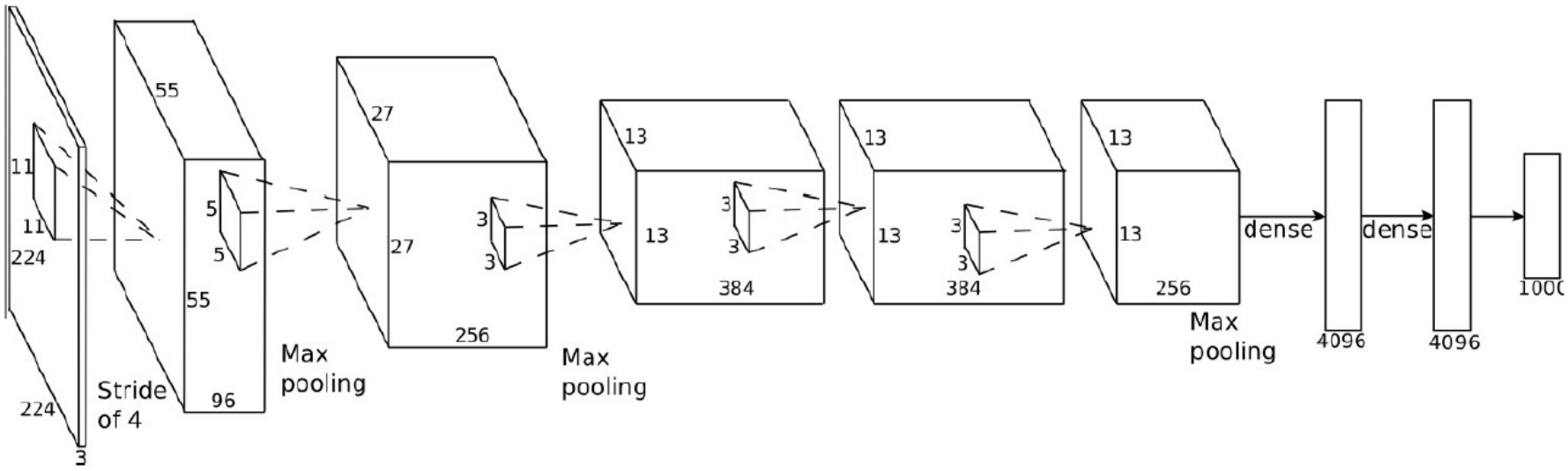
CNN for Object Recognition on ImageNet

- Krizhevsky, Sutskever, and Hinton, NIPS 2012
- Trained on one million images of 1000 categories collected from the web with two GPU; 2GB RAM on each GPU; 5GB of system memory
- Training lasts for one week

Rank	Name	Error rate	Description
1	**U. Toronto**	0.15315	Deep learning
2	U. Tokyo	0.26172	Hand-crafted features and learning models. Bottleneck.
3	U. Oxford	0.26979	
4	Xerox/INRIA	0.27058	

Model Architecture

- CNN + FC + Cross Entropy Loss
- Max-pooling layers follow 1st, 2nd, and 5th convolutional layers
- The number of neurons in each layer is given by 253440, 186624, 64896, 43264, 4096, 4096, 1000
- 650000 neurons, 60 million parameters, 630 connections



Normalization

- Normalize the input by subtracting the mean image on the training set

Input image (256 x 256)

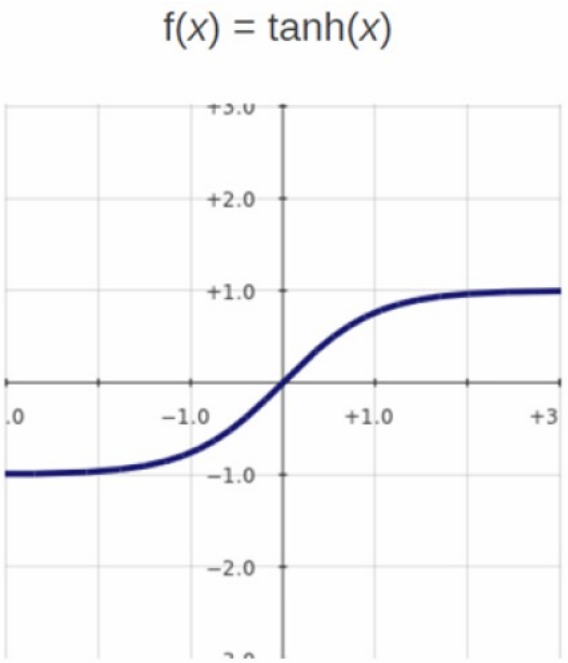
−

Mean image

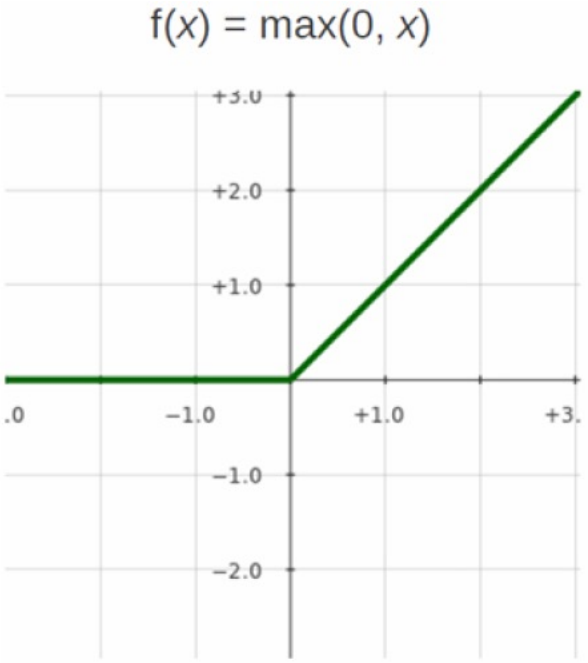
Krizhevsky 2012

Activation Function

- Rectified linear unit leads to sparse responses of neurons, such that weights can be effectively updated with BP



Sigmoid (slow to train)



Rectified linear unit (quick to train) √

Krizhevsky 2012

Data Augmentation

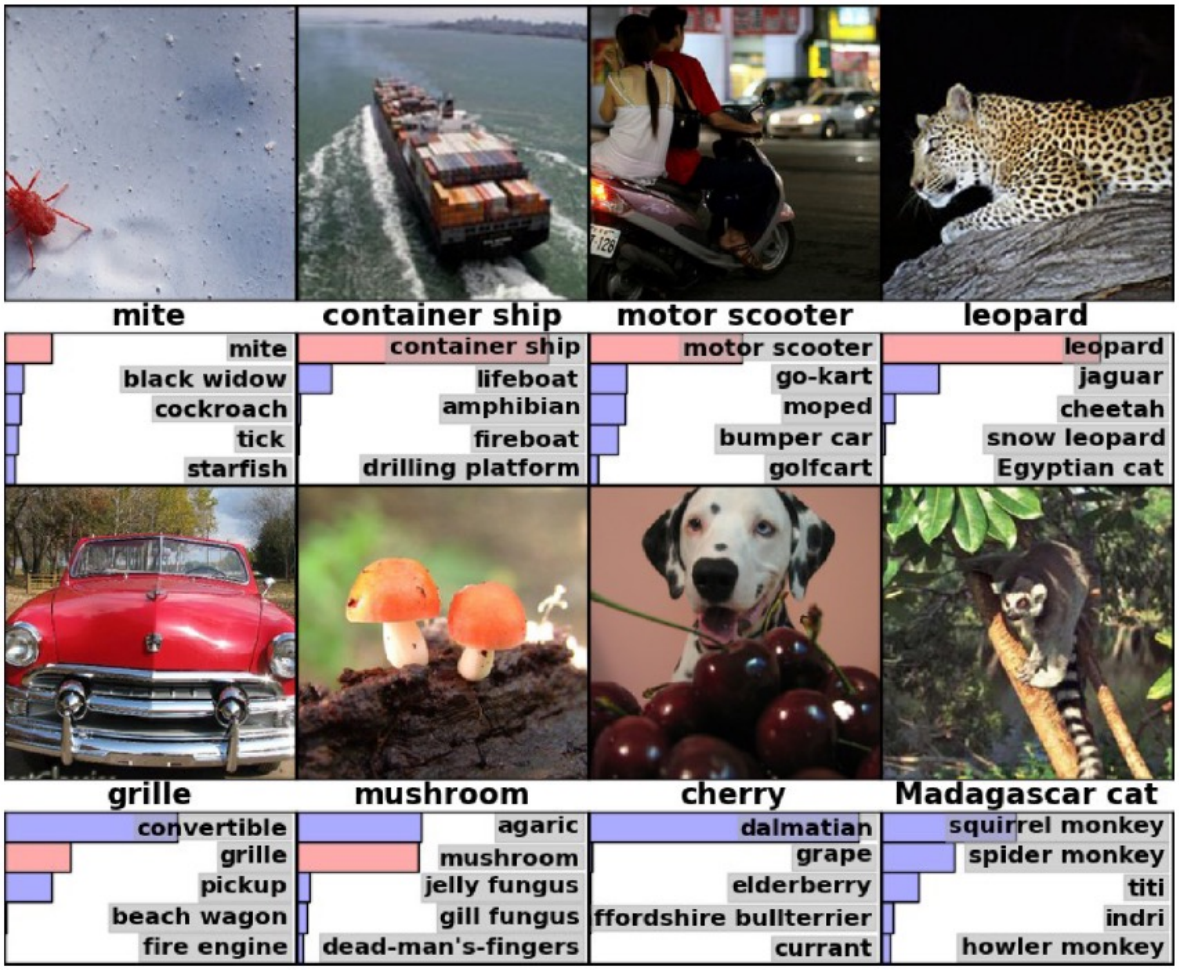
- The neural net has 60M parameters and it overfits
- Image regions are randomly cropped with shift; their horizontal reflections are also included

Krizhevsky 2012

Dropout

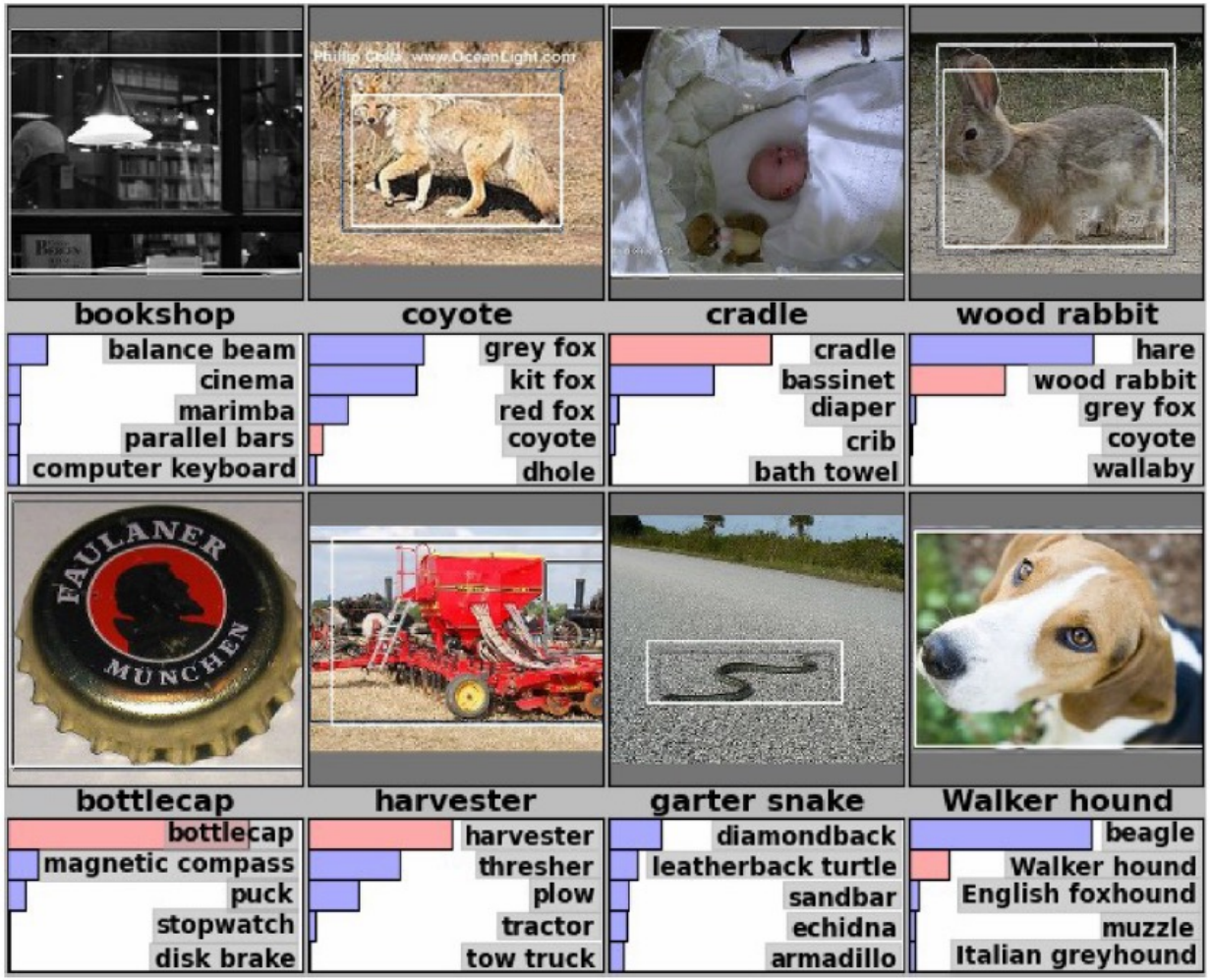
- Randomly set some input features and the outputs of hidden units as zero during the training process
- Feature co-adaptation: a feature is only helpful when other specific features are present
  - Because of the existence of noise and data corruption, some features or the responses of hidden nodes can be misdetected
- Dropout prevents feature co-adaptation and can significantly improve the generalization of the trained network
- Can be considered as another approach to regularization
- It can be viewed as averaging over many neural networks
- Slower convergence

Classification Result



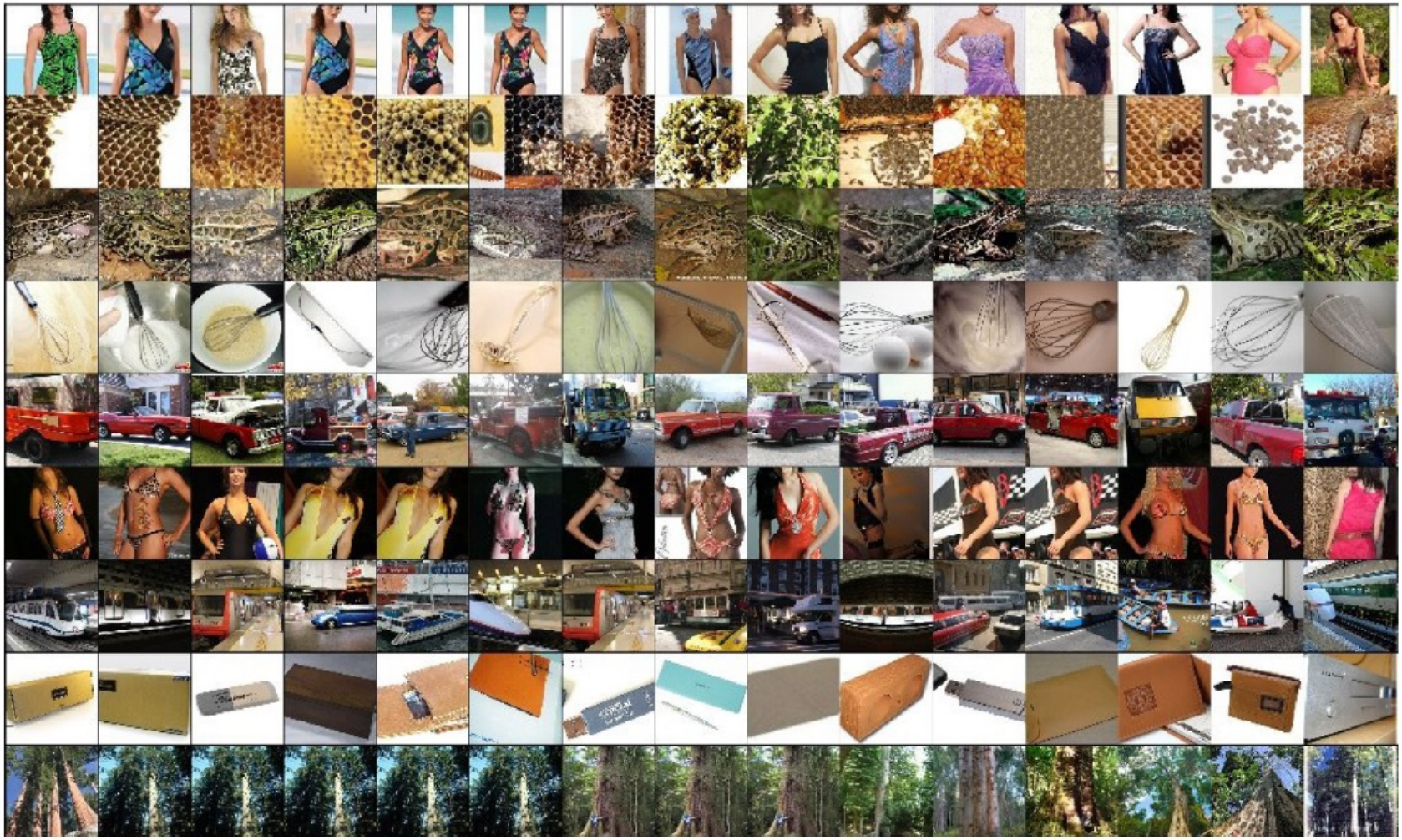
Krizhevsky 2012

Detection Result



Krizhevsky 2012

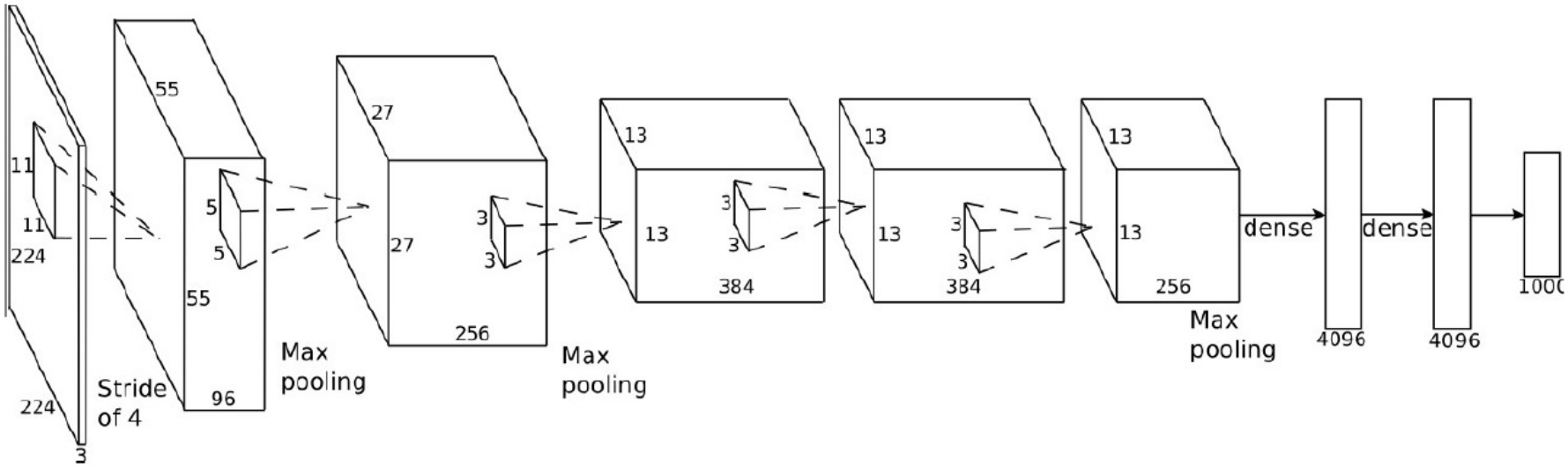
Image Retrieval



Krizhevsky 2012

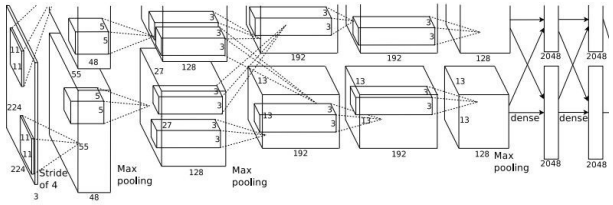
Adaptation to Smaller Datasets

- Directly use the feature representations learned from ImageNet and replace handcrafted features with them in image classification, scene recognition, fine grained object recognition, attribute recognition, image retrieval (Razavian et al. 2014, Gong et al. 2014)
- Use ImageNet to pre-train the model (good initialization), and use target dataset to fine-tune it (Girshick et al. CVPR 2014)
- Fix the bottom layers and only fine tune the top layers



So far: Image Classification

- CNN + FC + Cross Entropy Loss



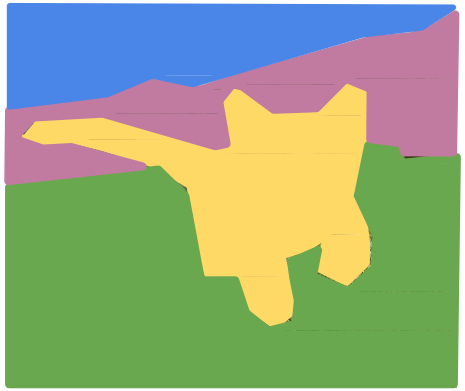
Vector:
4096

Fully-Connected:
4096 to 1000

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

Other Computer Vision Tasks

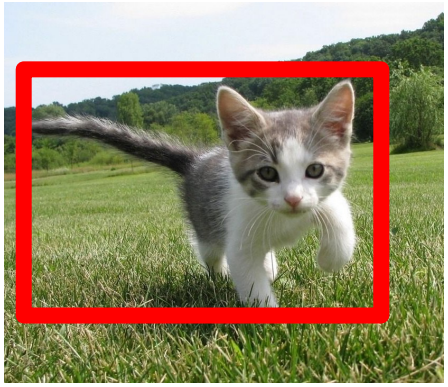
Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

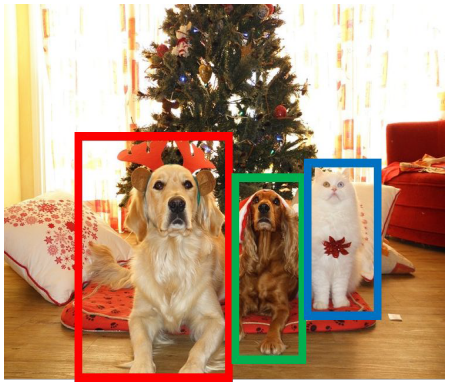
Classification + Localization



CAT

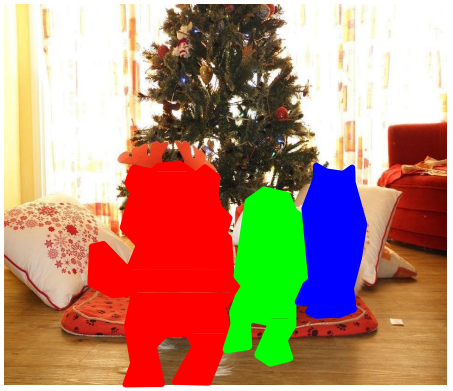
Single Object

Object Detection



DOG, DOG, CAT

Instance Segmentation



DOG, DOG, CAT

Multiple Object

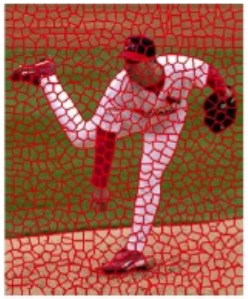
Outline

- Image classification and object recognition
- Image segmentation
- Object detection

Image and object segmentation

- **Image Segmentation**
  - Group pixels into regions that share some similar properties

Superpixels
(Ren ICCV 2003



- **Segmenting images into meaningful objects**
  - Object-level segmentation: accurate localization and recognition

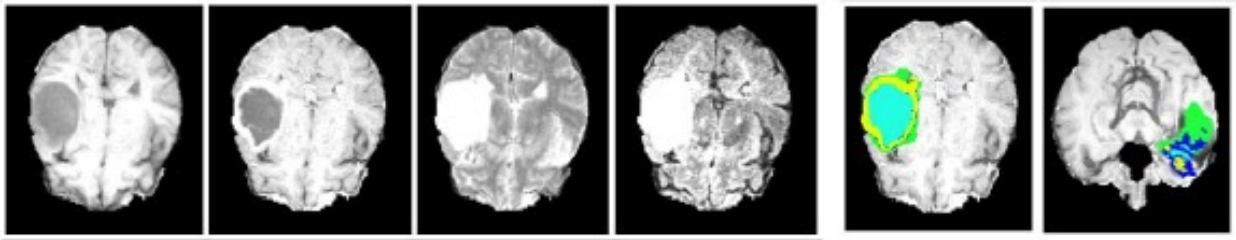
Object segmentation: applications

Image editing and composition (Xu, 2016)

Robotics

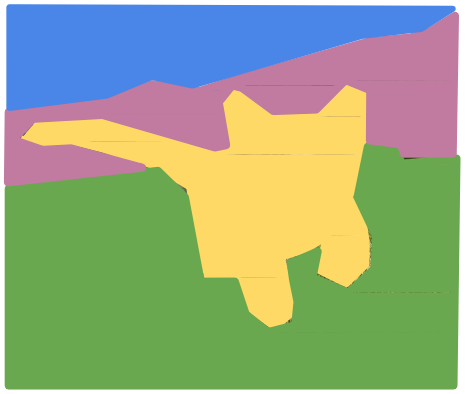
Autonomous driving
(cordts, 2016)

Medical image analysis
(Casamitjana, 2017)



Semantic Segmentation

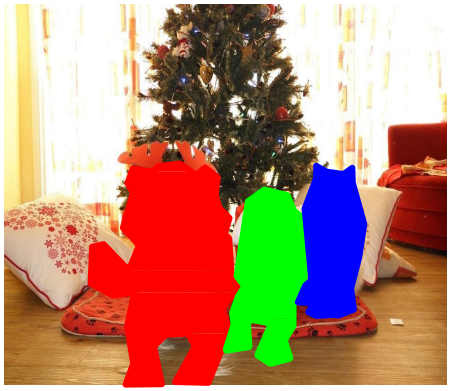
Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

Instance Segmentation

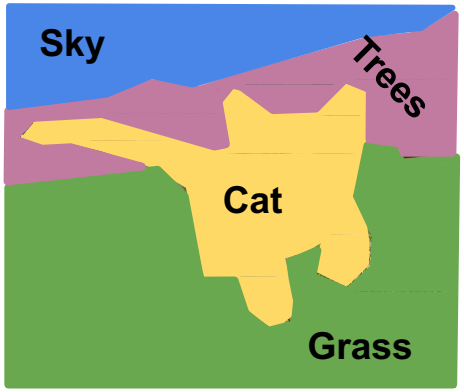


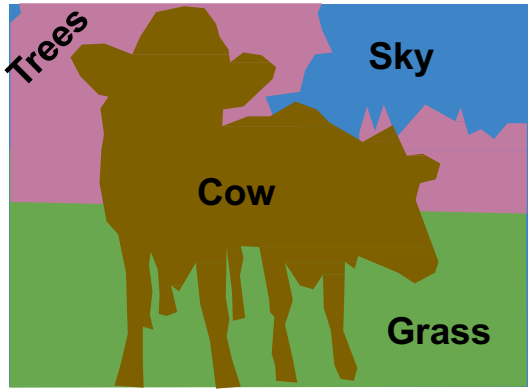
DOG, DOG, CAT

Semantic Segmentation

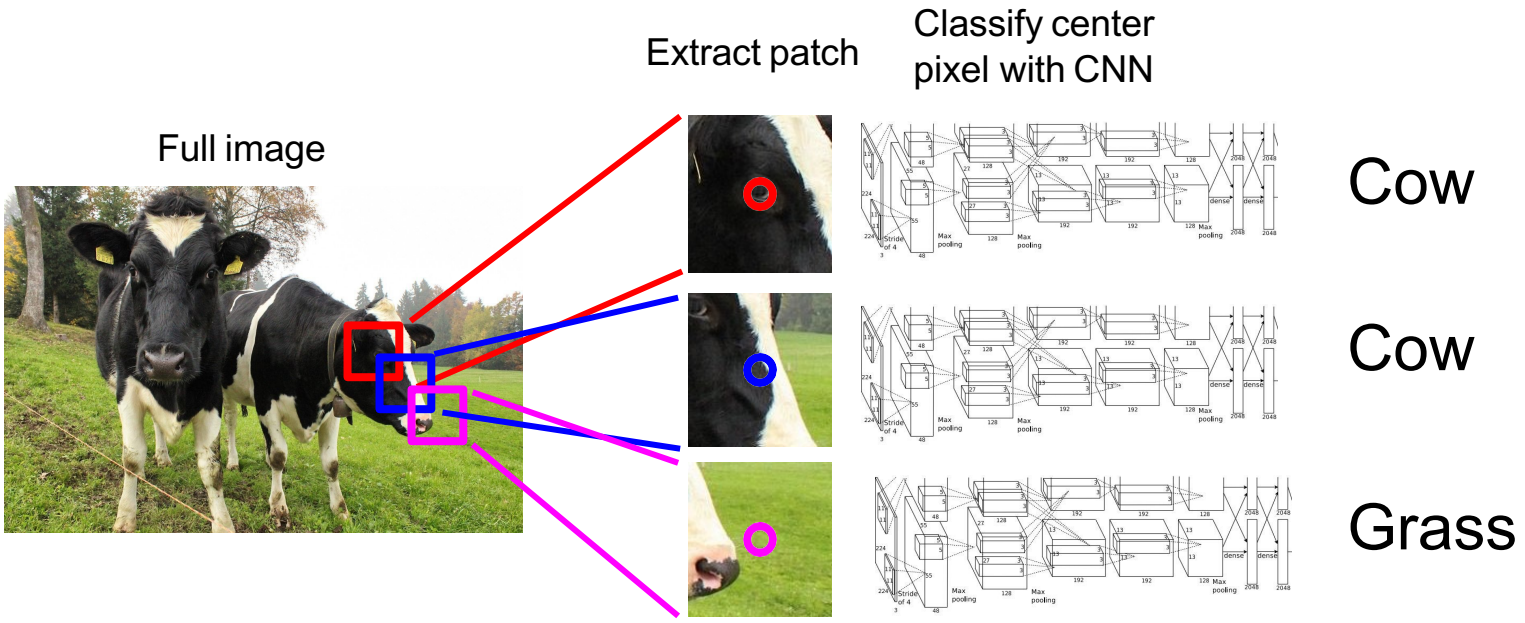
Label each pixel in the image with a category label

Don’t differentiate instances, only care about pixels



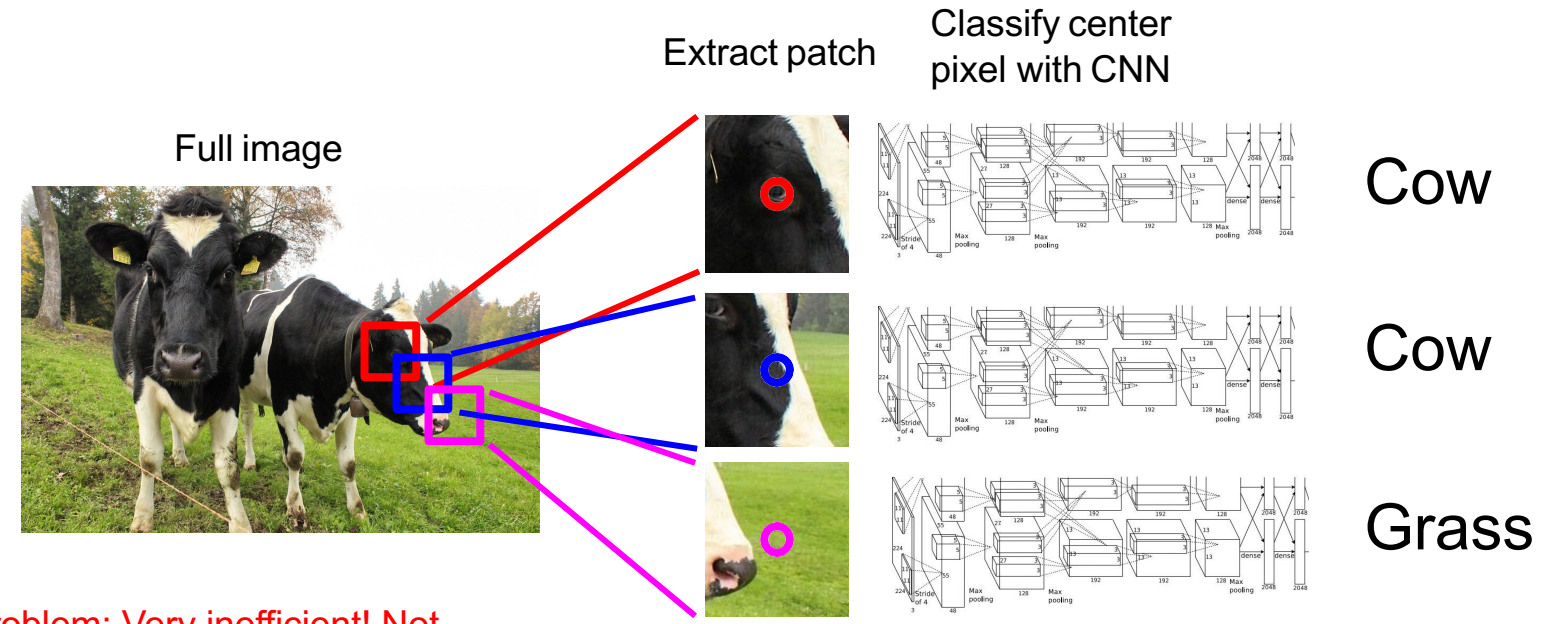


Semantic Segmentation Idea: Sliding Window



Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013
Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014

Semantic Segmentation Idea: Sliding Window



Problem: Very inefficient! Not reusing shared features between overlapping patches

Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013
Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014

Semantic Segmentation Idea: Fully Convolutional

Design a network as a bunch of convolutional layers
to make predictions for pixels all at once!

Input:
3 x H x W

Conv

Conv

Conv

Convolutions:
D x H x W

Conv

Scores:
C x H x W

argmax

Predictions:
H x W

Semantic Segmentation Idea: Fully Convolutional

Design a network as a bunch of convolutional layers to make predictions for pixels all at once!

Input:
3 x H x W

Conv → Conv → Conv

Convolutions:
D x H x W

Problem: convolutions at original image resolution will be very expensive ...

Conv

Scores:
C x H x W

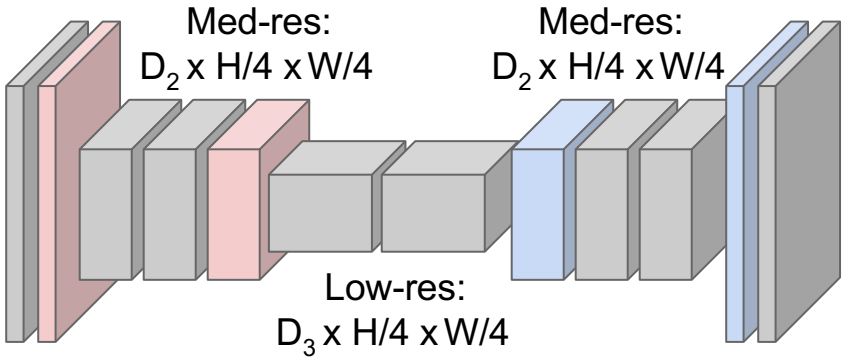
argmax

Predictions:
H x W

Semantic Segmentation Idea: Fully Convolutional

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Input:
3 x H x W



High-res:
D_1 x H/2 x W/2

High-res:
D_1 x H/2 x W/2

Predictions:
H x W

Long, Shelhamer, and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015
Noh et al, “Learning Deconvolution Network for Semantic Segmentation”, ICCV 2015

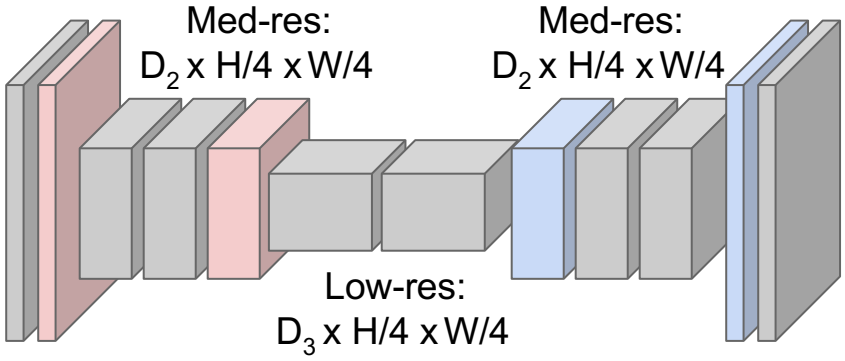
Semantic Segmentation Idea: Fully Convolutional

Downsampling: Pooling, strided convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Upsampling: ???

Input:
3 x H x W



High-res:
D_1 x H/2 x W/2

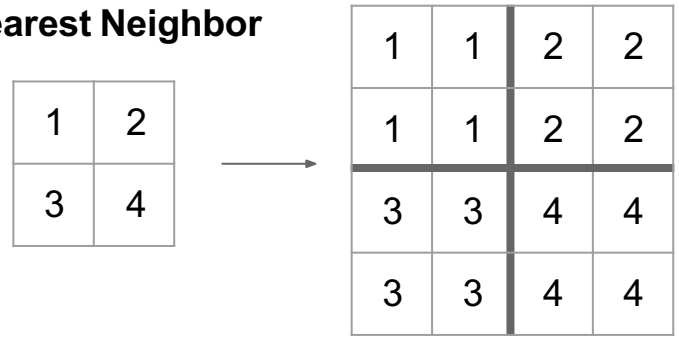
High-res:
D_1 x H/2 x W/2

Predictions:
H x W

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

In-Network upsampling

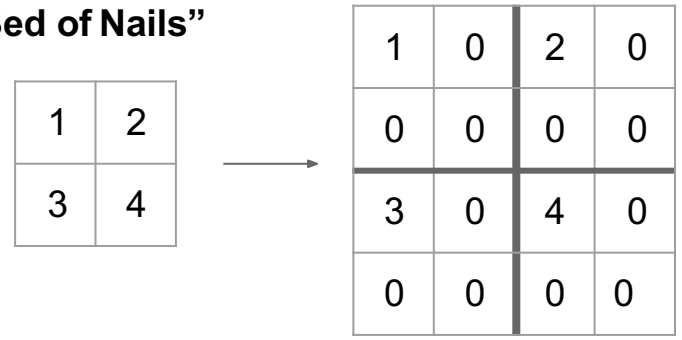
Nearest Neighbor



Input: 2 x 2

Output: 4 x 4

“Bed of Nails”



Input: 2 x 2

Output: 4 x 4

In-Network upsampling: “Max Unpooling”

Max Pooling
Remember which element was max!

1	2	6	3
3	5	2	1
1	2	2	1
7	3	4	8

Input: 4 x 4

5	6
7	8

Output: 2 x 2

Rest of the network

Max Unpooling
Use positions from pooling layer

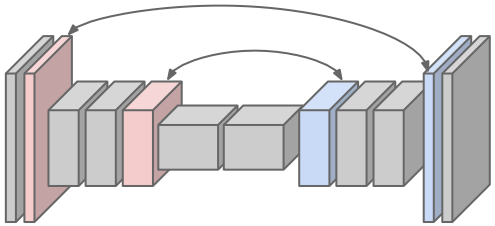
1	2
3	4

Input: 2 x 2

0	0	2	0
0	1	0	0
0	0	0	0
3	0	0	4

Output: 4 x 4

Corresponding pairs of downsampling and upsampling layers



In-Network upsampling

- **Other Solution:**
  - Deconvolution (upsampling followed by specifically designed convolution)

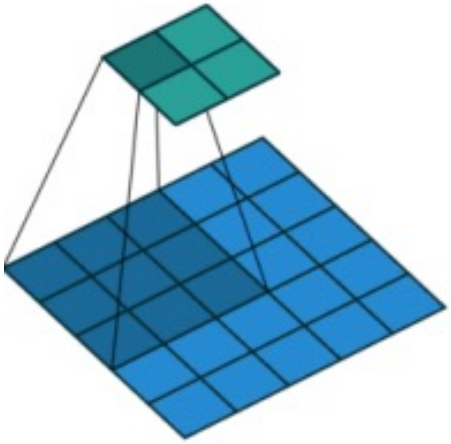


Figure 1: Normal Convolution

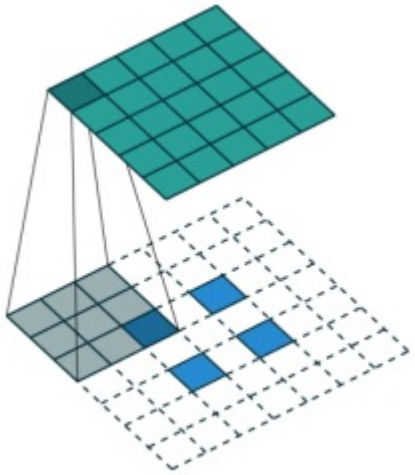
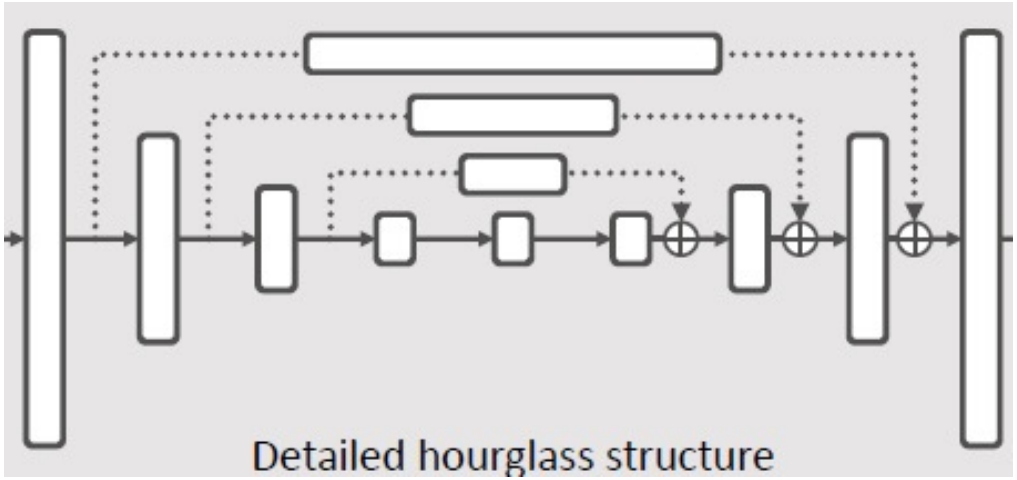
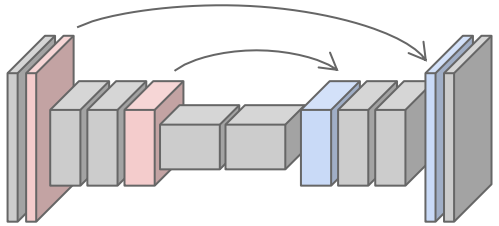


Figure 2: Transposed convolution.

- **Other Solution:**
  - Deconvolution (upsampling followed by specifically designed convolution)
  - Take the features in previous shallow layers into consideration (skip connections)





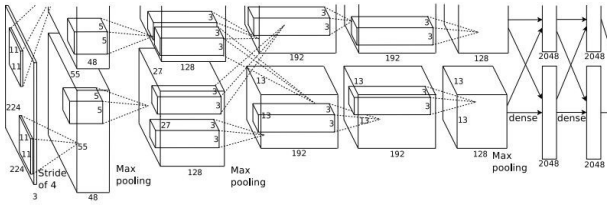
Outline

- Image classification and object recognition
- Image segmentation
- Object detection

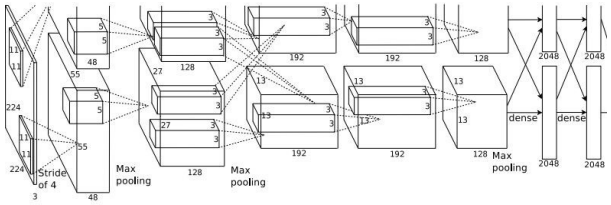
Demos

- https://www.youtube.com/watch?v=VlH3OEhZnow
- https://www.youtube.com/watch?v=nus5-4cZr7c
- https://www.youtube.com/watch?v=nDqnMpE6b8s
- https://www.youtube.com/watch?v=RCM0u2tBl5E
- https://www.youtube.com/watch?v=BtYMOOrBb2E

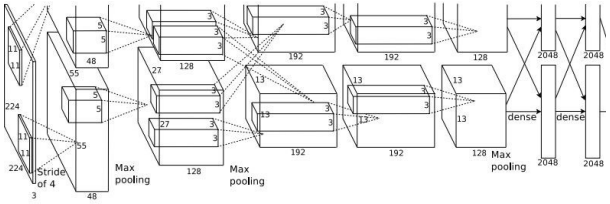
Object Detection as Regression?



CAT: (x, y, w, h)



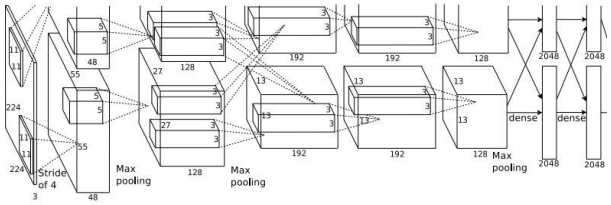
DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)



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

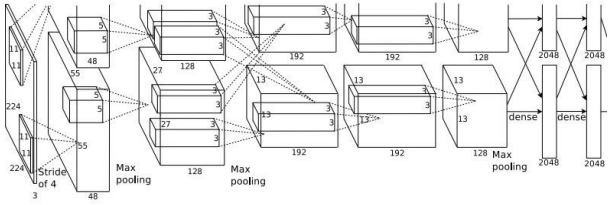
Object Detection as Regression?

Each image needs a different number of outputs!



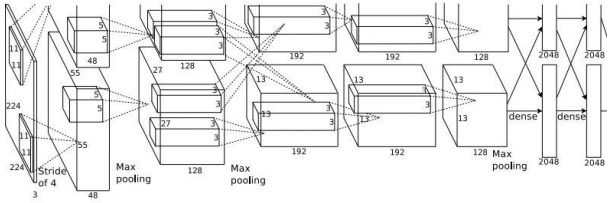
CAT: (x, y, w, h)

4 numbers



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

12 numbers

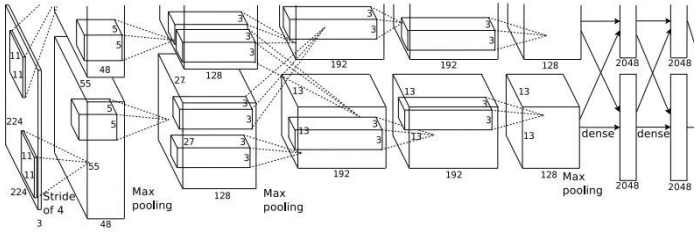


DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
….

Many numbers!

Object Detection as Classification: Sliding Window

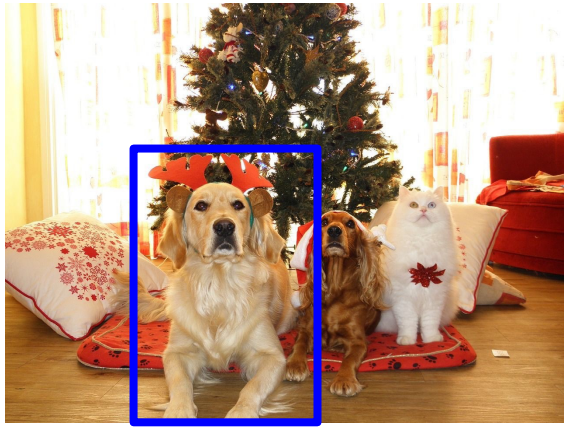
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

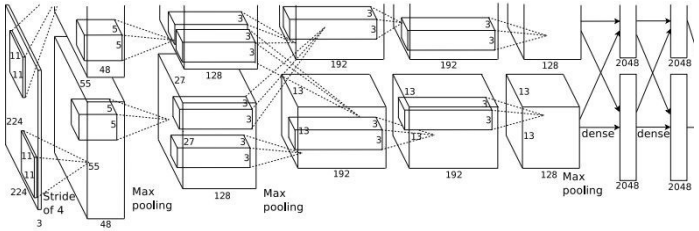


Dog? NO
Cat? NO
Background? YES

Object Detection as Classification: Sliding Window

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

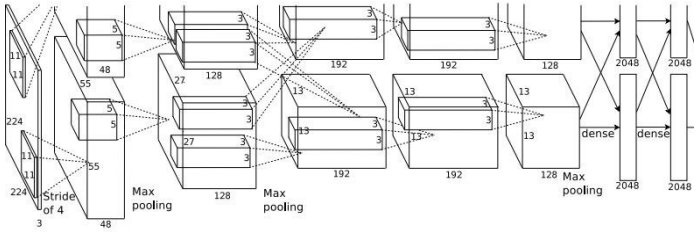




Dog? YES
Cat? NO
Background? NO

Object Detection as Classification: Sliding Window

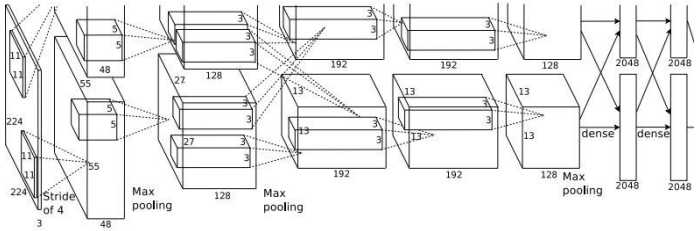
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

Object Detection as Classification: Sliding Window

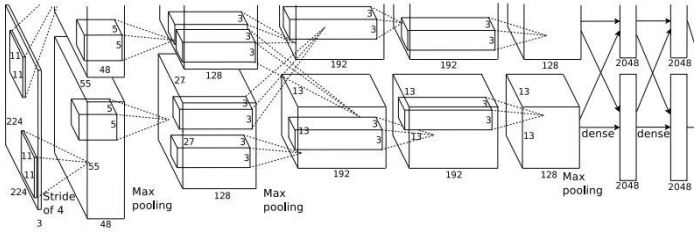
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



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

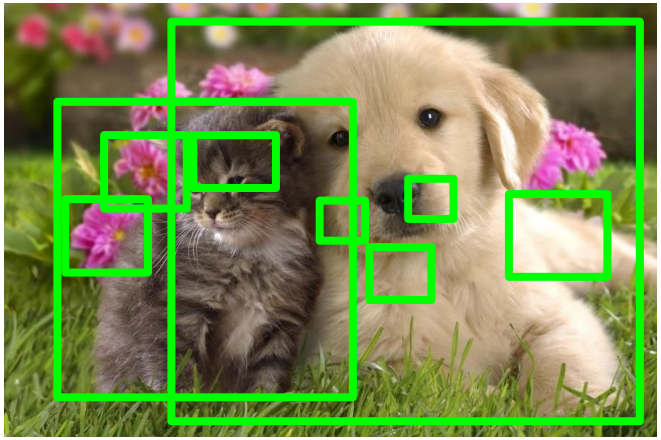


Dog? NO
Cat? YES
Background? NO

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

Region Proposals

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



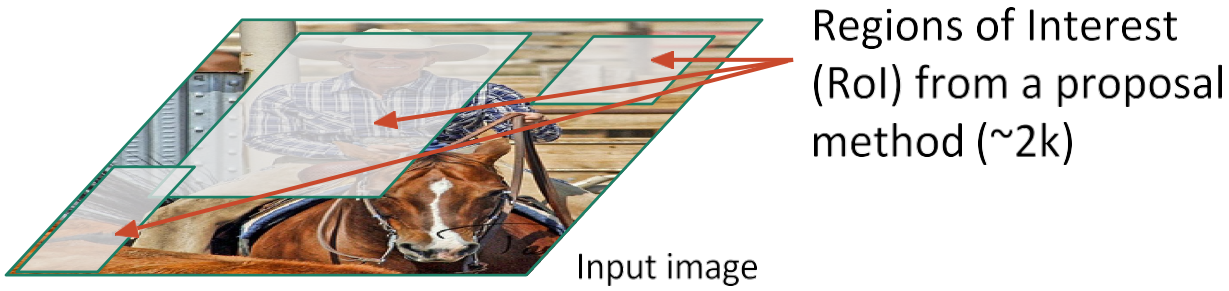
Alexe et al, “Measuring the objectness of image windows”, TPAMI 2012
Uijlings et al, “Selective Search for Object Recognition”, IJCV 2013
Cheng et al, “BING: Binarized normed gradients for objectness estimation at 300fps”, CVPR 2014
Zitnick and Dollar, “Edge boxes: Locating object proposals from edges”,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

R-CNN

Input image

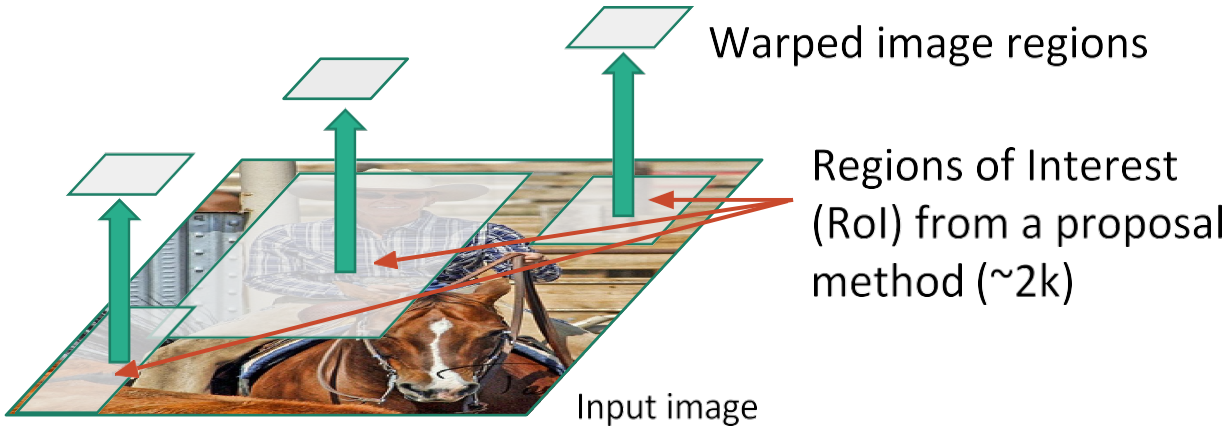
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.

R-CNN



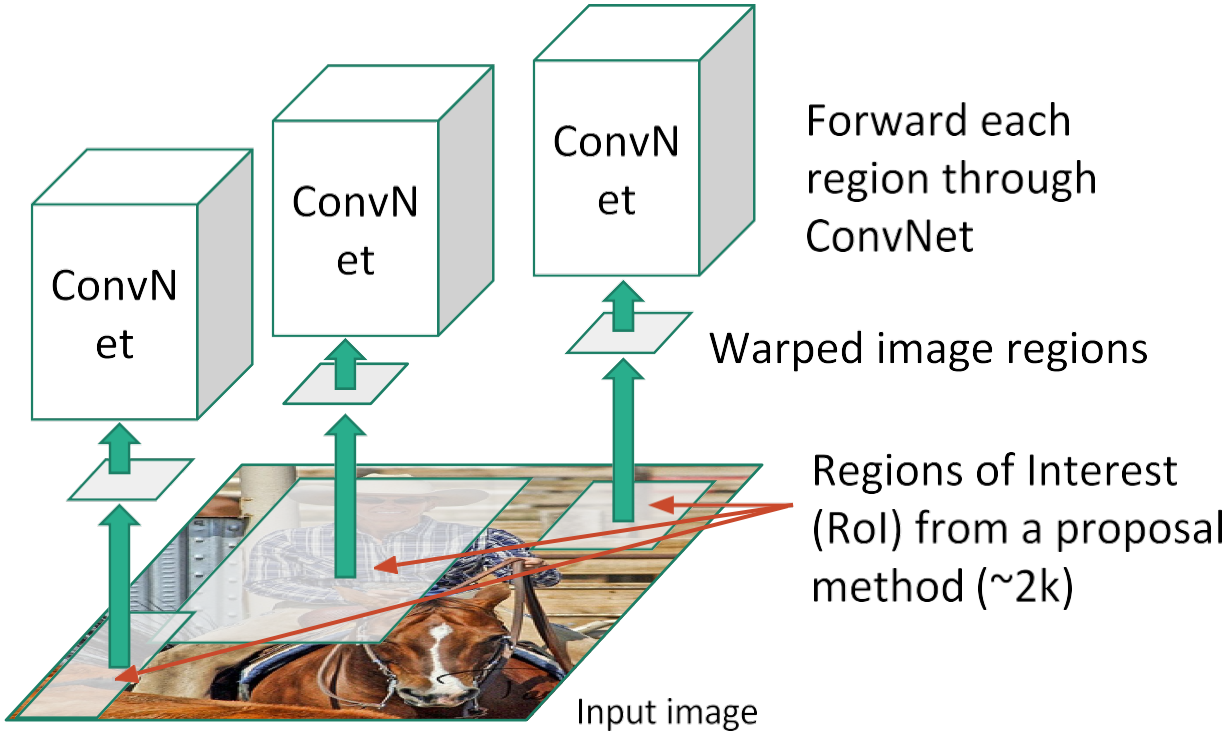
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

R-CNN



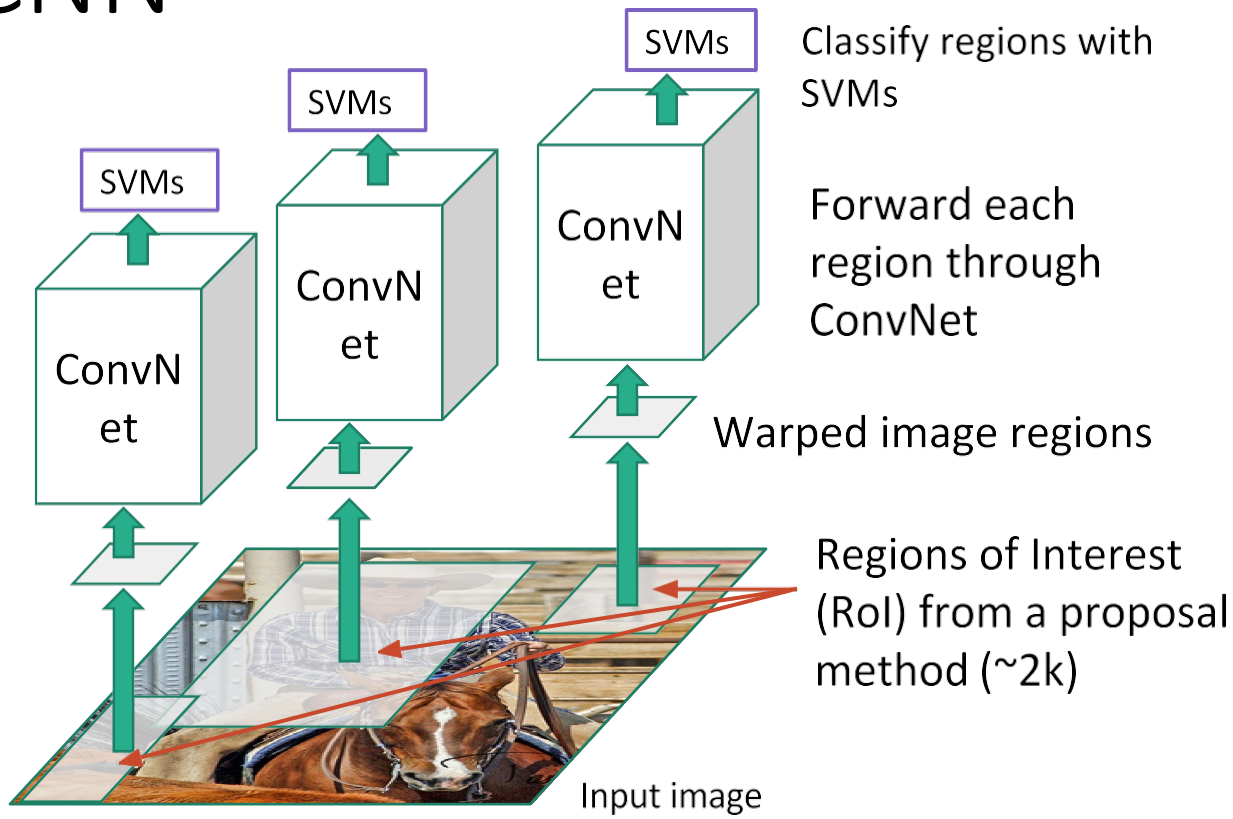
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

R-CNN



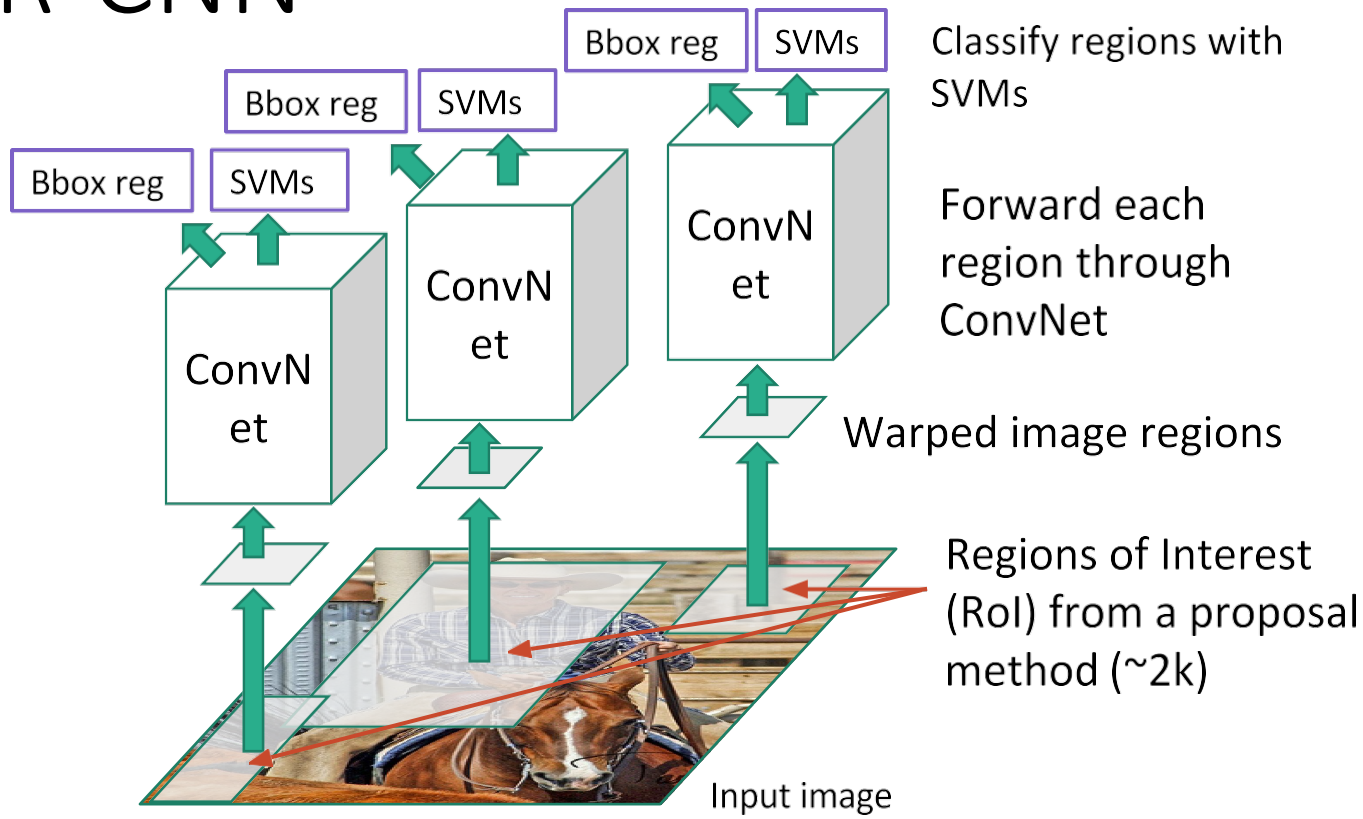
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.

R-CNN



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

R-CNN



Linear Regression for bounding box offsets

Classify regions with SVMs

Forward each region through ConvNet

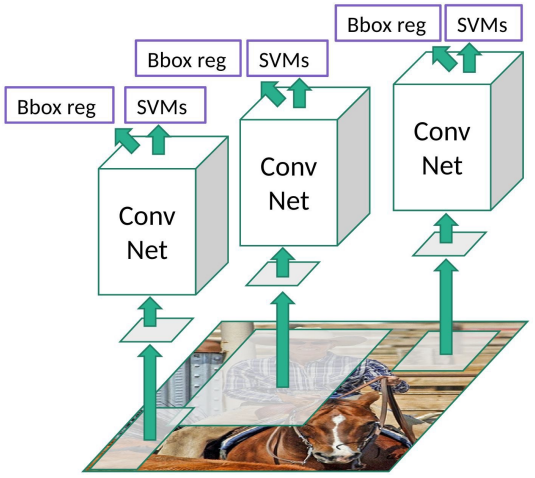
Warped image regions

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: 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]



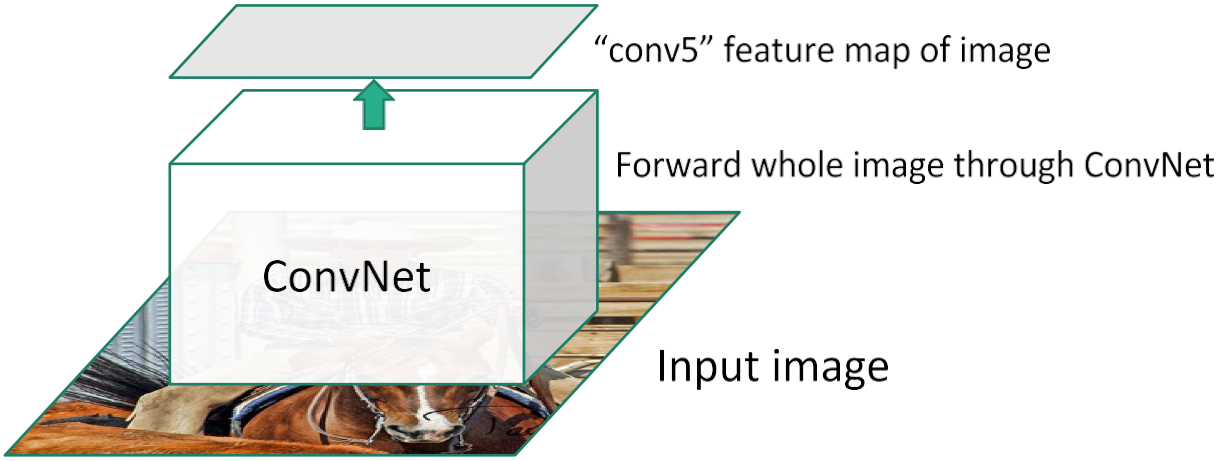
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.

Fast R-CNN

Input image

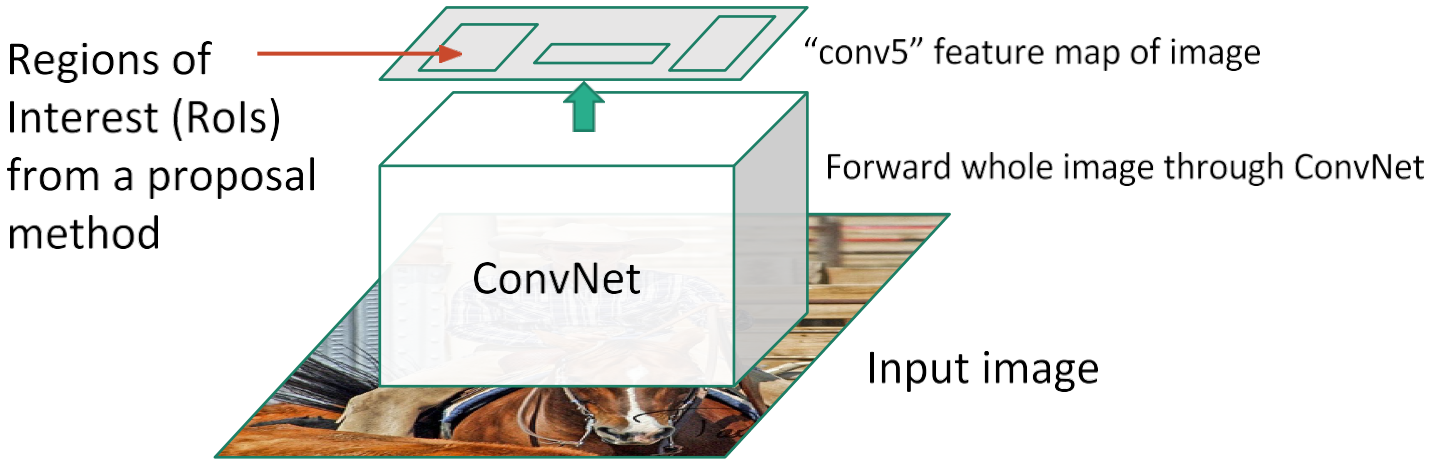
Girshick, “Fast R-CNN”, ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Fast R-CNN



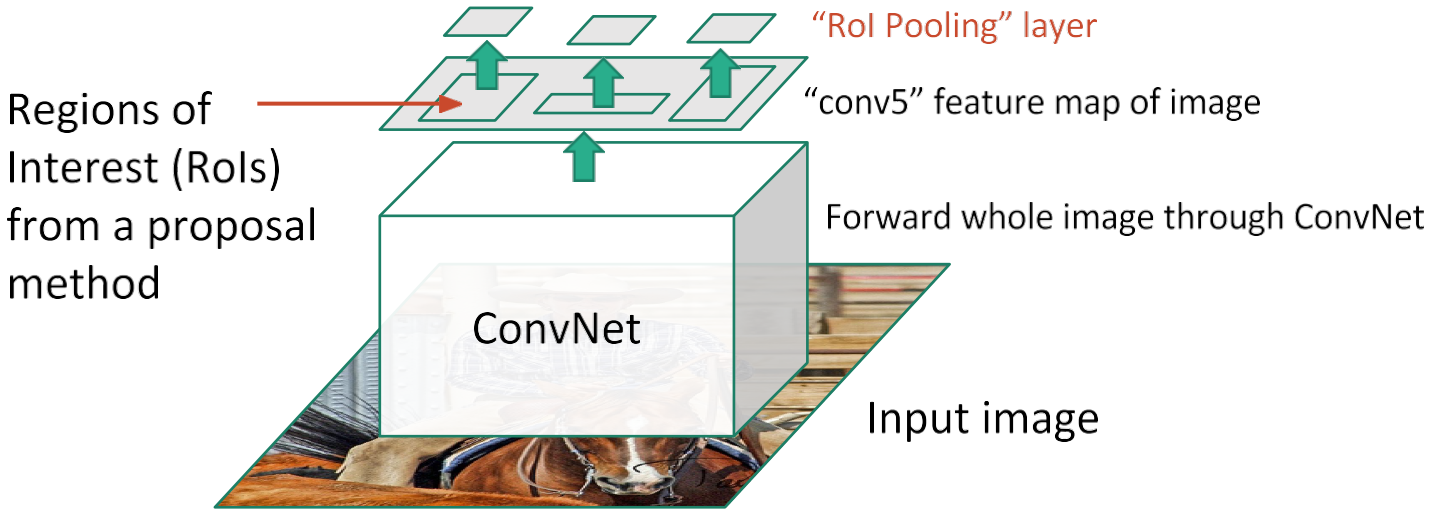
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

Fast R-CNN



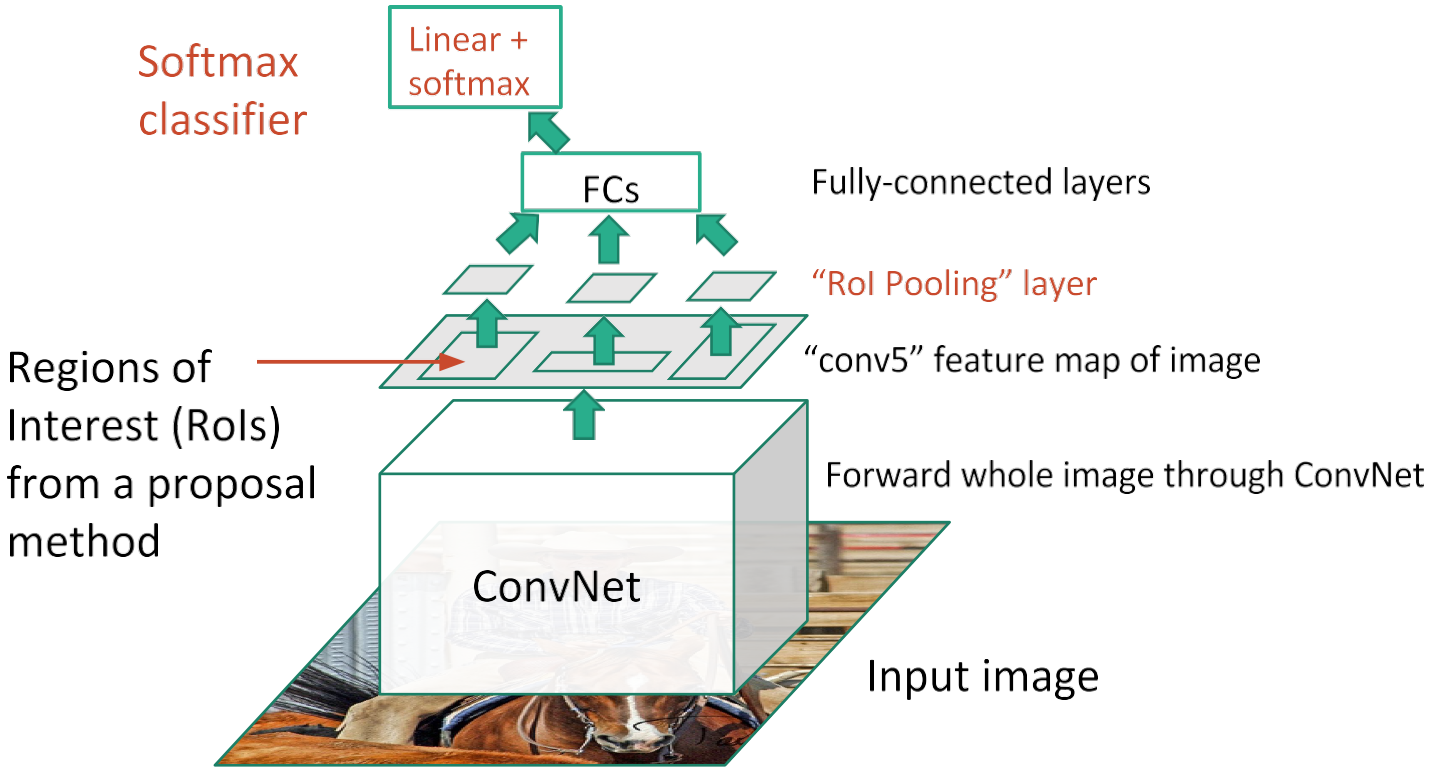
Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Fast R-CNN



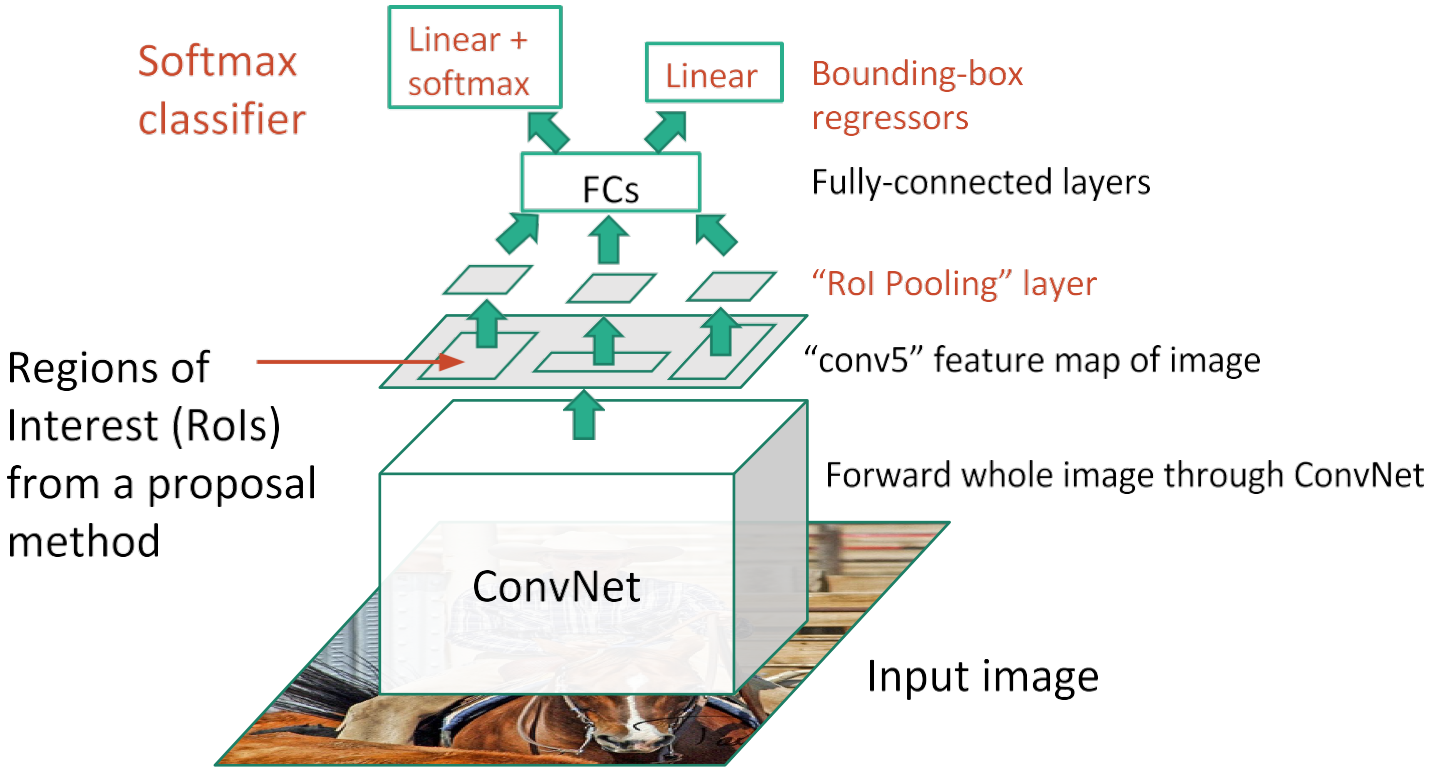
Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Fast R-CNN



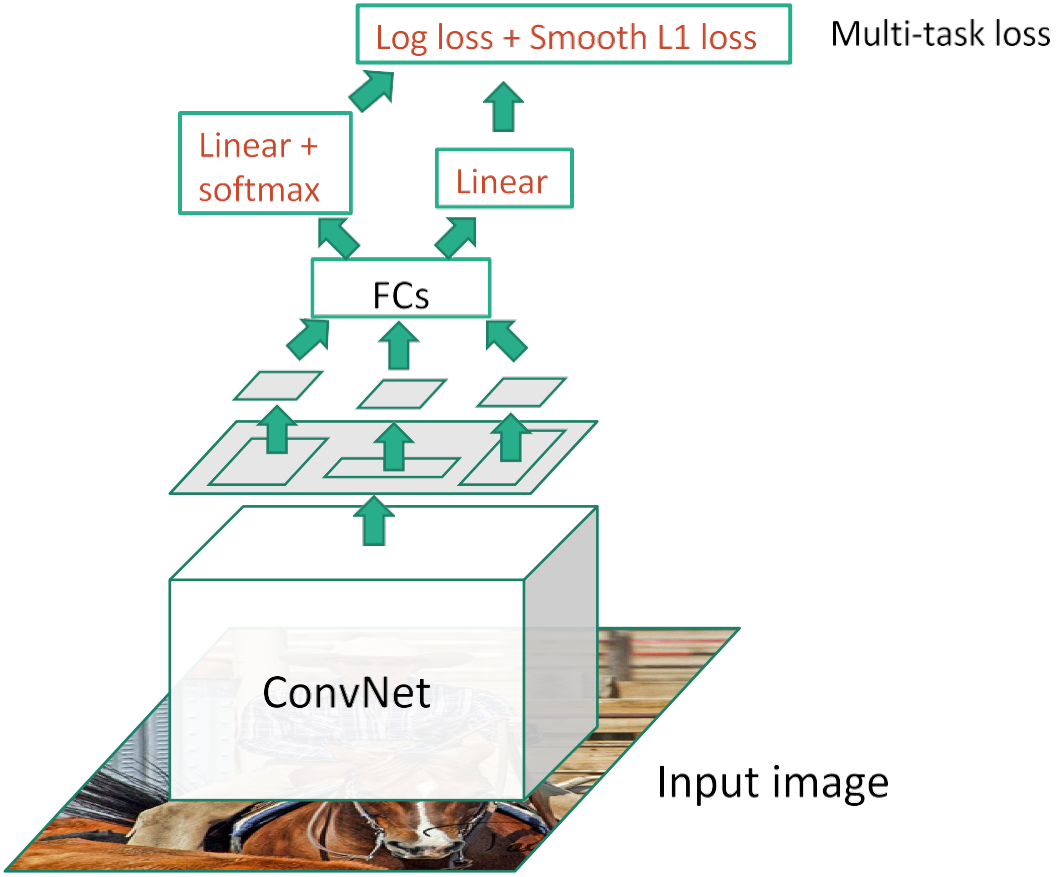
Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Fast R-CNN



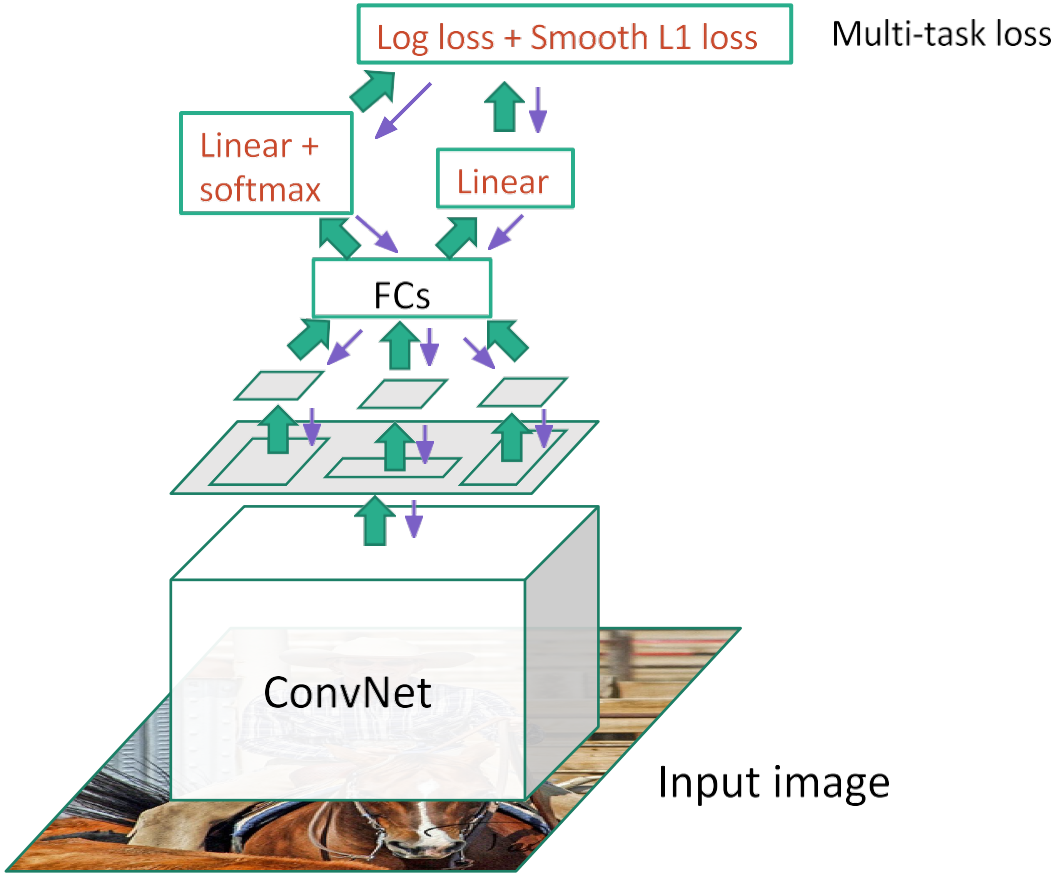
Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Fast R-CNN (Training)



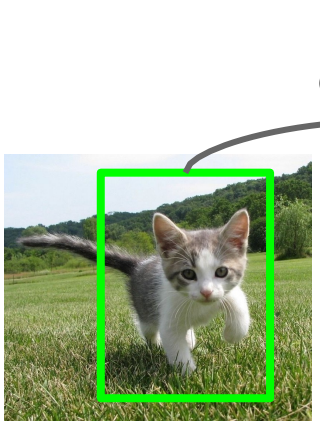
Girshick, “Fast R-CNN”, ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Fast R-CNN (Training)



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

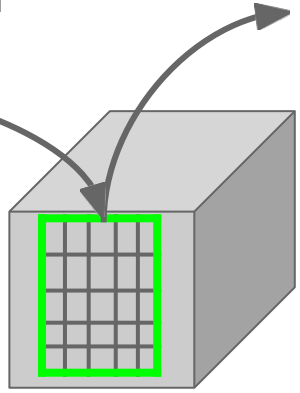
Fast R-CNN: RoI Pooling



Hi-res input image:
3 x 640 x 480
with region
proposal

Project proposal
onto features

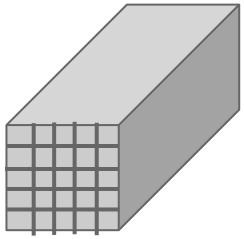
CNN



Hi-res conv features:
512 x 20 x 15;

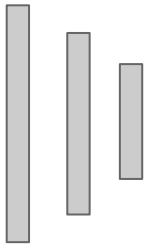
Projected region
proposal is e.g.
512 x 18 x 8
(varies per proposal)

Divide projected
proposal into 7x7
grid, max-pool
within each cell



RoI conv features:
512 x 7 x 7
for region proposal

Fully-connected
layers

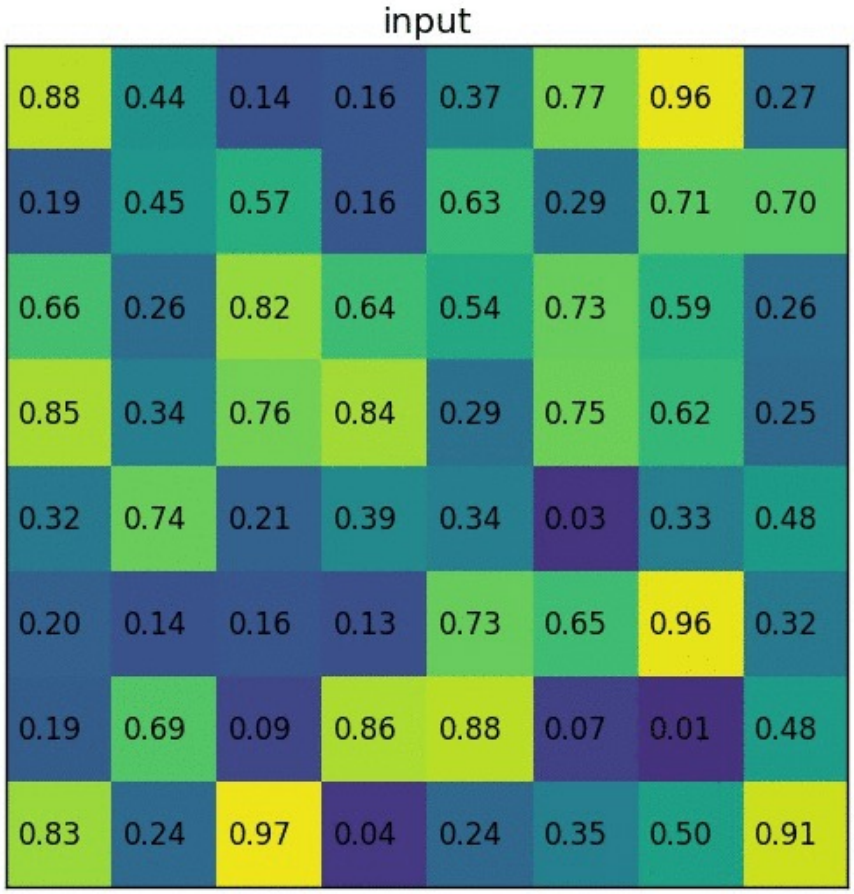


Fully-connected layers expect
low-res conv features:
512 x 7 x 7

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

Fast R-CNN: RoI Pooling

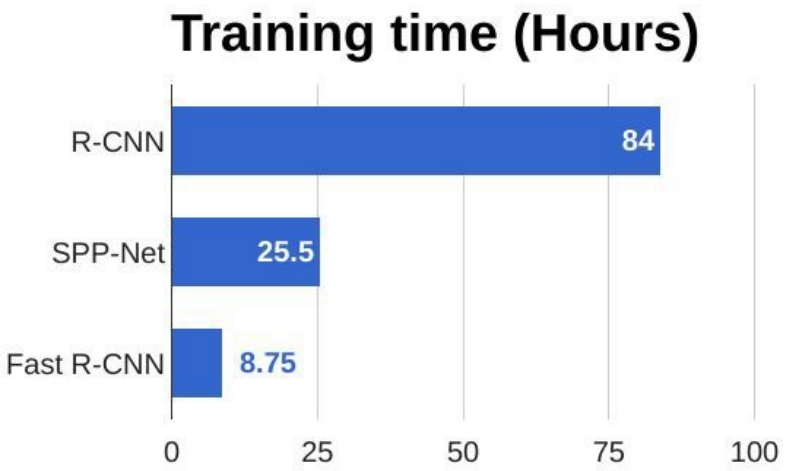
Output Size: 2x2

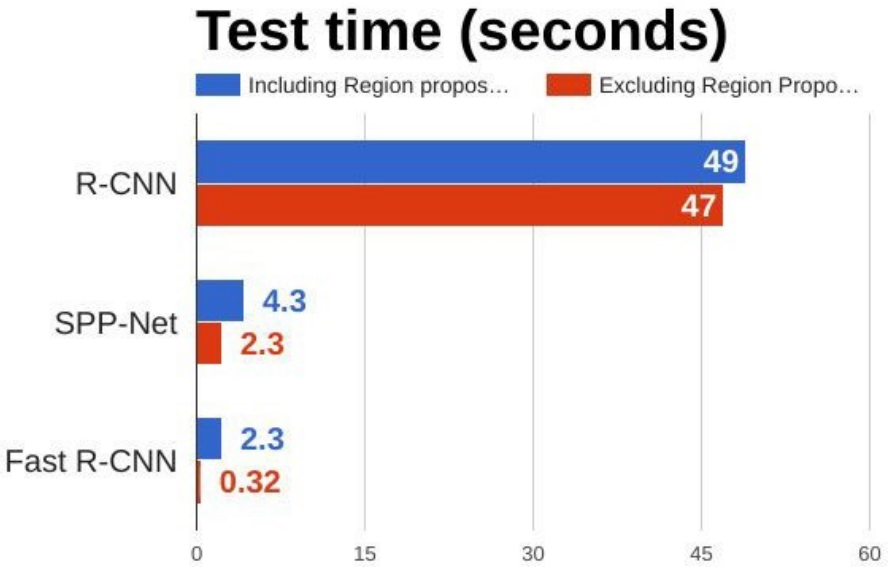


Fast R-CNN vs R-CNN

- passing one instead of 2,000 regions per image to the ConvNet
- using one instead of three different models for extracting features, classification and generating bounding boxes.

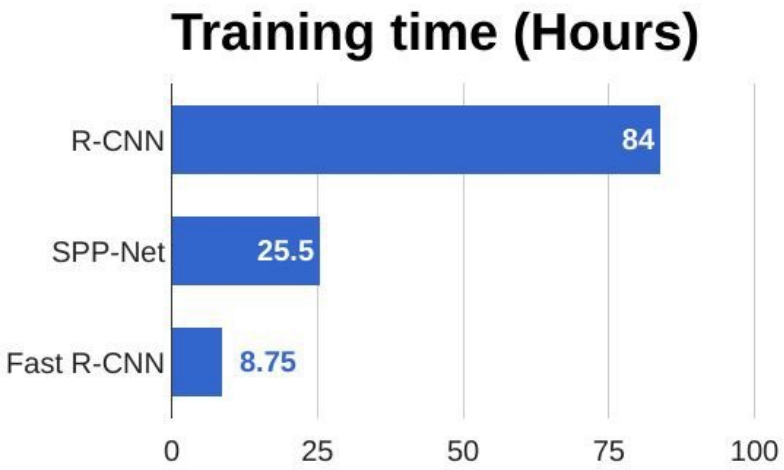
R-CNN vs SPP vs Fast R-CNN

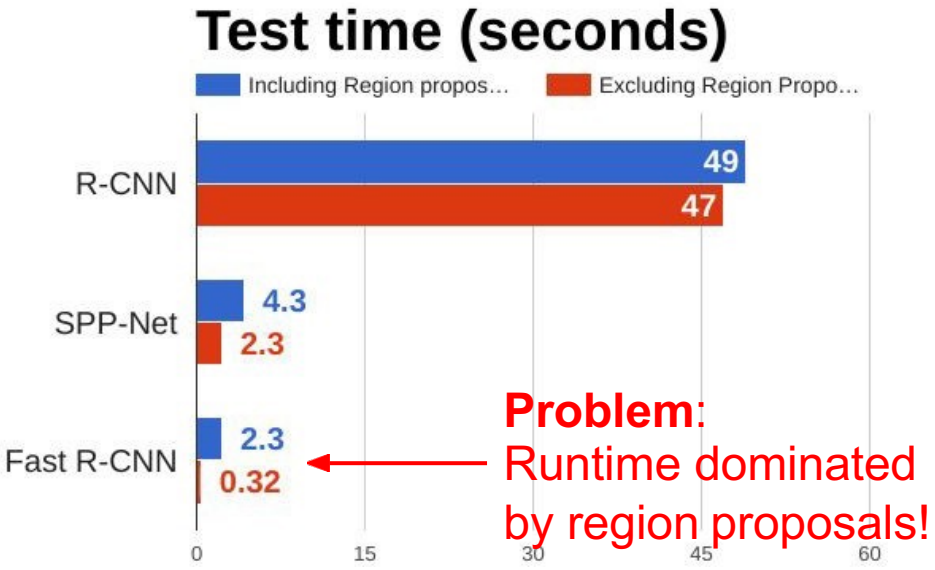




Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
He et al, “Spatial pyramid pooling in deep convolutional networks for visual recognition”, ECCV 2014
Girshick, “Fast R-CNN”, ICCV 2015

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

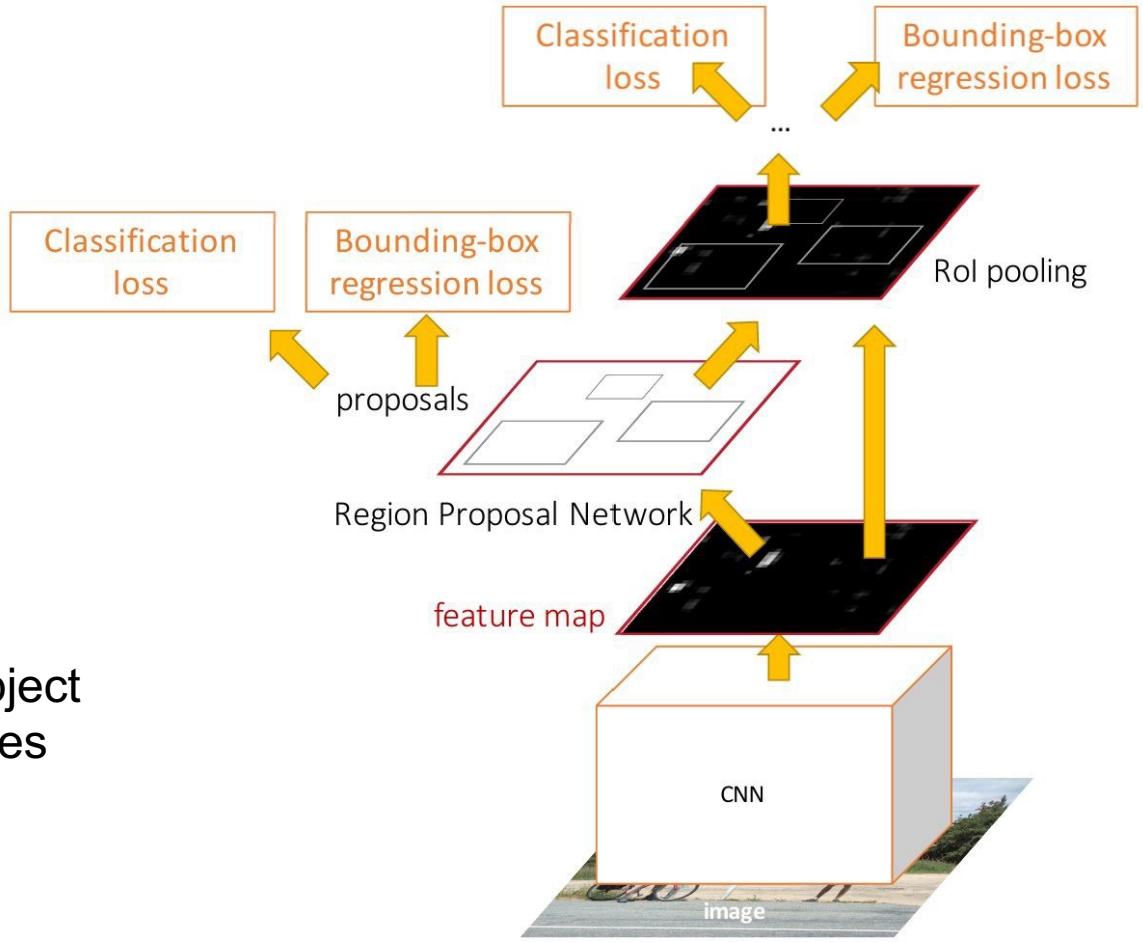
Faster R-CNN:

Make CNN do proposals!

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

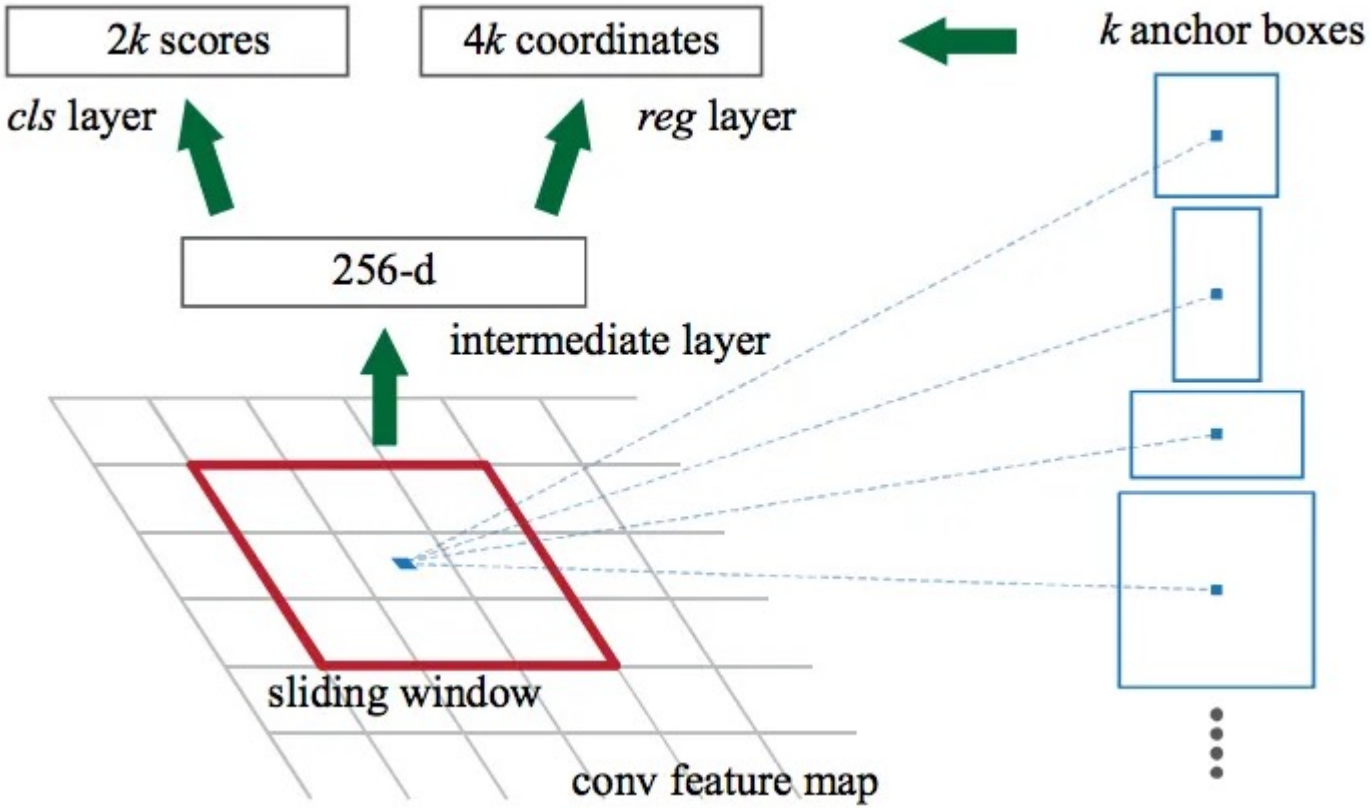
Jointly train with 4 losses:

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



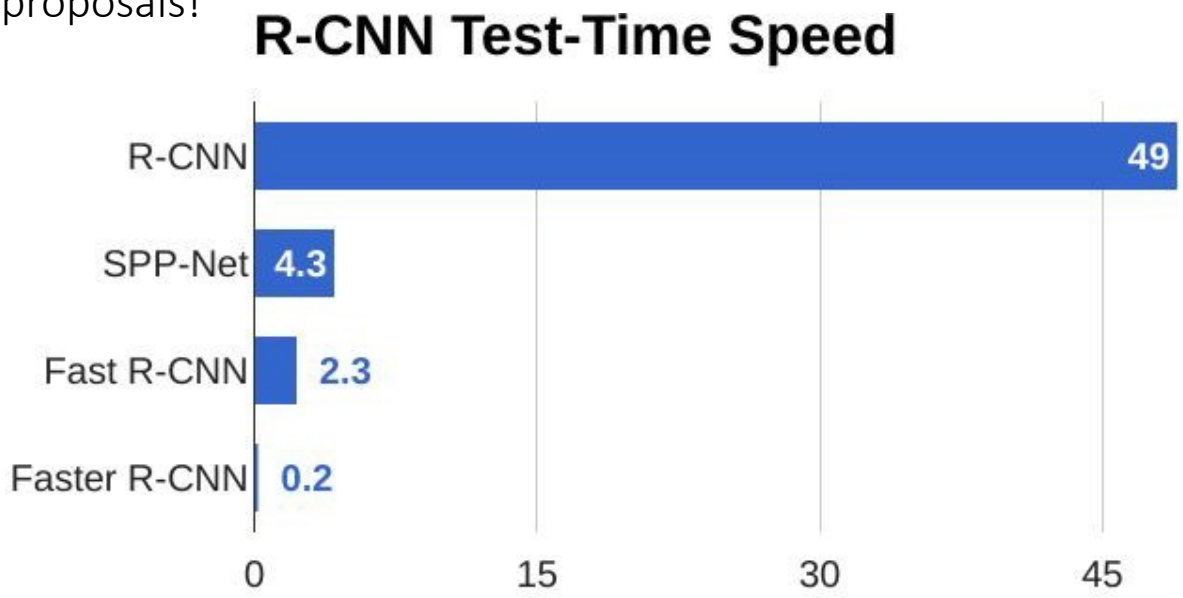
Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015 Figure copyright 2015, Ross Girshick; reproduced with permission

Faster R-CNN: Region Proposal Network



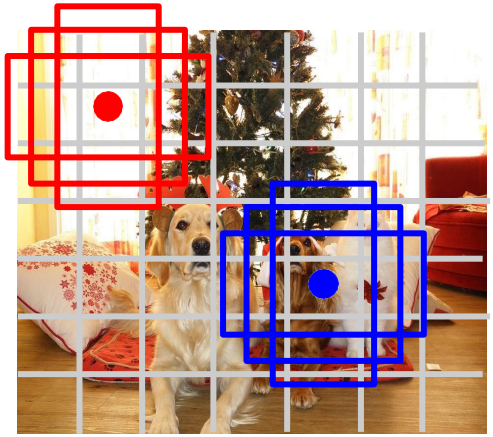
Fast**er** R-CNN:

Make CNN do proposals!



Detection without Proposals: YOLO / SSD

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)

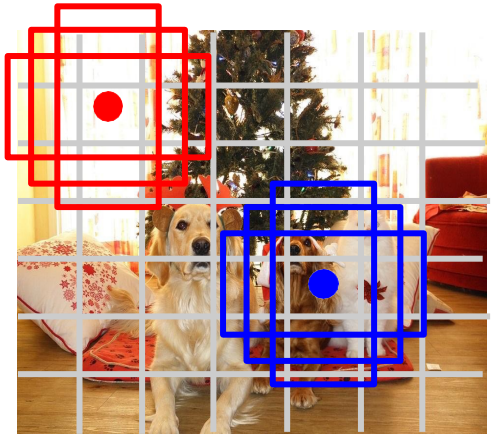
Output:
7 x 7 x (5 * B + C)

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

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)

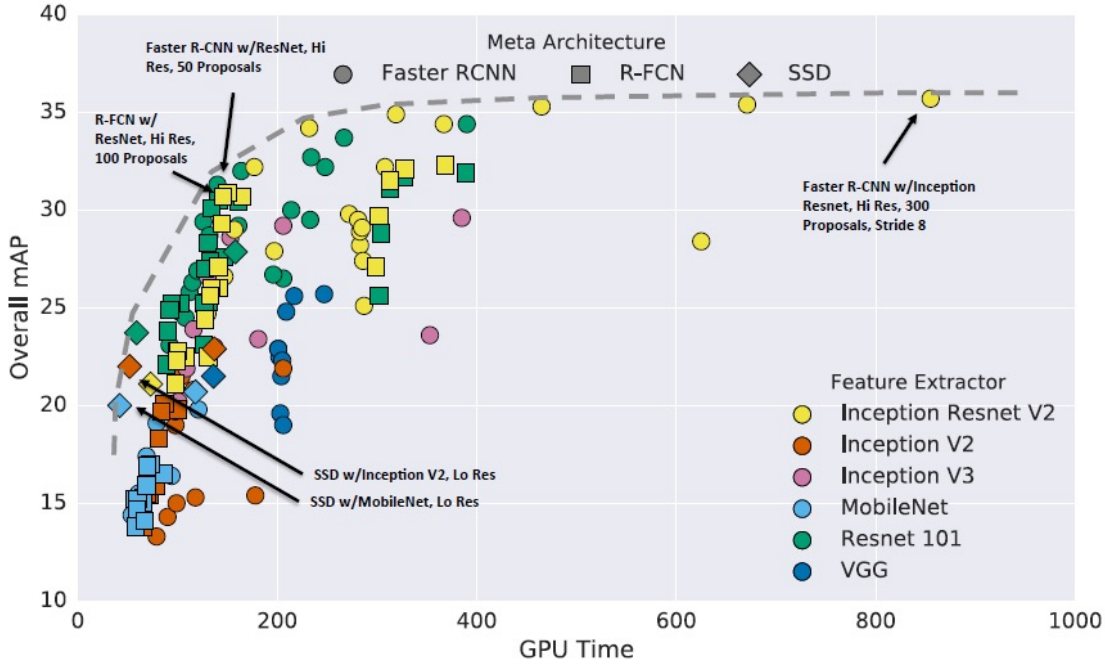
Redmon et al, “You Only Look Once: Unified, Real-Time Object Detection”, CVPR 2016
Liu et al, “SSD: Single-Shot MultiBox Detector”, ECCV 2016

Object Detection: Lots of variables …

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

Object Detection architecture
Faster R-CNN
R-FCN
SSD

Image Size
 # Region Proposals
…

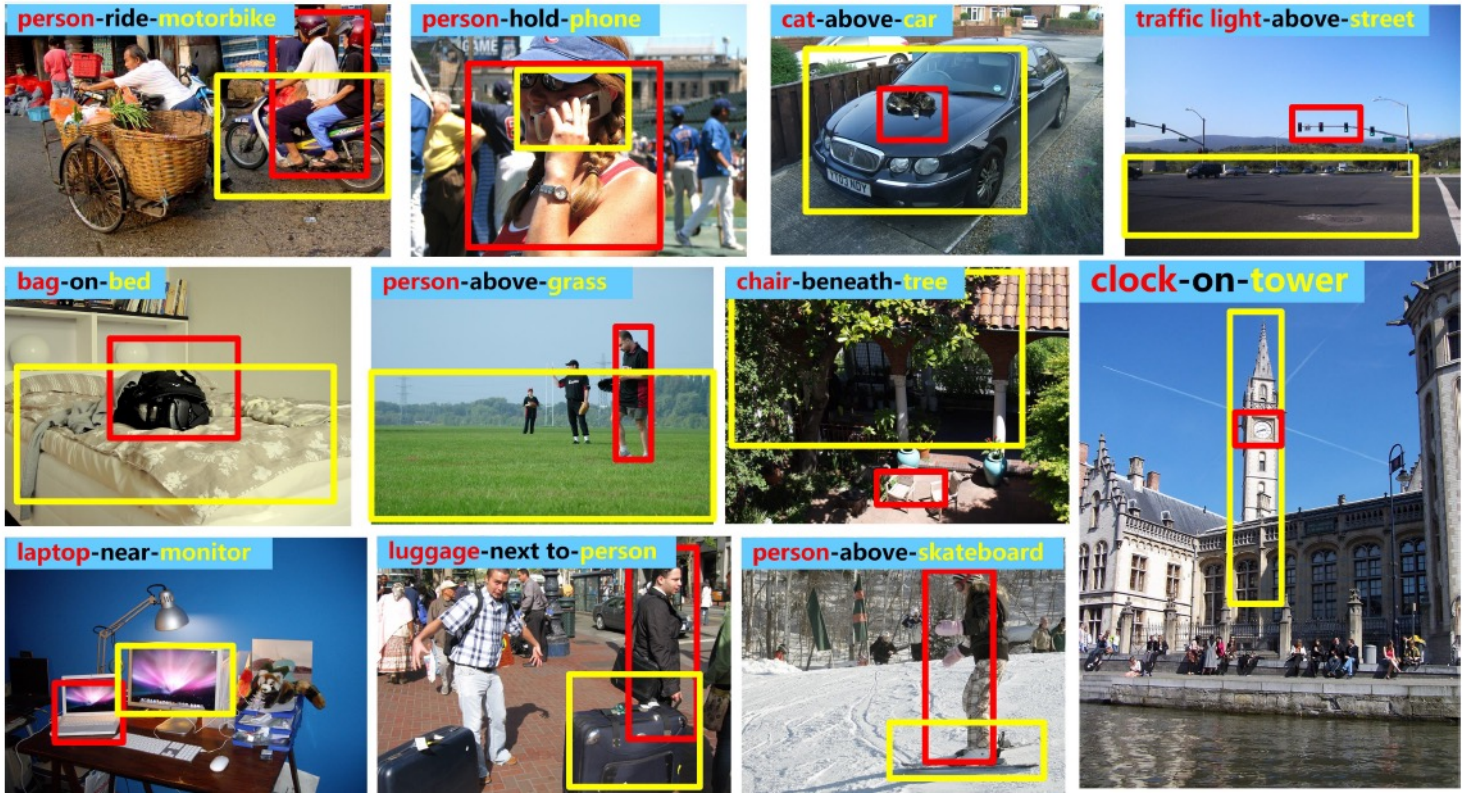


Huang et al, “Speed/accuracy trade-offs for modern convolutional object detectors”, CVPR 2017

R-FCN: Dai et al, “R-FCN: Object Detection via Region-based Fully Convolutional Networks”, NIPS 2016
Inception-V2: Ioffe and Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, ICML 2015
Inception V3: Szegedy et al, “Rethinking the Inception Architecture for Computer Vision”, arXiv 2016
Inception ResNet: Szegedy et al, “Inception-V4, Inception-ResNet and the Impact of Residual Connections on Learning”, arXiv 2016
MobileNet: Howard et al, “Efficient Convolutional Neural Networks for Mobile Vision Applications”, arXiv 2017

What can we do with detection results?

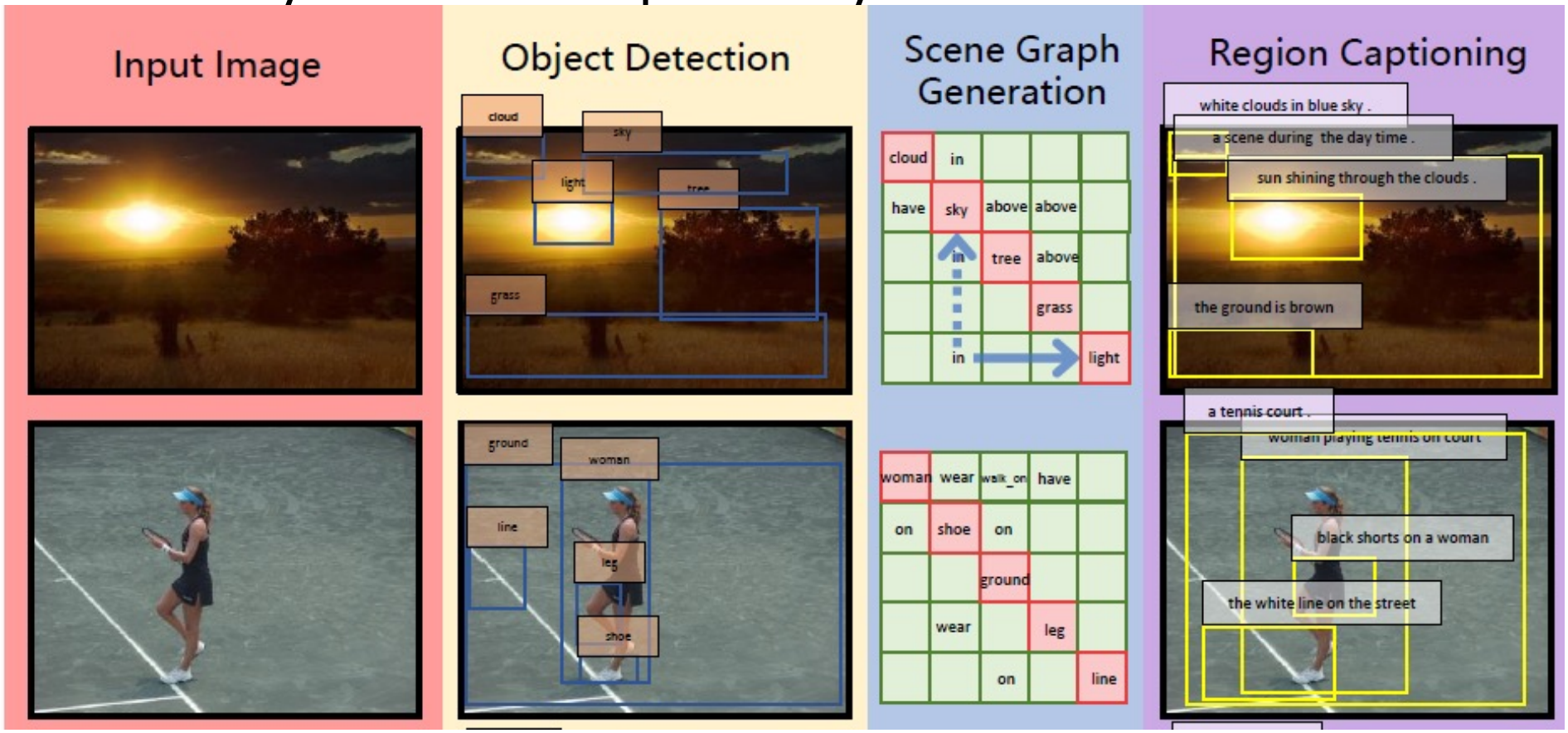
- Analyse relationship



Yikang LI , **W. Ouyang** , Xiaogang Wang. "ViP-CNN: A Visual Phrase Reasoning Convolutional Neural Network for Visual Relationship Detection", *Proc. CVPR* , 2017.
Yikang Li, Wanli Ouyang, Bolei Zhou, Yanwen Cui, Jianping Shi, Chao Zhang, Xiaogang Wang. "Factorizable Net: An Efficient Subgraph-based Framework for Scene Graph Generation" Proc. ECCV, 2018.

What can we do with detection results?

- Analyse relationship and say some words



Yikang Li, **Wanli Ouyang**, Bolei Zhou, Kun Wang, Xiaogang Wang. "Scene Graph Generation from Objects, Phrases and Region Captions", Proc. ICCV, 2017.