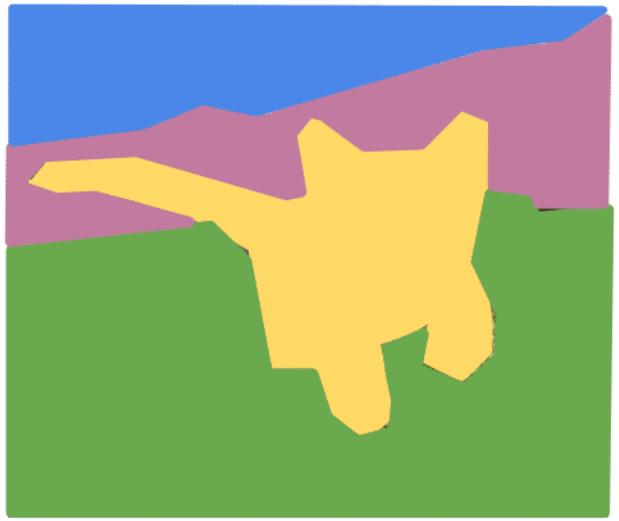
Компьютерное зрение

Лекция 3 – Сегментация и распознавание

2021

Other Computer Vision Tasks

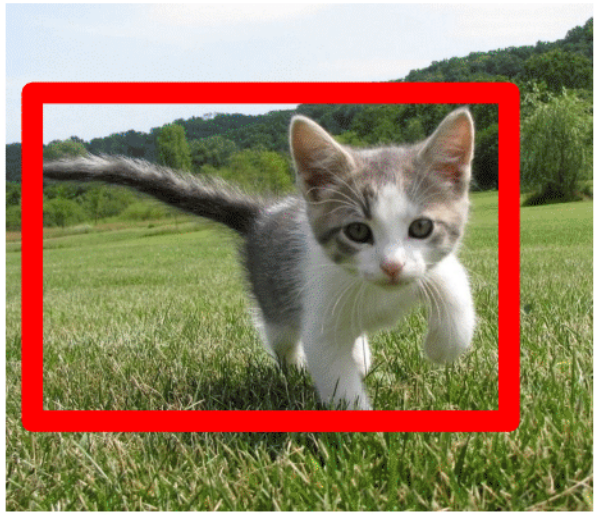
Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

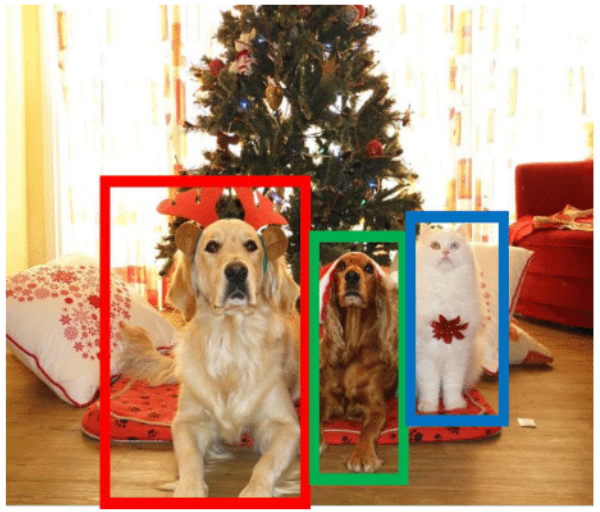
Classification + Localization



CAT

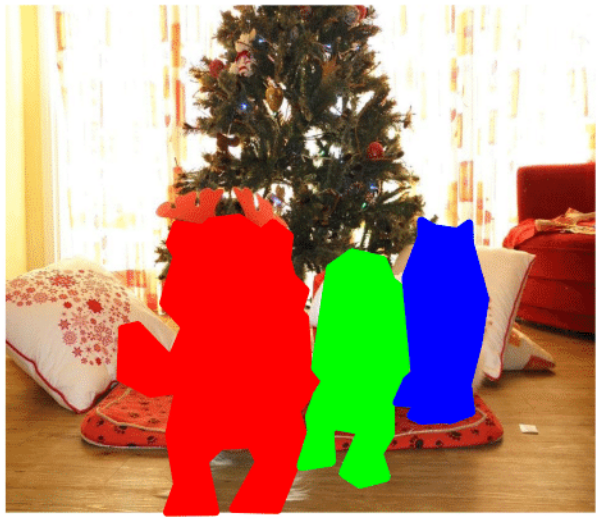
Single Object

Object Detection



DOG, DOG, CAT

Instance Segmentation

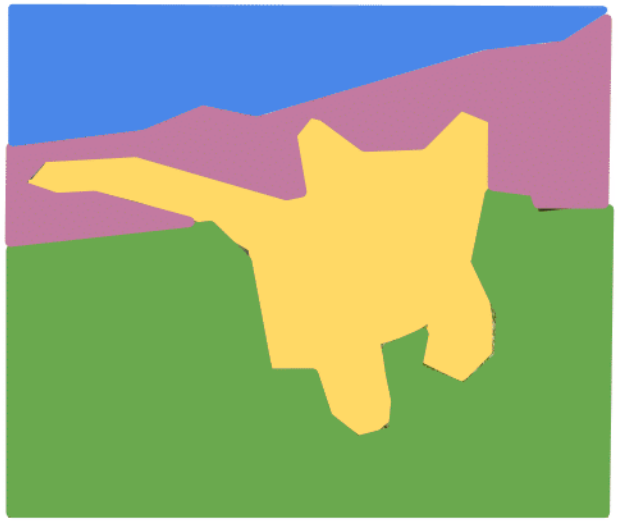


DOG, DOG, CAT

Multiple Object

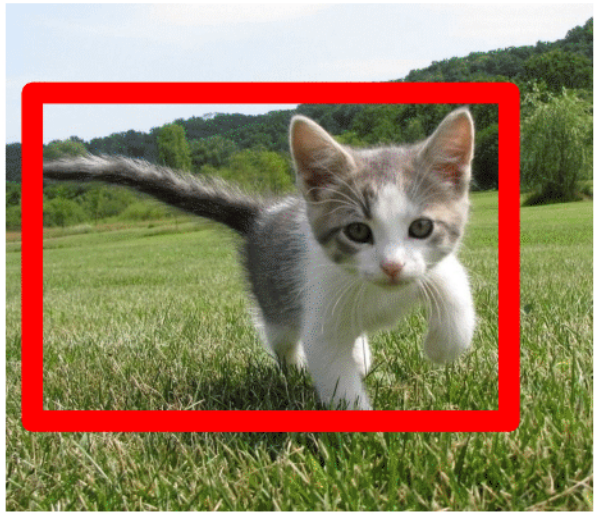
Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Semantic Segmentation



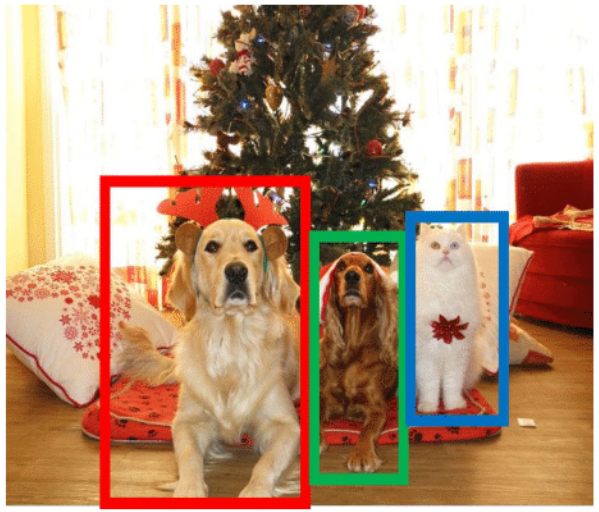
GRASS, CAT, TREE, SKY

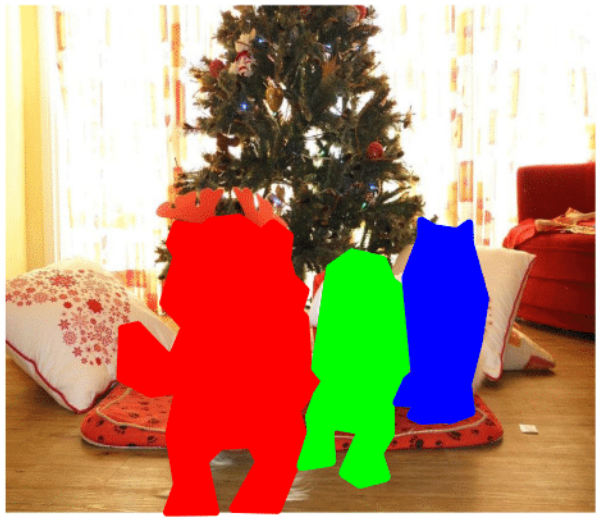
No objects, just pixels



CAT

Single Object





DOG, DOG, CAT

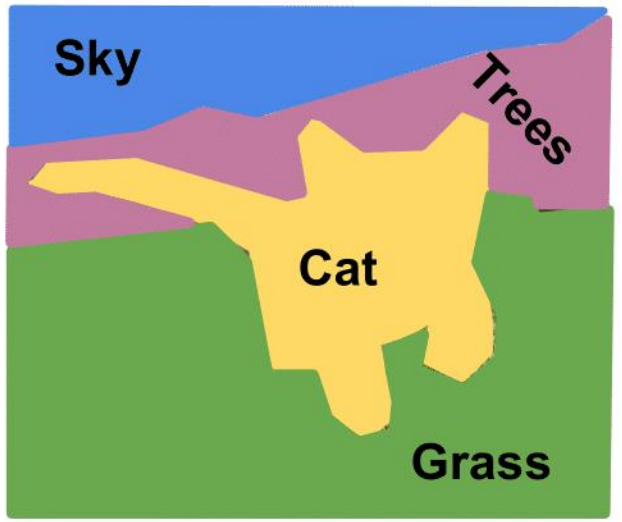
Multiple Object

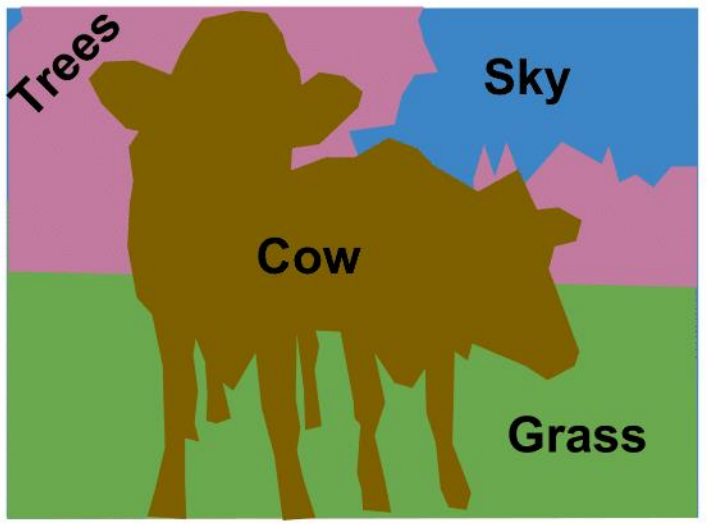
Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Semantic Segmentation

Label each pixel in the image with a category label

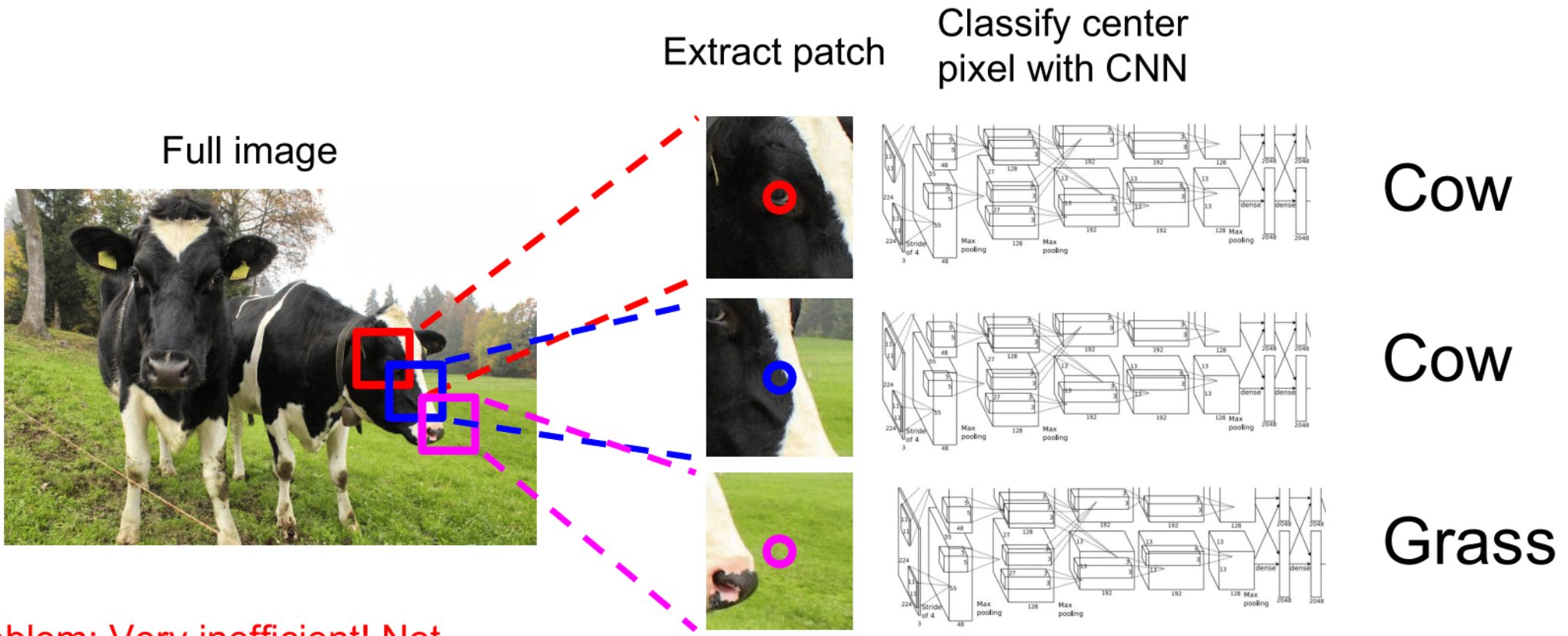
Don’t differentiate instances, only care about pixels





Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Semantic Segmentation Idea: Sliding Window



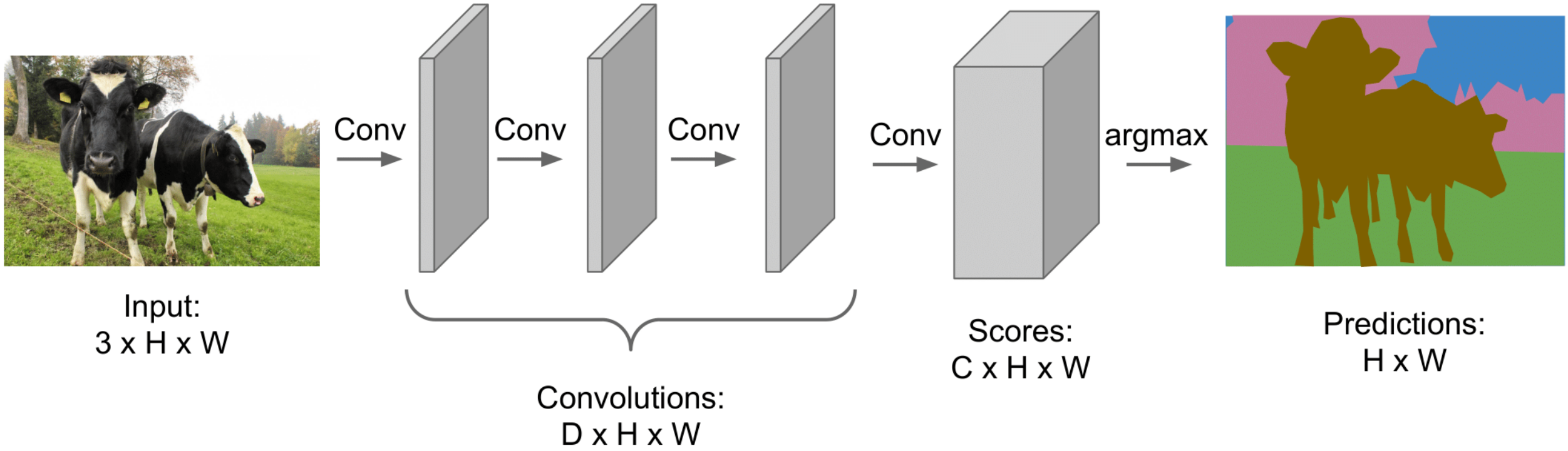
Problem: Very inefficient! Not reusing shared features between overlapping patches

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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Semantic Segmentation Idea: Fully Convolutional

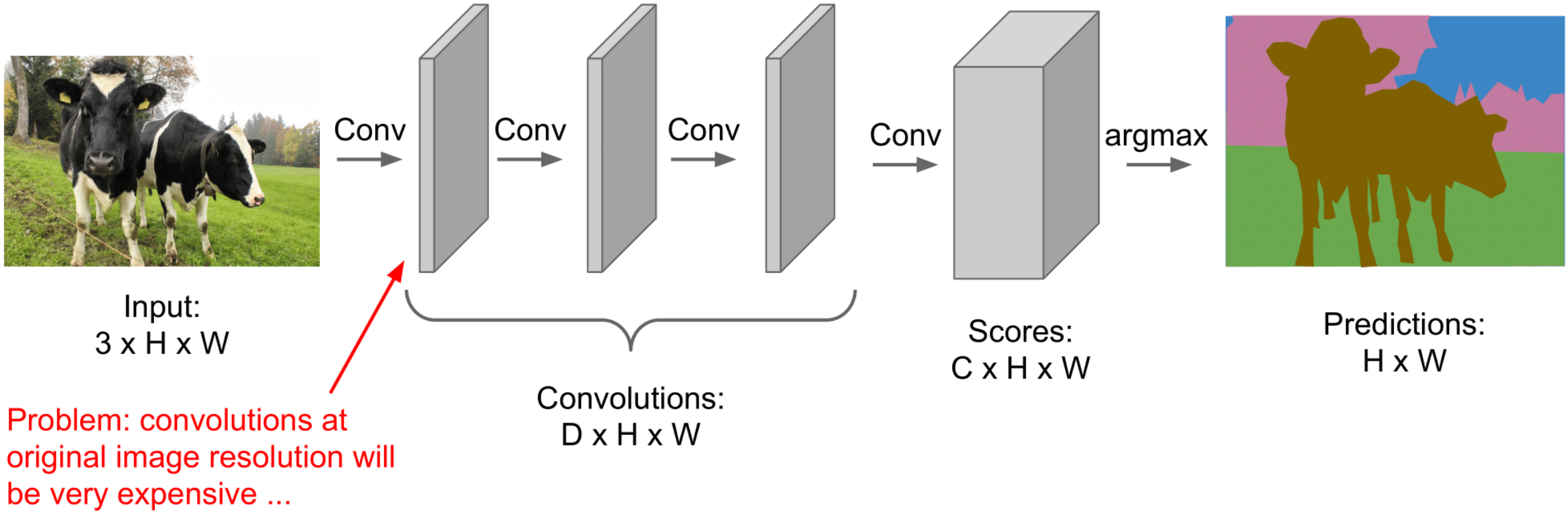
Design a network as a bunch of convolutional layers
to make predictions for pixels all at once!



Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Semantic Segmentation Idea: Fully Convolutional

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

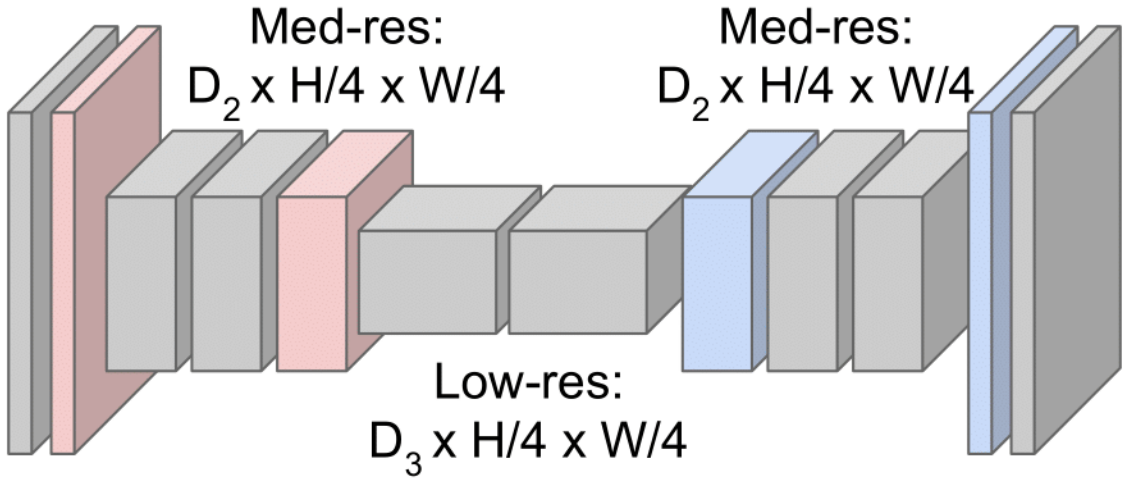


Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

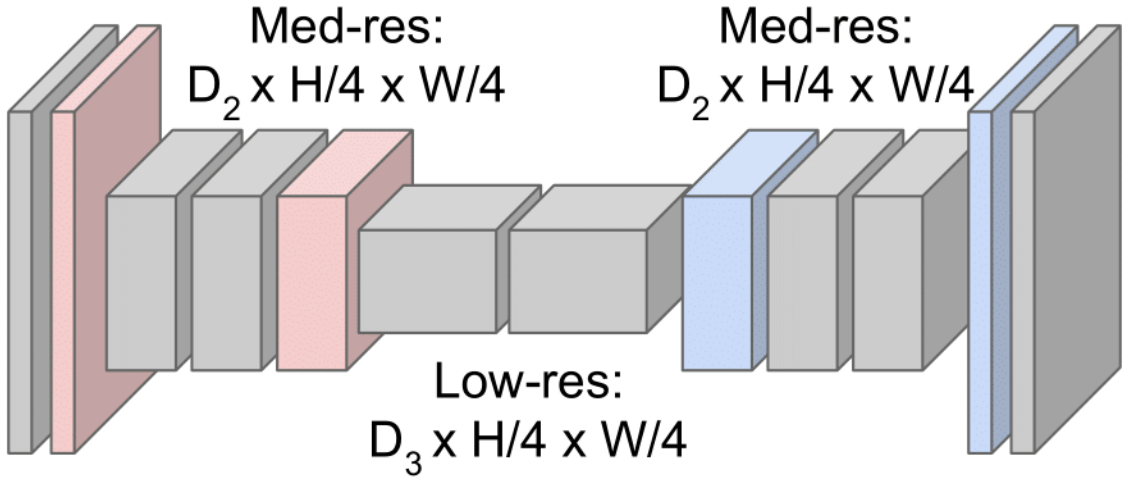
Semantic Segmentation Idea: Fully Convolutional

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

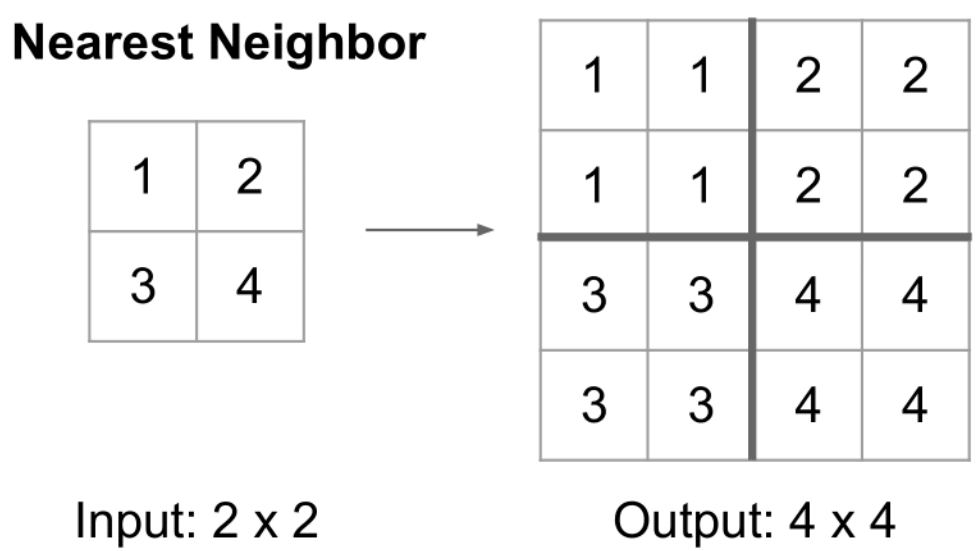
High-res:
D_1 x H/2 x W/2

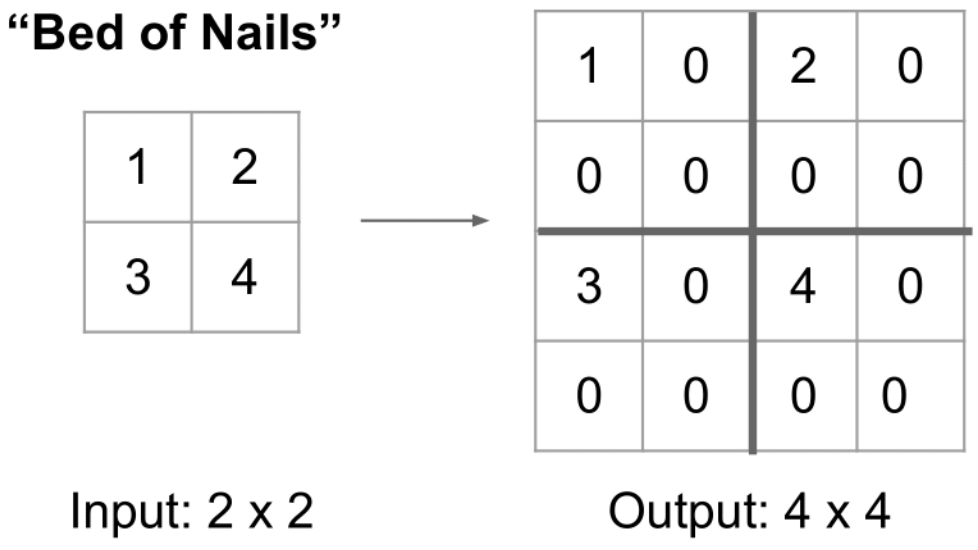
Predictions:
H x W

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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

In-Network upsampling: “Unpooling”





Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

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

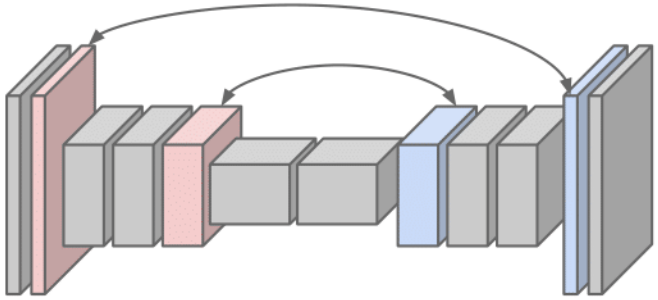
1	2
3	4

Input: 2 x 2

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

Output: 4 x 4

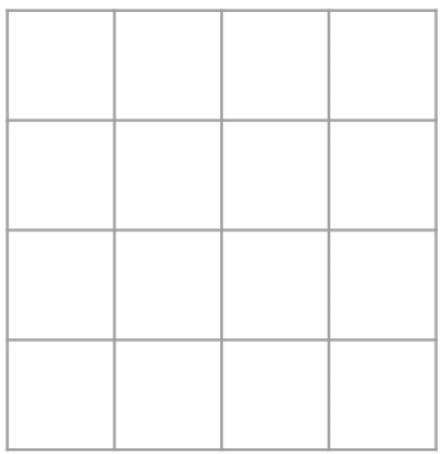
Corresponding pairs of downsampling and upsampling layers



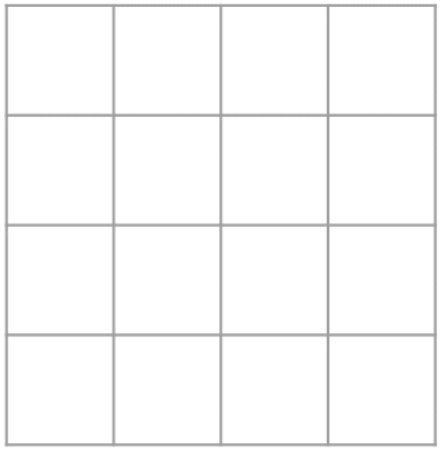
Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Learnable Upsampling: Transpose Convolution

Recall: Typical 3 x 3 convolution, stride 1 pad 1



Input: 4 x 4

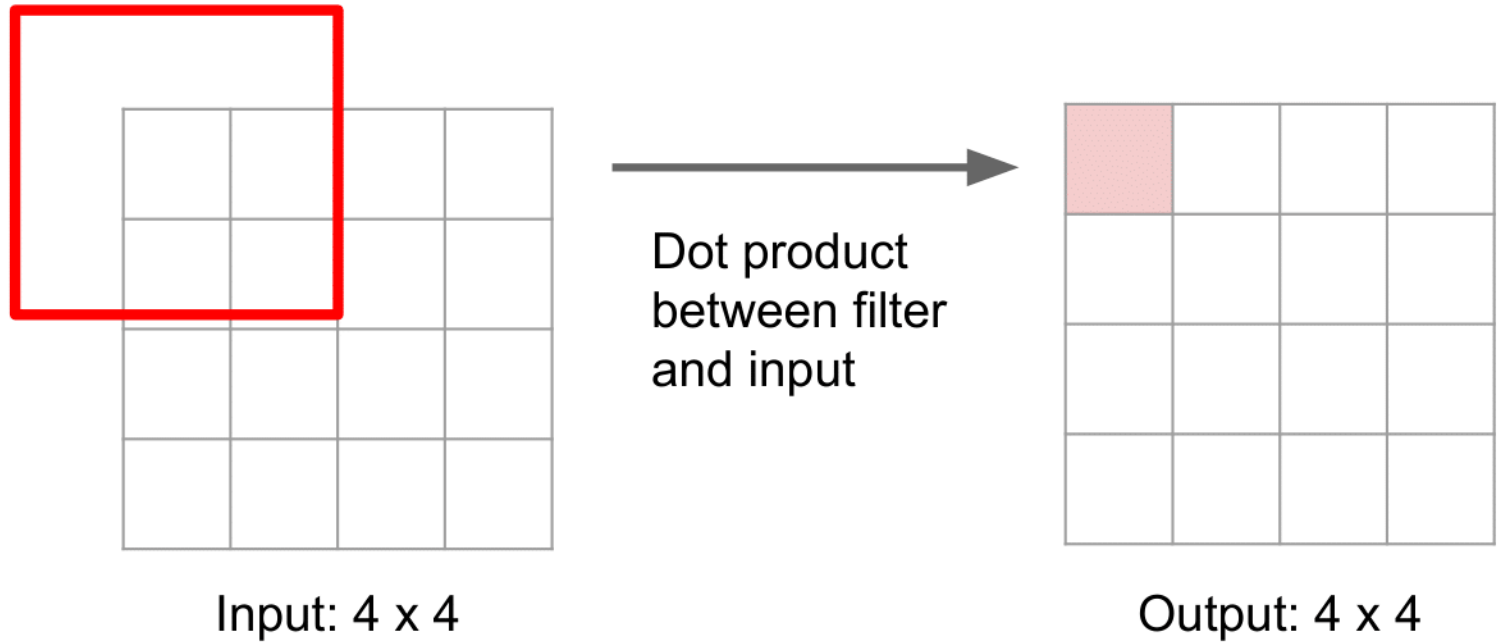


Output: 4 x 4

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Learnable Upsampling: Transpose Convolution

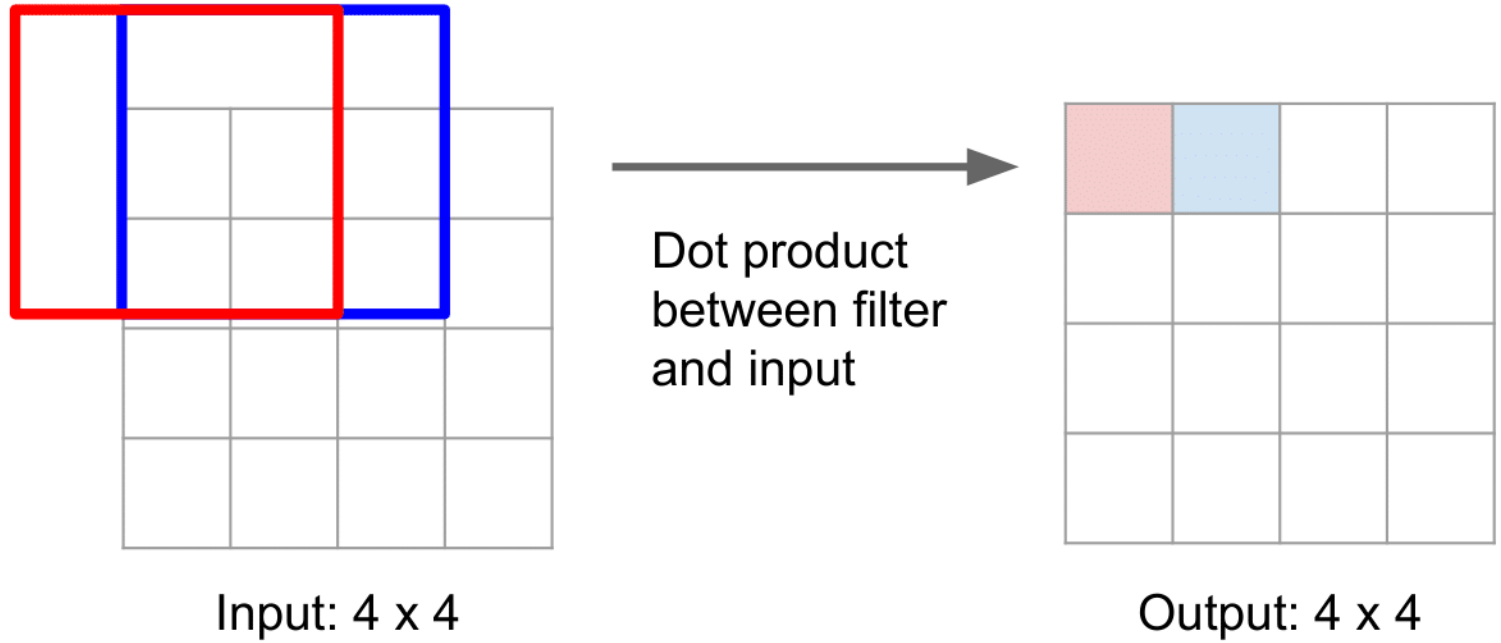
Recall: Normal 3 x 3 convolution, stride 1 pad 1



Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Learnable Upsampling: Transpose Convolution

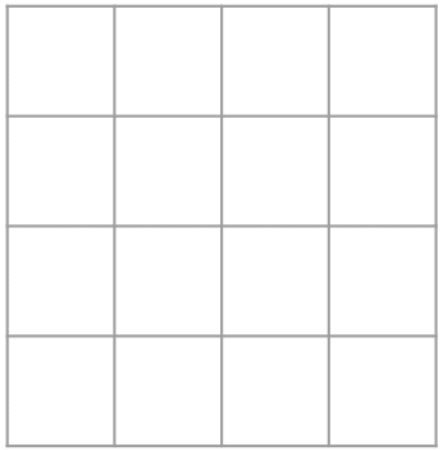
Recall: Normal 3 x 3 convolution, stride 1 pad 1



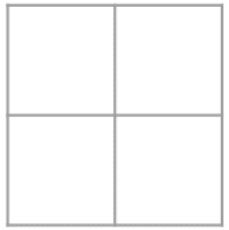
Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Learnable Upsampling: Transpose Convolution

Recall: Normal 3 x 3 convolution, stride 2 pad 1



Input: 4 x 4

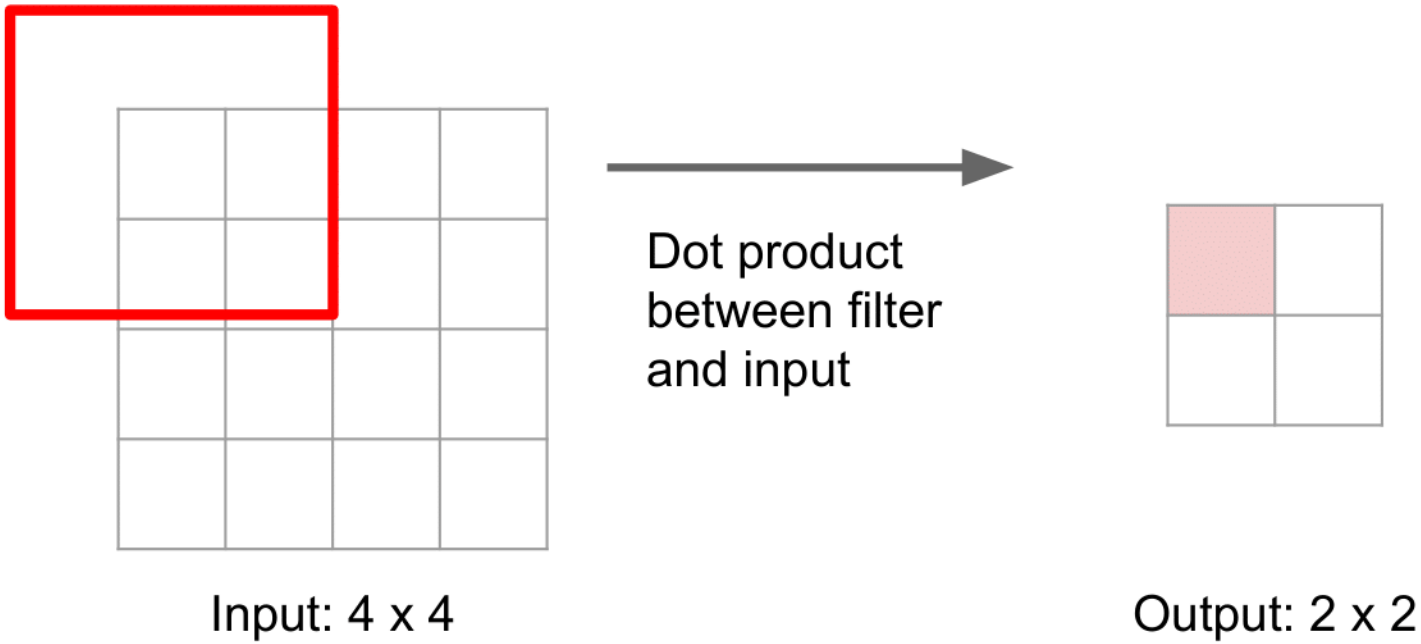


Output: 2 x 2

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Learnable Upsampling: Transpose Convolution

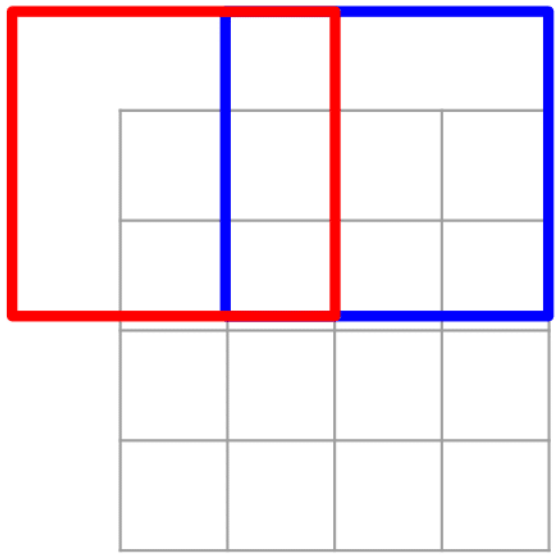
Recall: Normal 3 x 3 convolution, stride 2 pad 1



Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

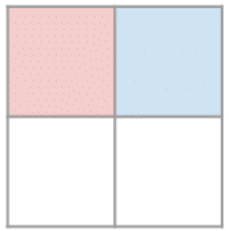
Learnable Upsampling: Transpose Convolution

Recall: Normal 3 x 3 convolution, stride 2 pad 1



Input: 4 x 4

Dot product between filter and input



Output: 2 x 2

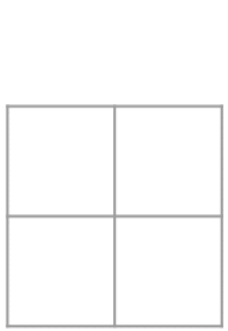
Filter moves 2 pixels in the input for every one pixel in the output

Stride gives ratio between movement in input and output

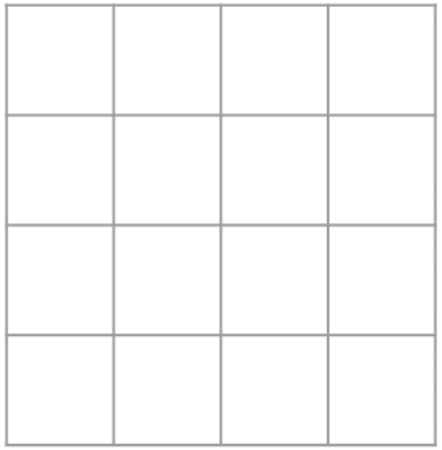
Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Learnable Upsampling: Transpose Convolution

3 x 3 **transpose** convolution, stride 2 pad 1



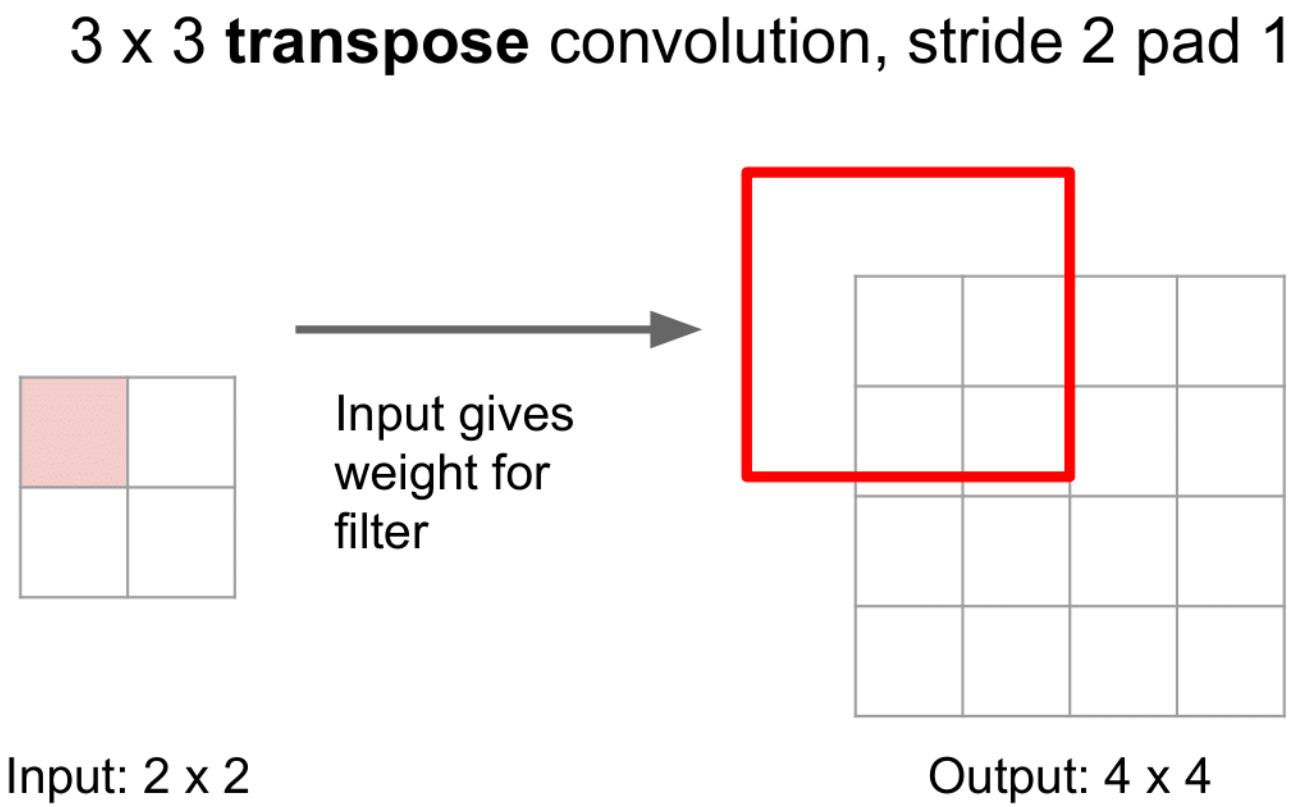
Input: 2 x 2



Output: 4 x 4

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Learnable Upsampling: Transpose Convolution

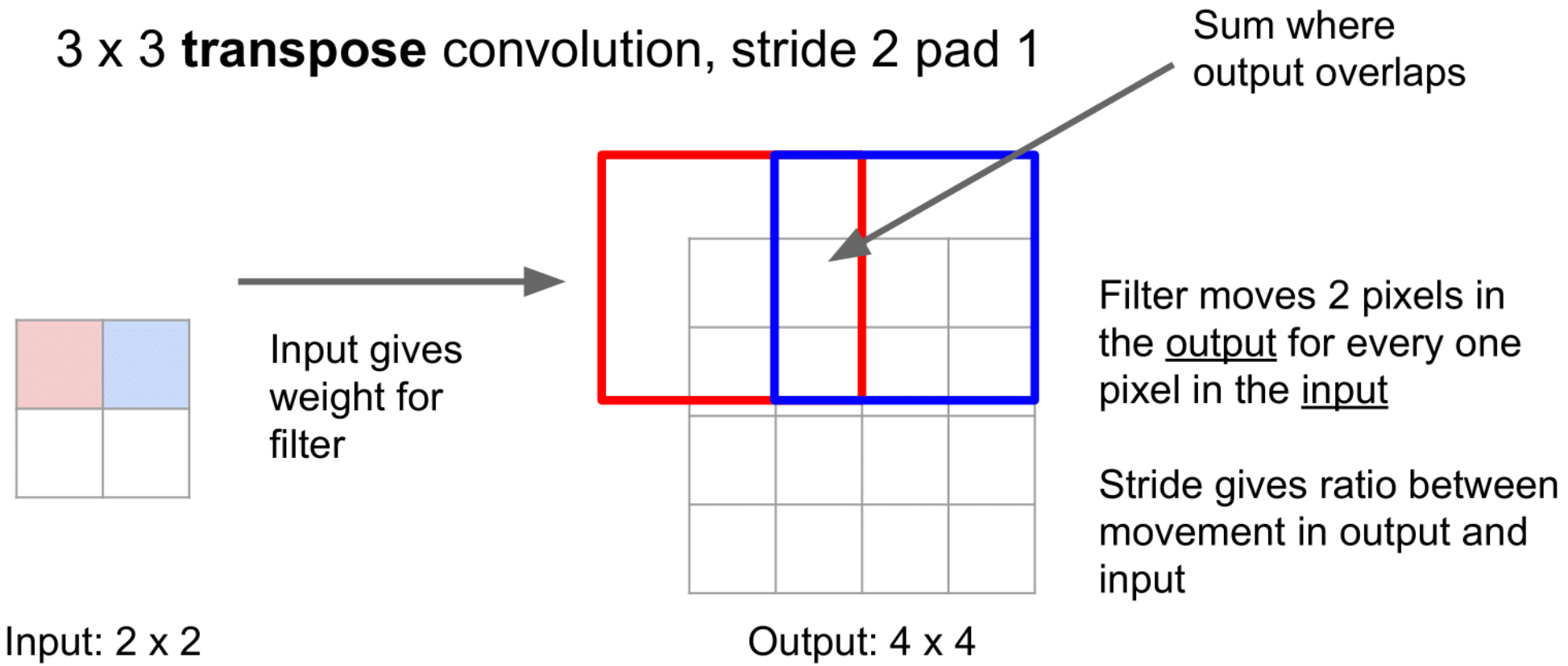


Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Learnable Upsampling: Transpose Convolution

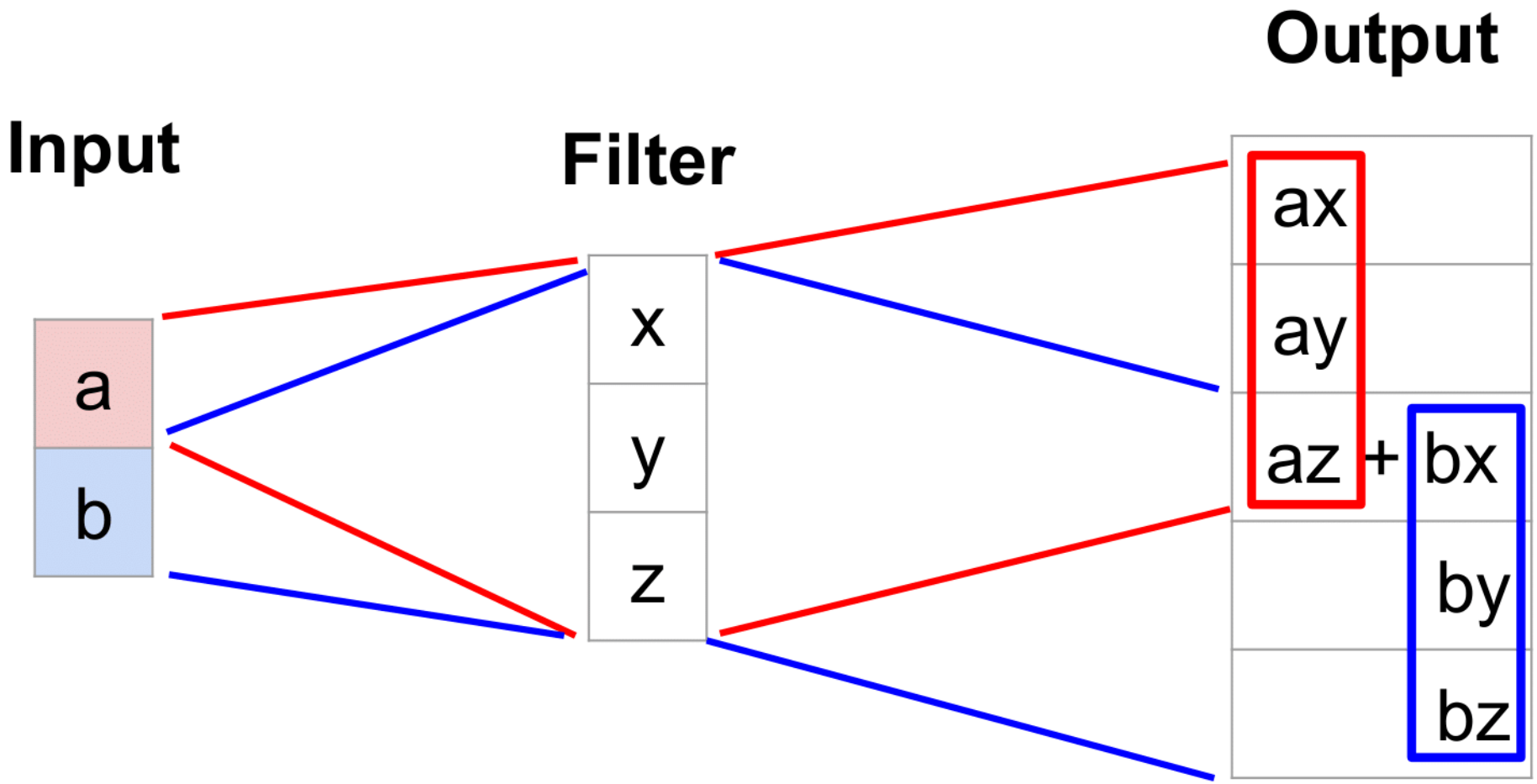
Other names:
-Deconvolution (bad)
-Upconvolution
-Fractionally strided convolution
-Backward strided convolution

3 x 3 **transpose** convolution, stride 2 pad 1



Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Transpose Convolution: 1D Example



Output contains copies of the filter weighted by the input, summing at where at overlaps in the output

Need to crop one pixel from output to make output exactly 2x input

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Convolution as Matrix Multiplication (1D Example)

We can express convolution in terms of a matrix multiplication

$$\vec{x} * \vec{a} = X\vec{a}$$

$$\begin{bmatrix} x & y & x & 0 & 0 & 0 \\ 0 & x & y & x & 0 & 0 \\ 0 & 0 & x & y & x & 0 \\ 0 & 0 & 0 & x & y & x \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ ax + by + cz \\ bx + cy + dz \\ cx + dy \end{bmatrix}$$

Example: 1D conv, kernel size=3, stride=1, padding=1

Convolution transpose multiplies by the transpose of the same matrix:

$$\vec{x} *^T \vec{a} = X^T \vec{a}$$

$$\begin{bmatrix} x & 0 & 0 & 0 \\ y & x & 0 & 0 \\ z & y & x & 0 \\ 0 & z & y & x \\ 0 & 0 & z & y \\ 0 & 0 & 0 & z \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} ax \\ ay + bx \\ az + by + cx \\ bz + cy + dx \\ cz + dy \\ dz \end{bmatrix}$$

When stride=1, convolution transpose is just a regular convolution (with different padding rules)

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Convolution as Matrix Multiplication (1D Example)

We can express convolution in terms of a matrix multiplication

$$\vec{x} * \vec{a} = X\vec{a}$$

$$\begin{bmatrix} x & y & z & 0 & 0 & 0 \\ 0 & 0 & x & y & z & 0 \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ bx + cy + dz \end{bmatrix}$$

Example: 1D conv, kernel size=3, stride=2, padding=1

Convolution transpose multiplies by the transpose of the same matrix:

$$\vec{x} *^T \vec{a} = X^T \vec{a}$$

$$\begin{bmatrix} x & 0 \\ y & 0 \\ z & x \\ 0 & y \\ 0 & z \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} ax \\ ay \\ az + bx \\ by \\ bz \\ 0 \end{bmatrix}$$

When stride>1, convolution transpose is no longer a normal convolution!

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

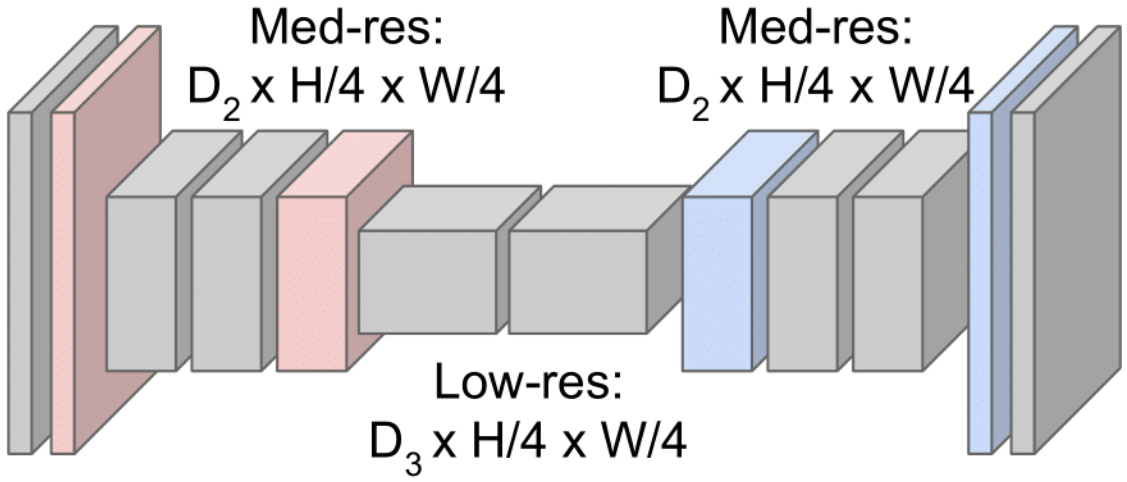
Semantic Segmentation Idea: Fully Convolutional

Downsampling:
Pooling, strided convolution

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

Upsampling:
Unpooling or strided transpose convolution

Input:
3 x H x W



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

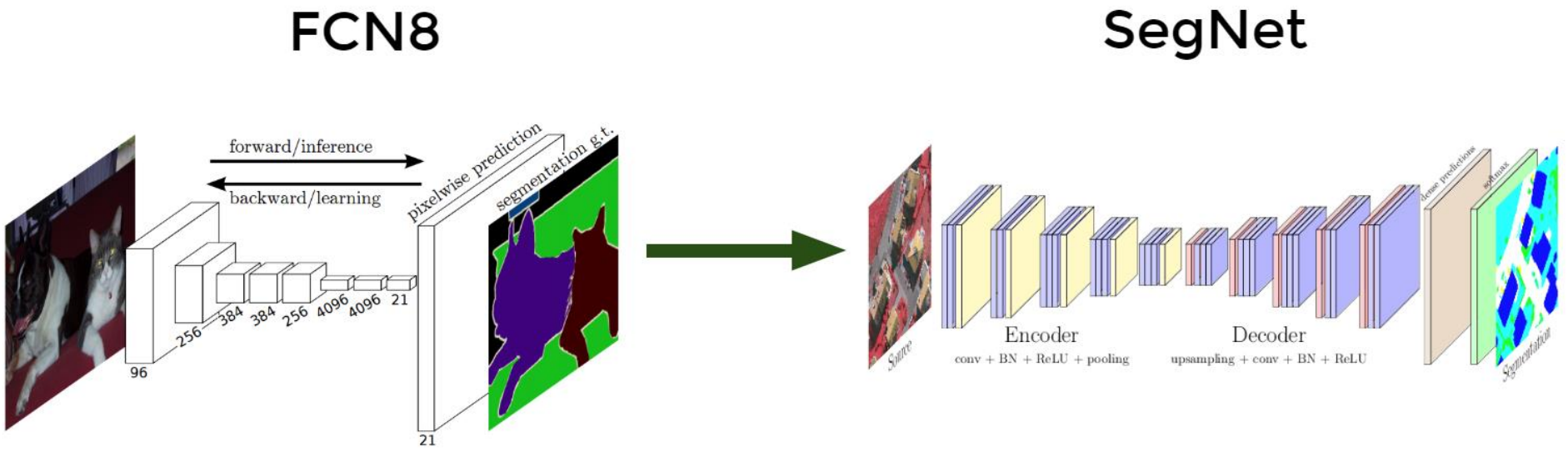
High-res:
D_1 x H/2 x W/2

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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

FCN8 to SegNet



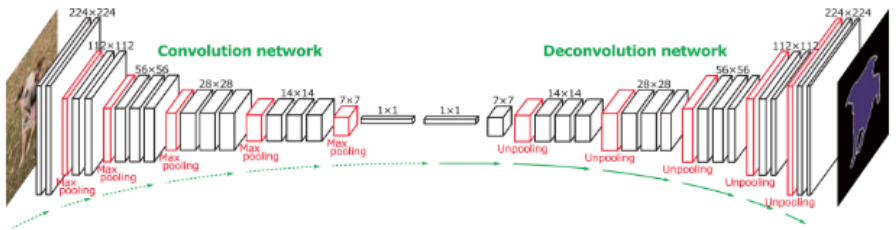
Заменить Upsampling на иерархический Upsampling

V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” arXiv:1511.00561, 2015

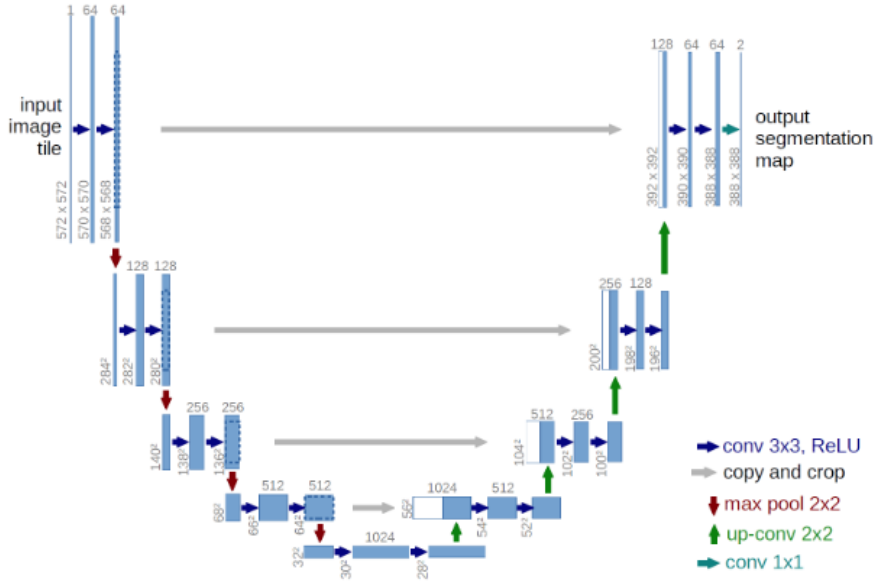
slides by @Vladimir Iglovikov

SegNet to UNet

SegNet



UNet

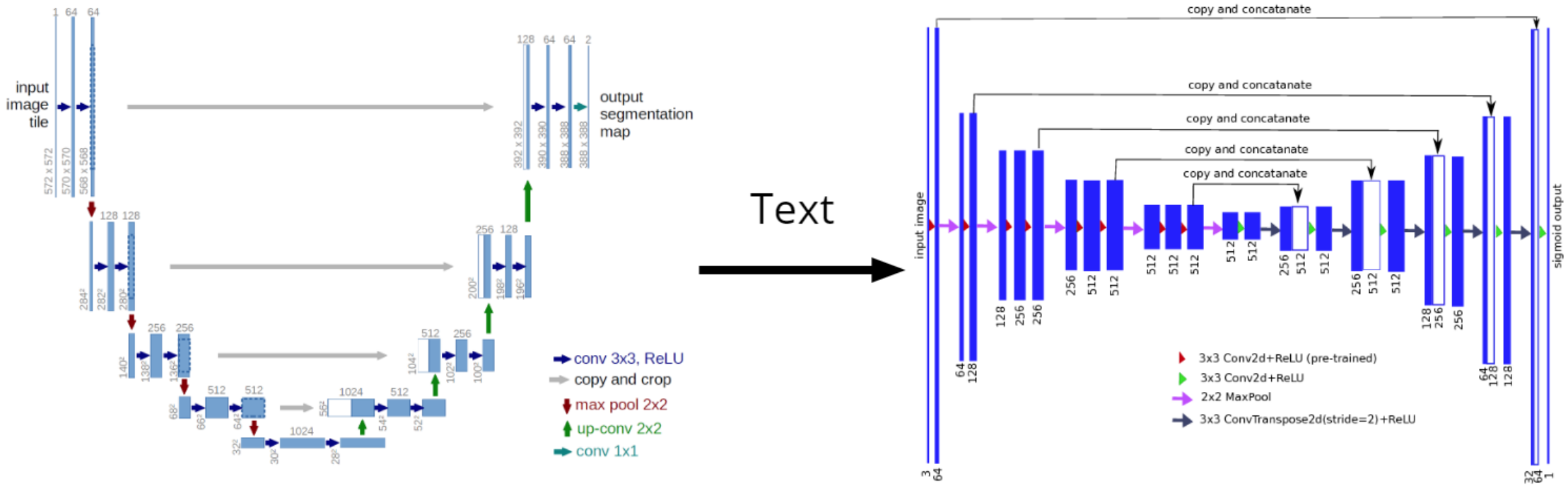


Added skip connections

O. Ronneberger P. Fischer T. Brox "U-net: Convolutional networks for biomedical image segmentation" Proc. Med. Image Comput. Comput.-Assisted Intervention pp. 234-241 2015.

slides by @Vladimir Iglovikov

Unet to TernausNet

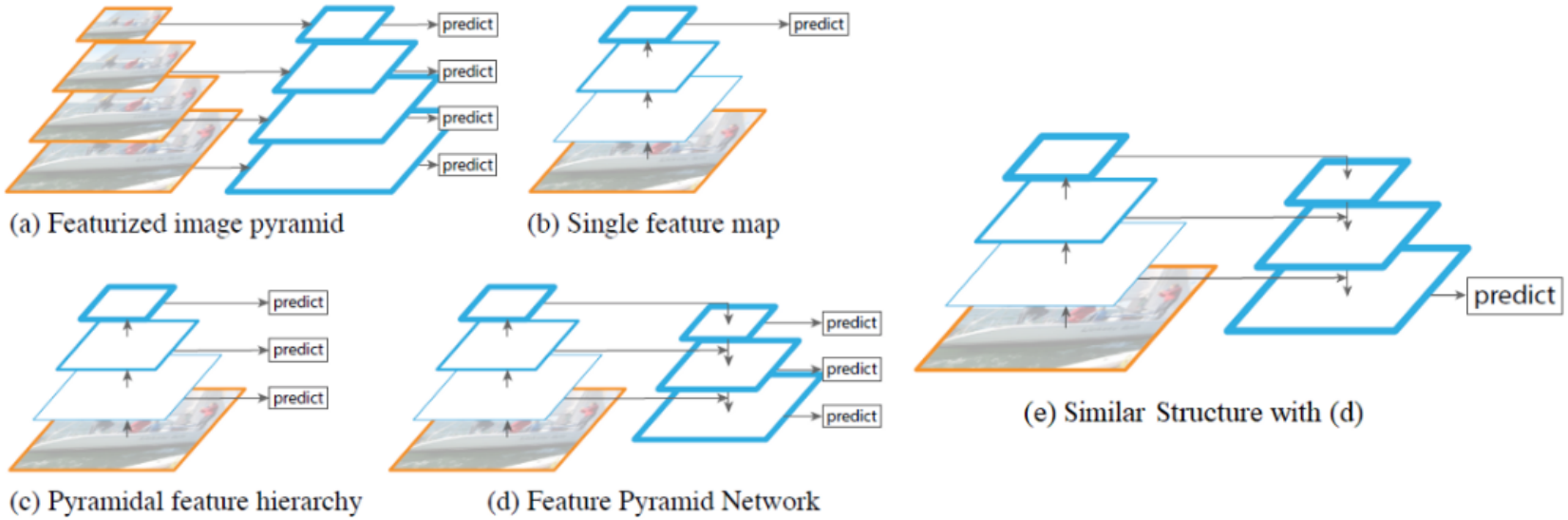


Энкодер инициализируем весами с ImageNet

Iglovikov, V., Shvets, A.: Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. arXiv preprint arXiv:1801.05746 (2018)

slides by @Vladimir Iglovikov

Feature Pyramid Networks (FPN)



(a) Featurized image pyramid (b) Single feature map

(c) Pyramidal feature hierarchy (d) Feature Pyramid Network

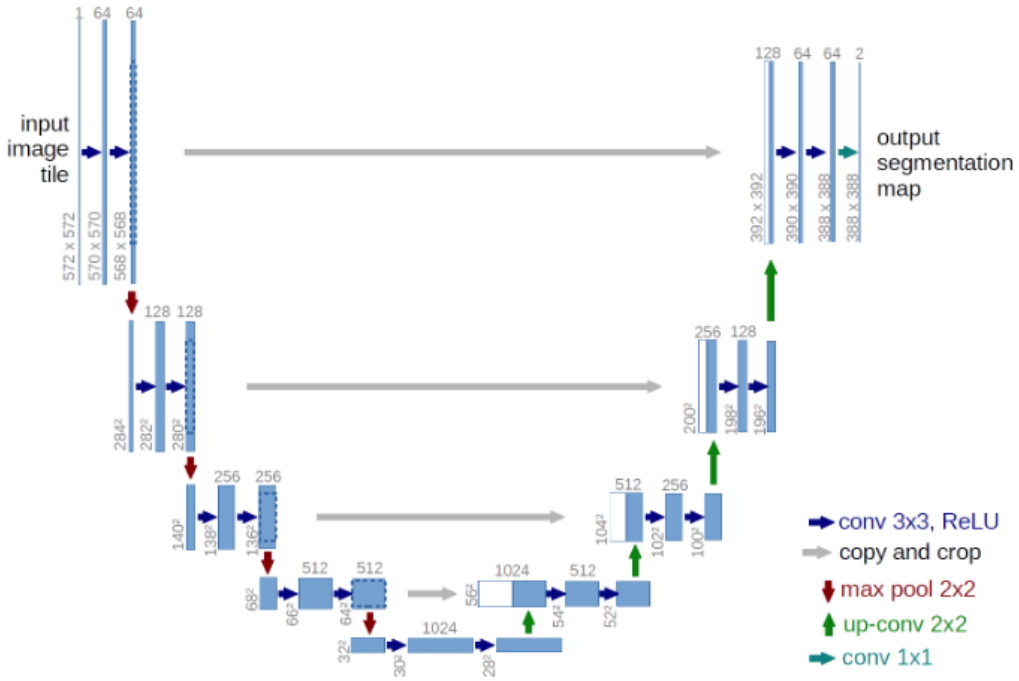
(e) Similar Structure with (d)

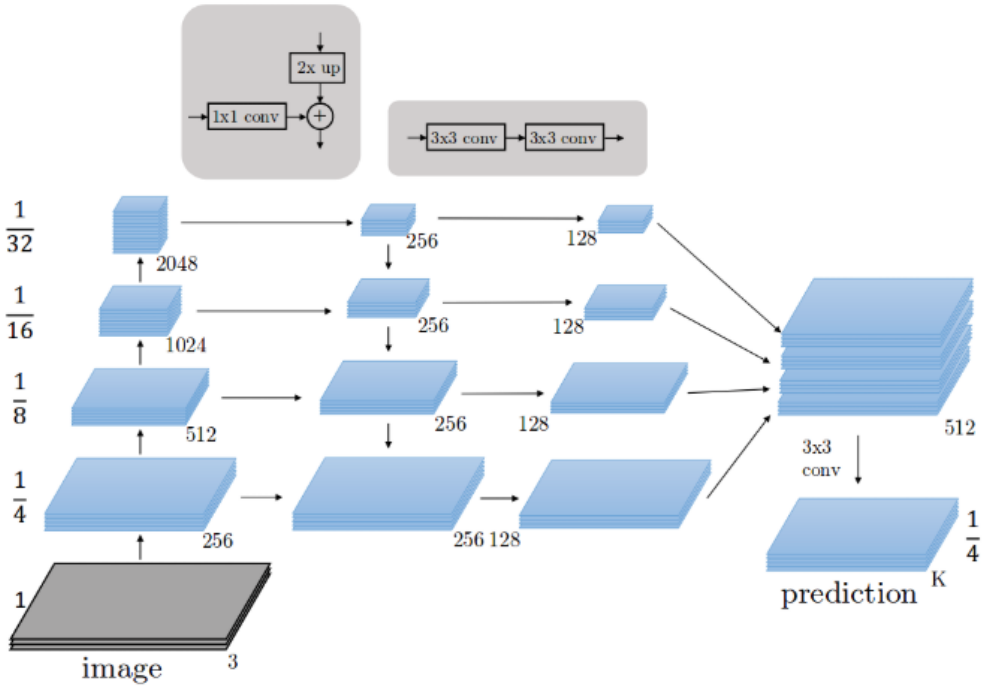
1. Легко добавить во многие архитектуры.
2. Помогает с multiscale

Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2117-2125

slides by @Vladimir Iglovikov

Unet + FPN





slides by @Vladimir Iglovikov

Segmentation Loss Function

Каждый пиксель классификатор => Categorical / Binary Cross Entropy(CCE, BCE)

Но! Метрика Dice / Jaccard

Dice / Jaccard недифференцируемы =>

Soft Dice / Soft Jaccard

и добавляем в loss

Lovasz-Softmax loss

Использовать для FineTune

$$CCE = \sum_c p(x) \log q(x)$$

$$LOSS = BCE - \ln(DICE)$$

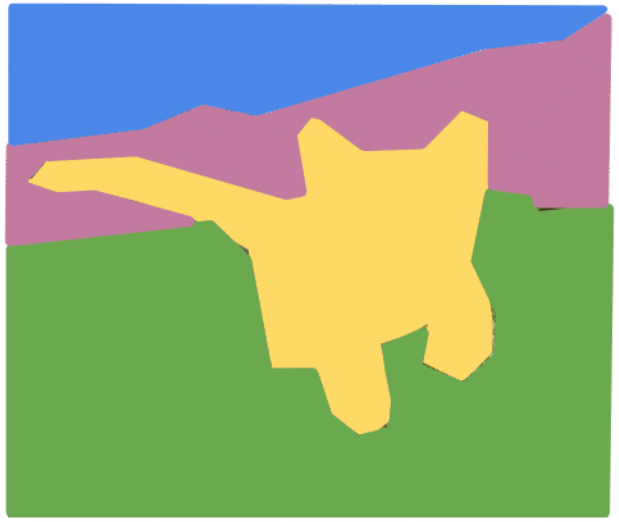
$$BCE = -\sum_i (y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i))$$

$$DICE = 2\frac{\sum_i y_i p_i}{\sum_u y_i + \sum p_i}$$

Berman, M., Rannen Triki, A., Blaschko, M.B.: The lovász-softmax loss: a tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4413–4421 (2018)

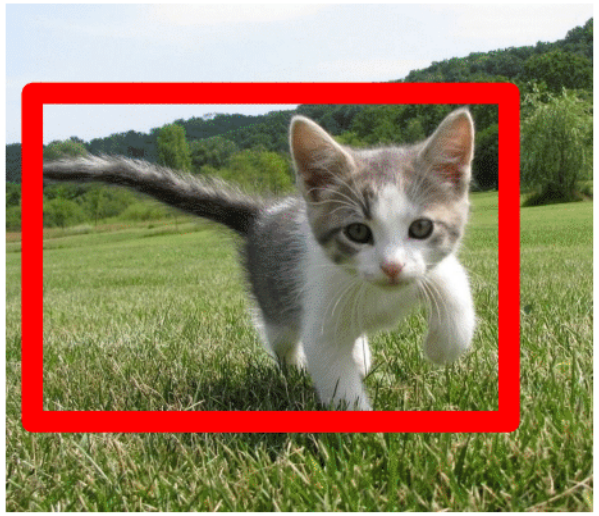
slides by @Vladimir Iglovikov

Classification + Localization



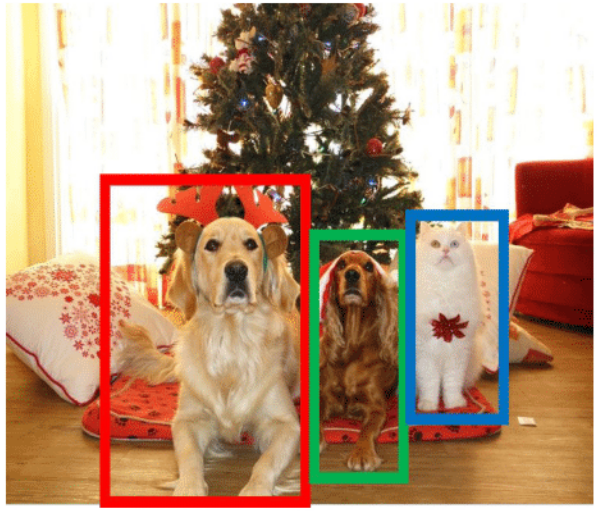
GRASS, CAT, TREE, SKY

No objects, just pixels

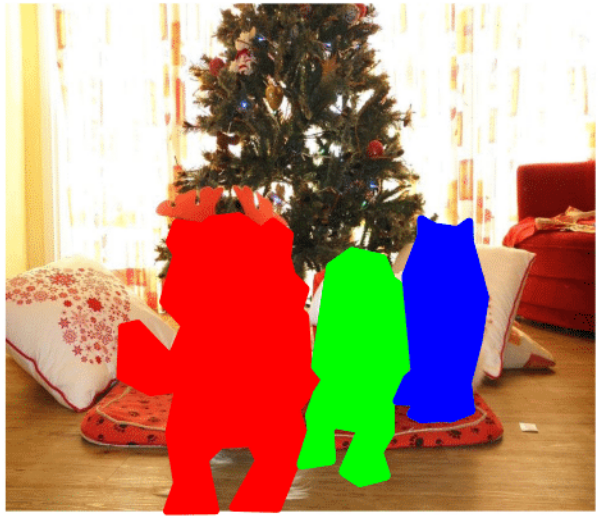


CAT

Single Object



DOG, DOG, CAT



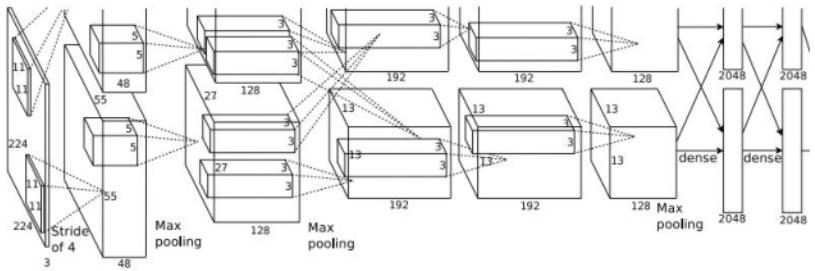
DOG, DOG, CAT

Multiple Object

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

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
...

Softmax Loss

Correct label: Cat

Fully Connected: 4096 to 4

Box Coordinates (x, y, w, h)

L2 Loss

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

Multitask Loss

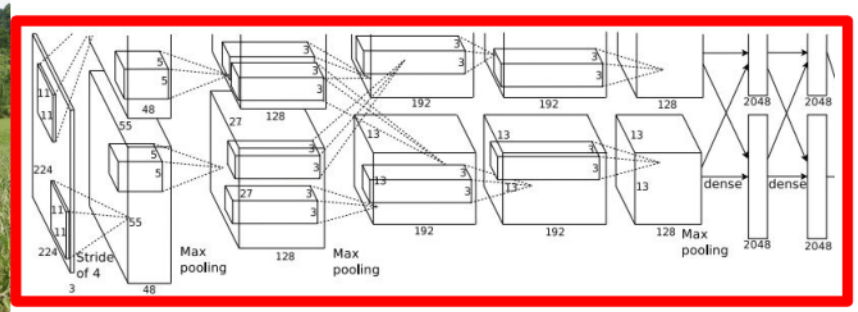
Loss

Treat localization as a regression problem!

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

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
...

Softmax Loss

Correct label:
Cat

Fully Connected: 4096 to 4

Box Coordinates
(x, y, w, h)

L2 Loss

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

+ → **Loss**

Treat localization as a regression problem!

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Aside: Human Pose Estimation

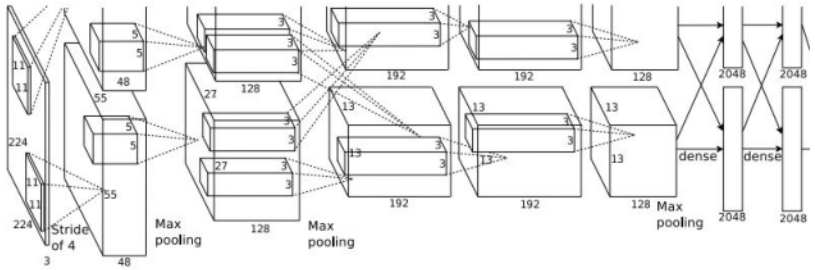
Represent pose as a set of 14 joint positions:

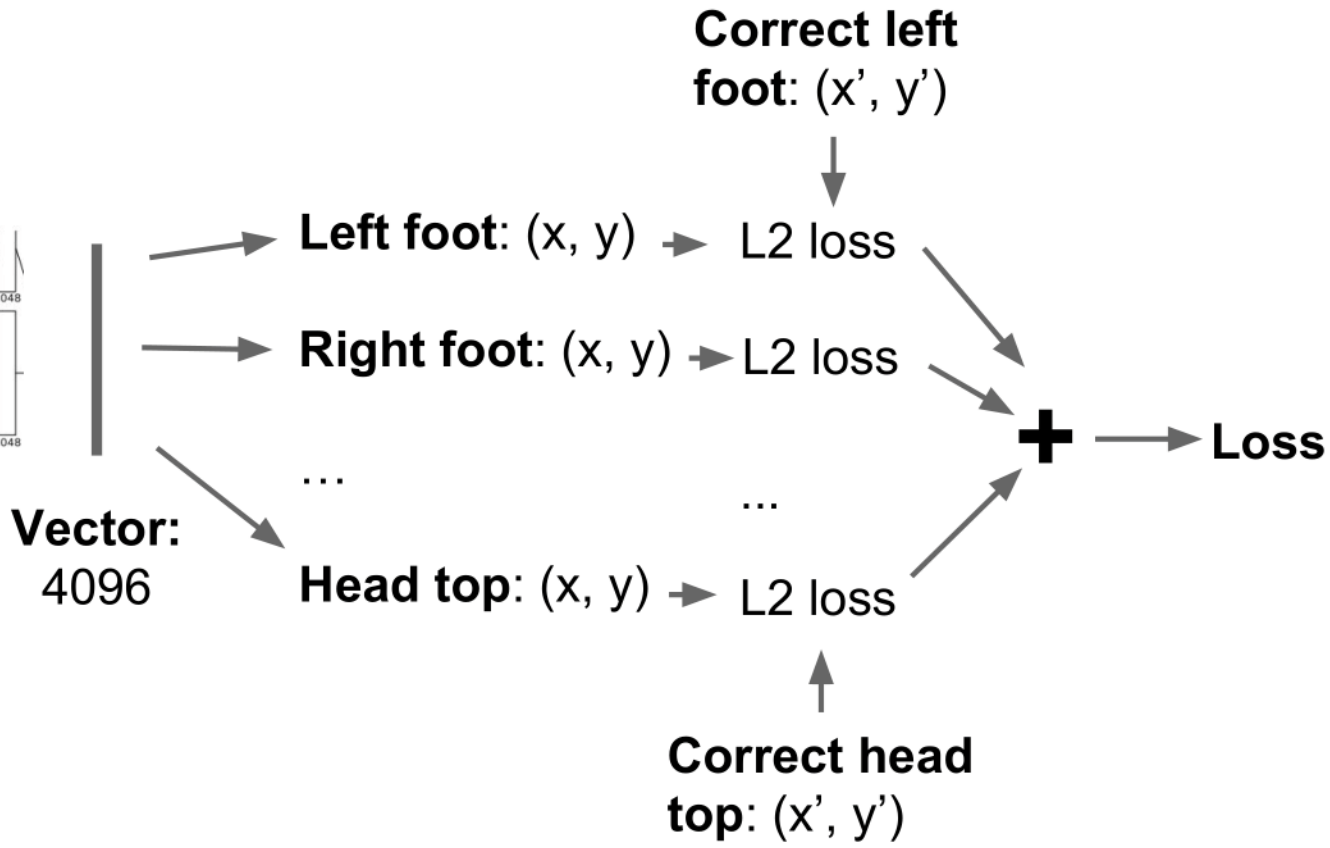
Left / right foot
Left / right knee
Left / right hip
Left / right shoulder
Left / right elbow
Left / right hand
Neck
Head top

Johnson and Everingham, "Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation", BMVC 2010

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Aside: Human Pose Estimation

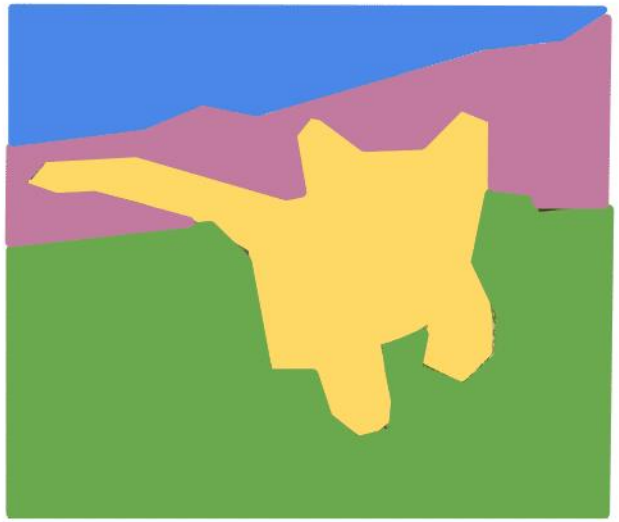




Toshev and Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks", CVPR 2014

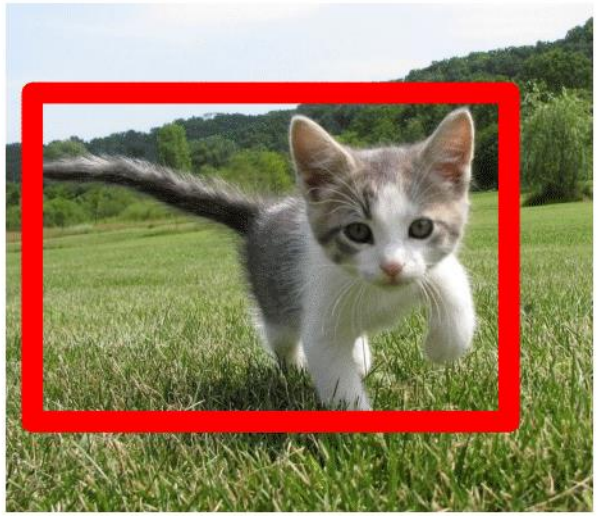
Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Object Detection



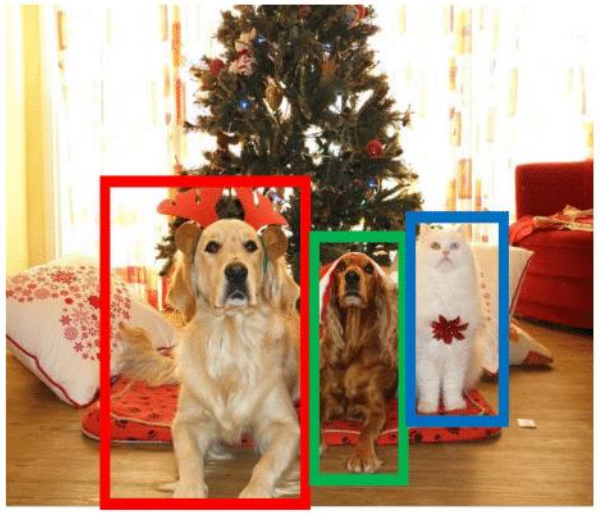
GRASS, CAT, TREE, SKY

No objects, just pixels

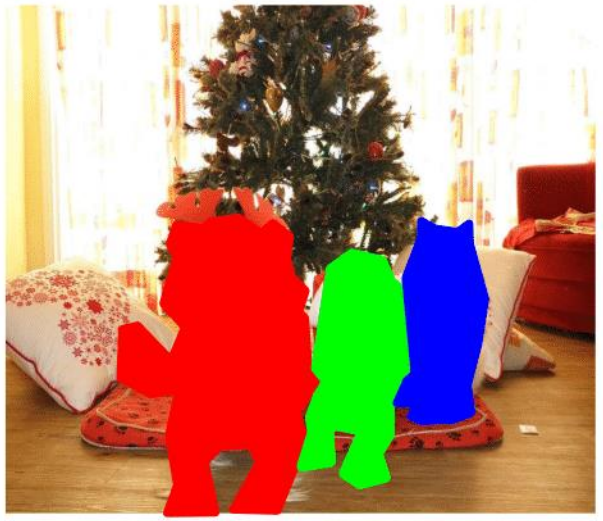


CAT

Single Object



DOG, DOG, CAT

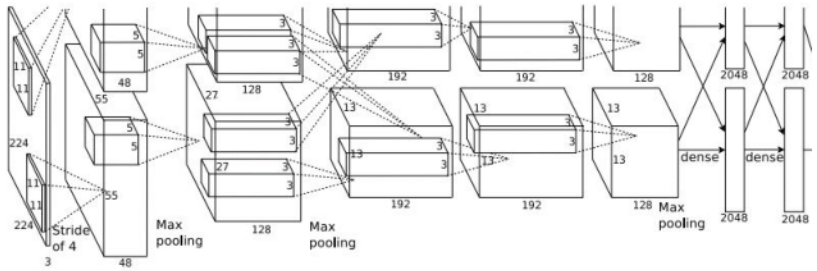


DOG, DOG, CAT

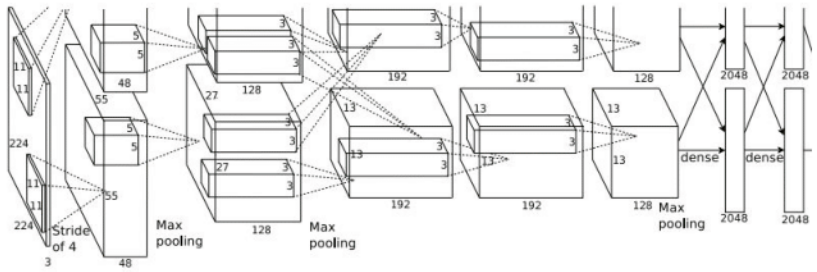
Multiple Object

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

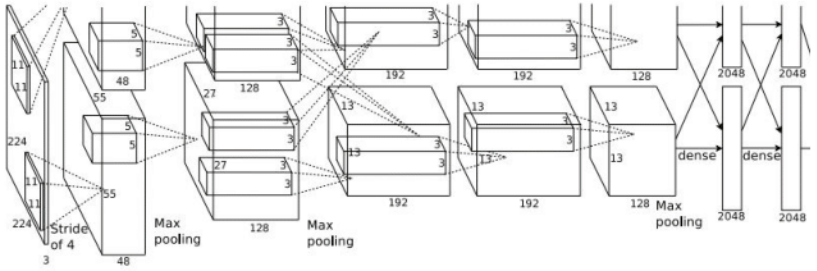
Object Detection as Regression?



CAT: (x, y, w, h)



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

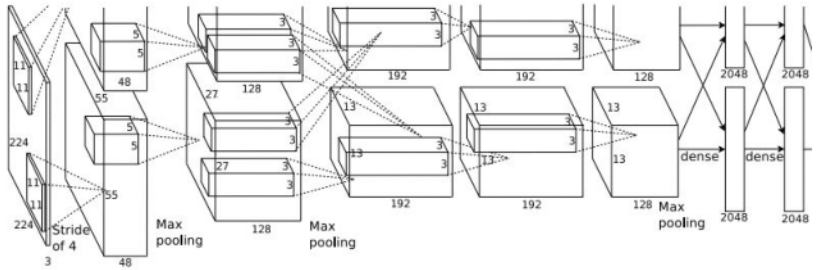


DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
….

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

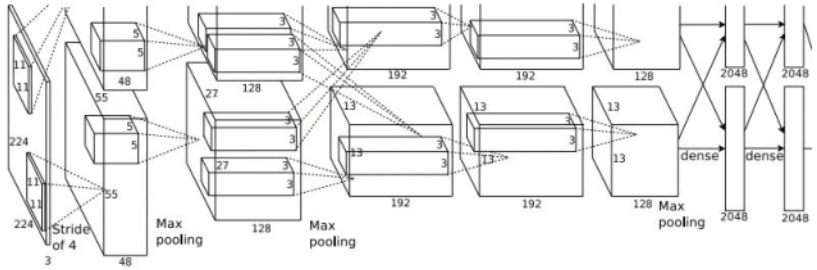
Object Detection as Regression?

Each image needs a different number of outputs!



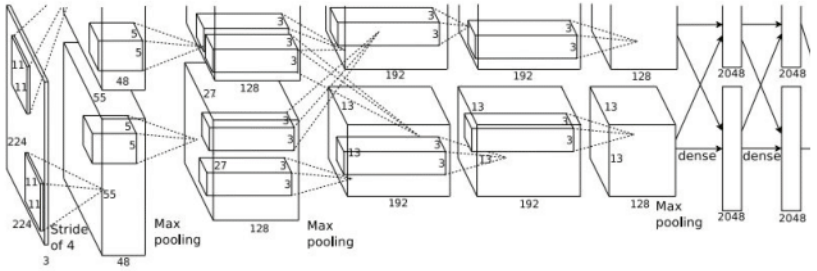
CAT: (x, y, w, h)

4 numbers



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

16 numbers



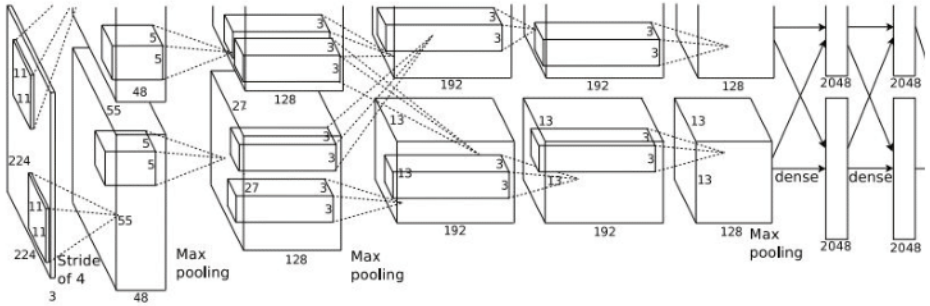
DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

Many numbers!

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Object Detection as Classification: Sliding Window

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

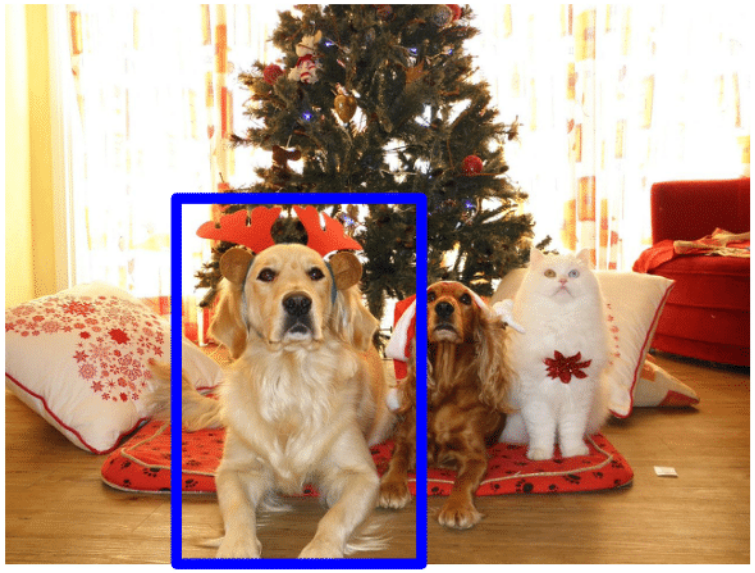


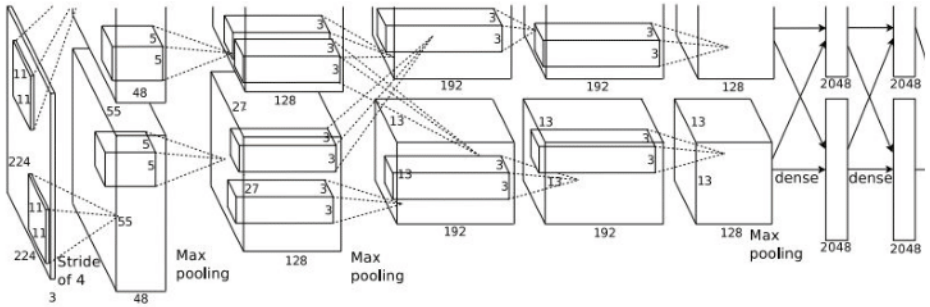
Dog? NO
Cat? NO
Background? YES

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Object Detection as Classification: Sliding Window

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



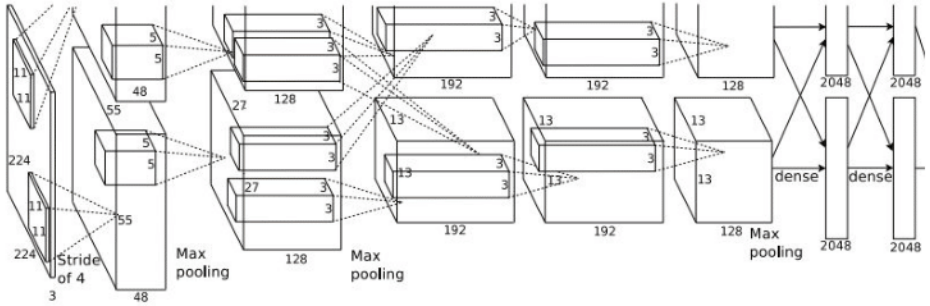


Dog? YES
Cat? NO
Background? NO

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Object Detection as Classification: Sliding Window

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

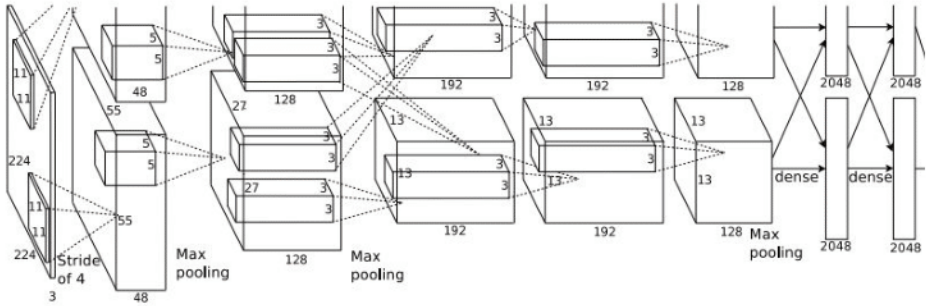


Dog? YES
Cat? NO
Background? NO

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Object Detection as Classification: Sliding Window

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

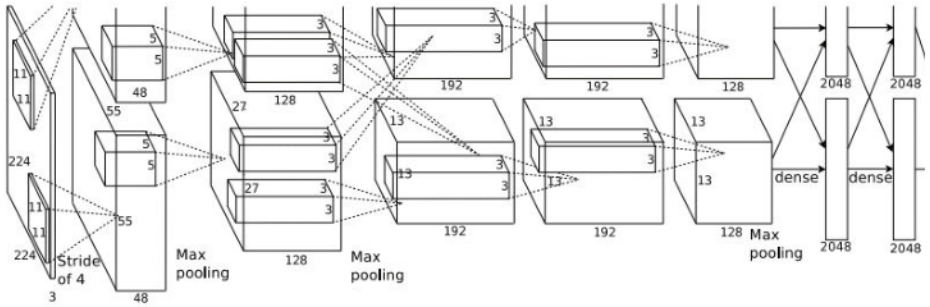


Dog? NO
Cat? YES
Background? NO

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Object Detection as Classification: Sliding Window

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



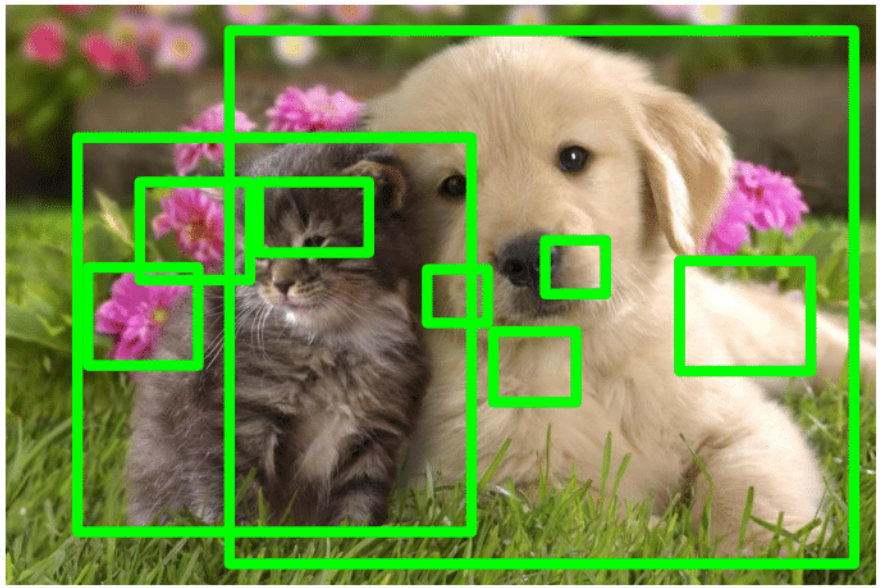
Dog? NO
Cat? YES
Background? NO

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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

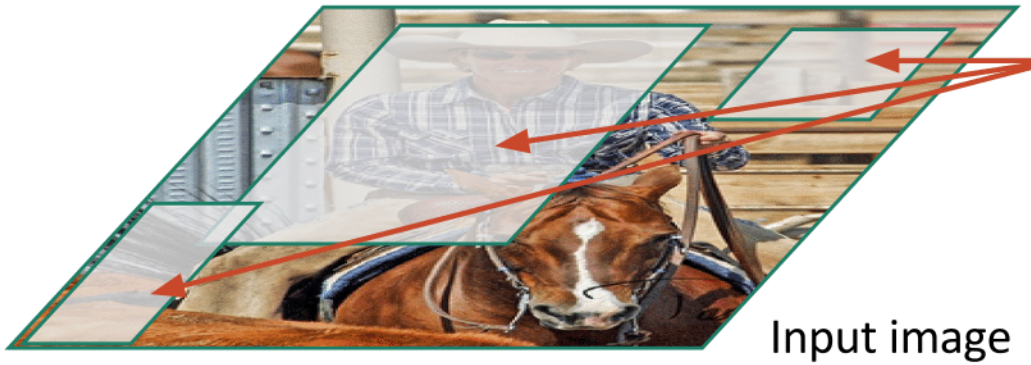
R-CNN

Input image

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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

R-CNN



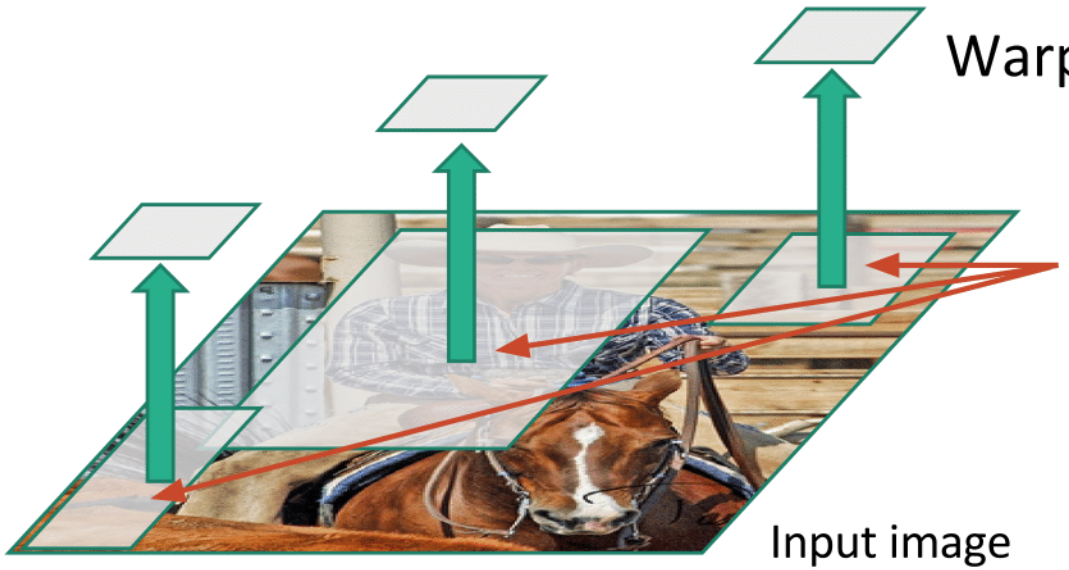
Input image

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.

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

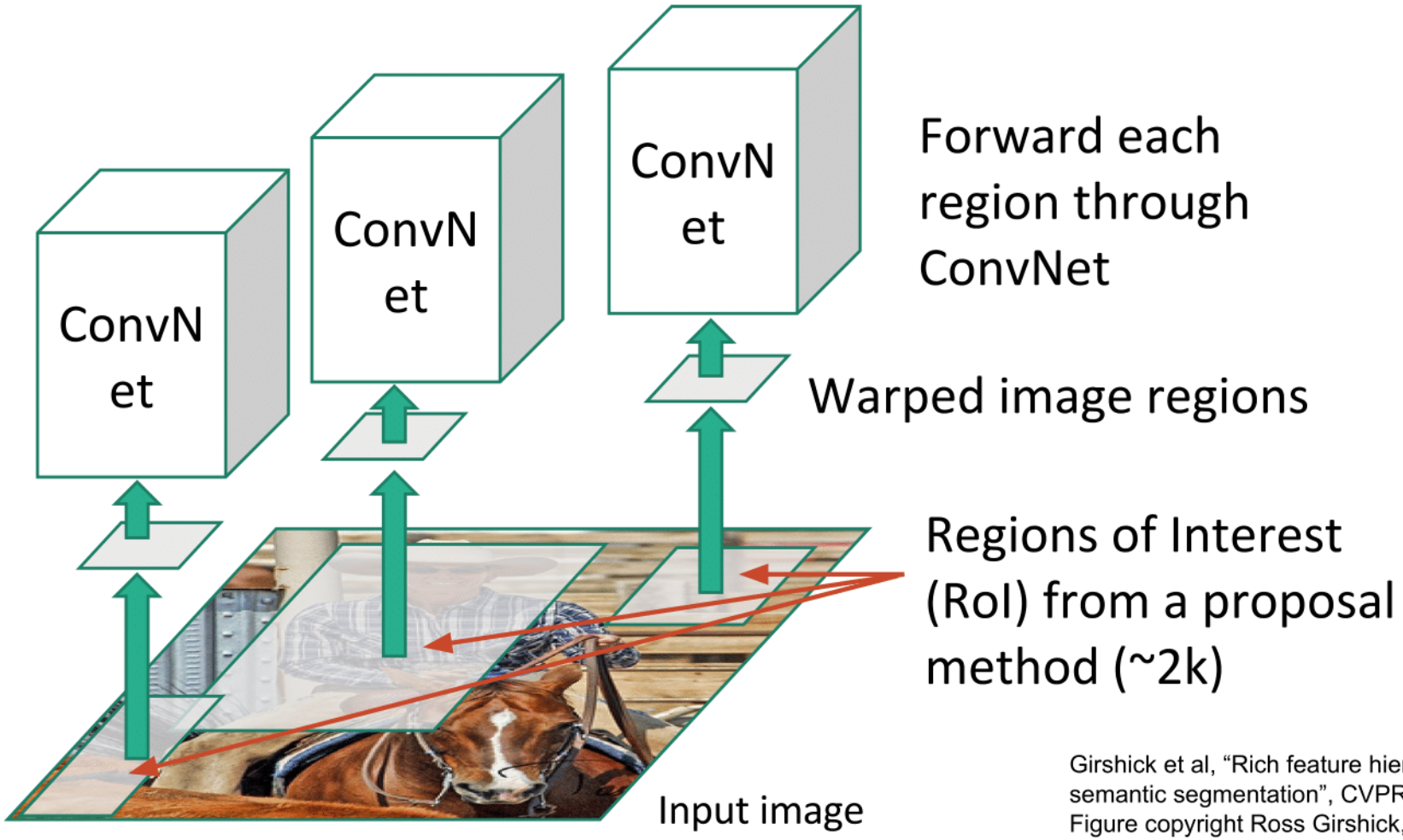
R-CNN



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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

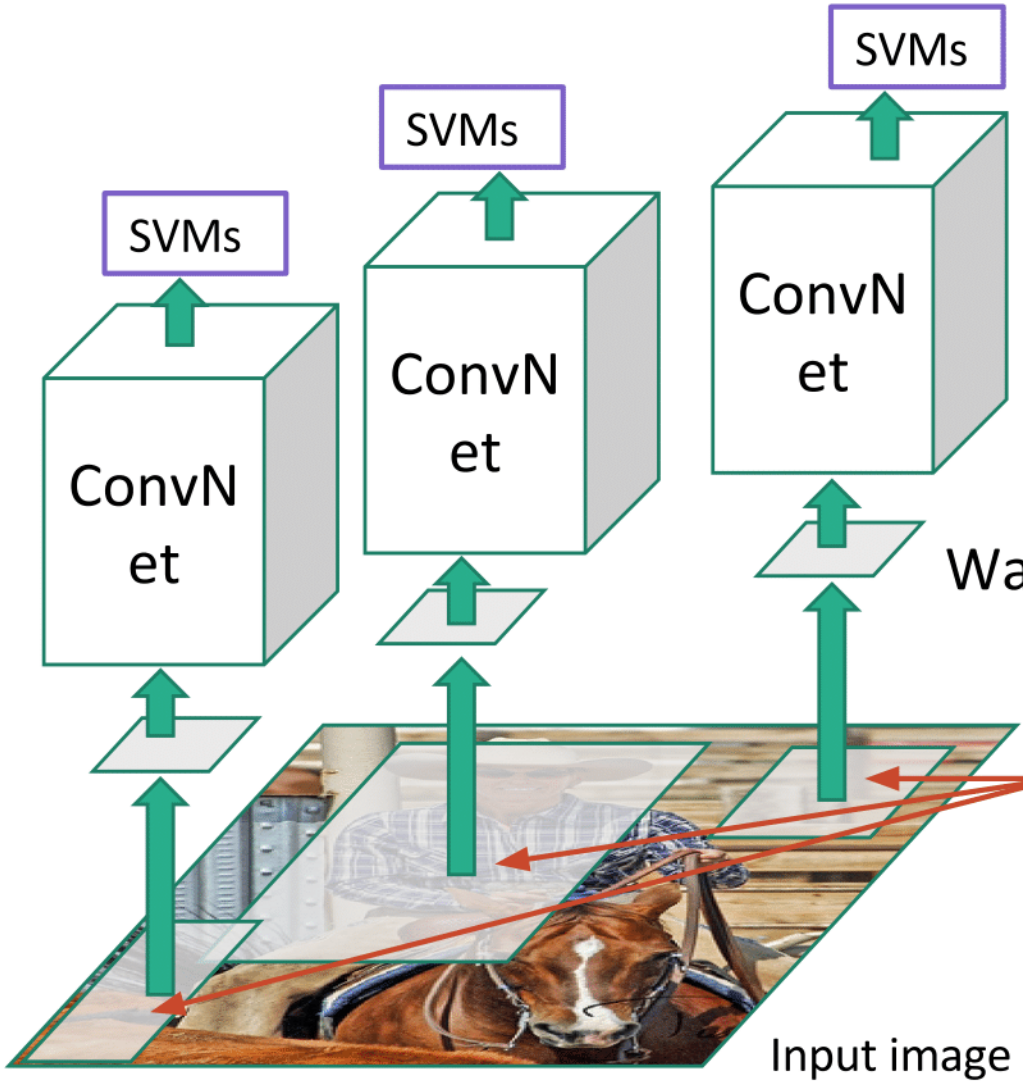
R-CNN



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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

R-CNN



Classify regions with SVMs

Forward each region through ConvNet

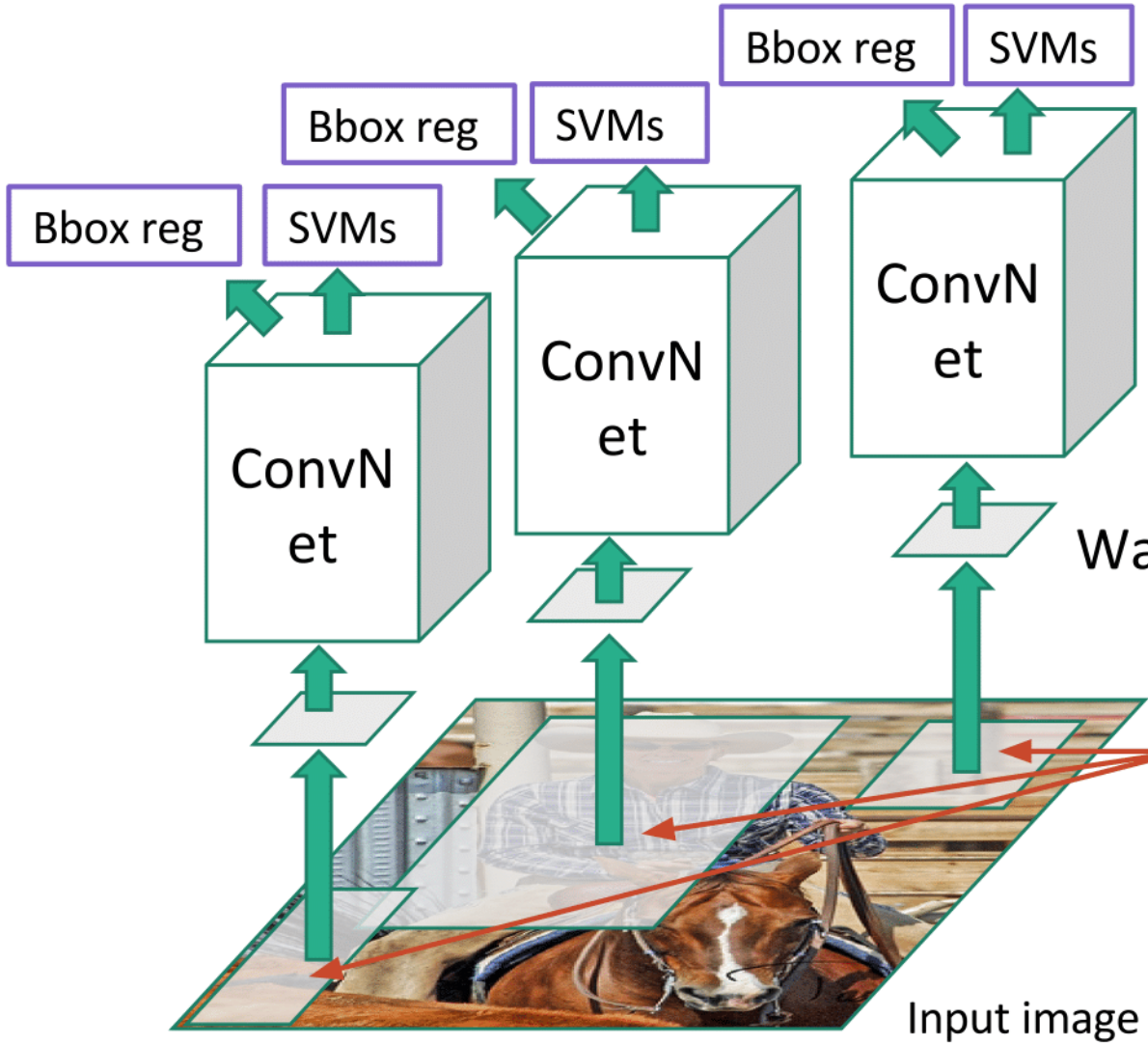
Warped image regions

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

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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

R-CNN



Linear Regression for bounding box offsets

Classify regions with SVMs

Forward each region through ConvNet

Warped image regions

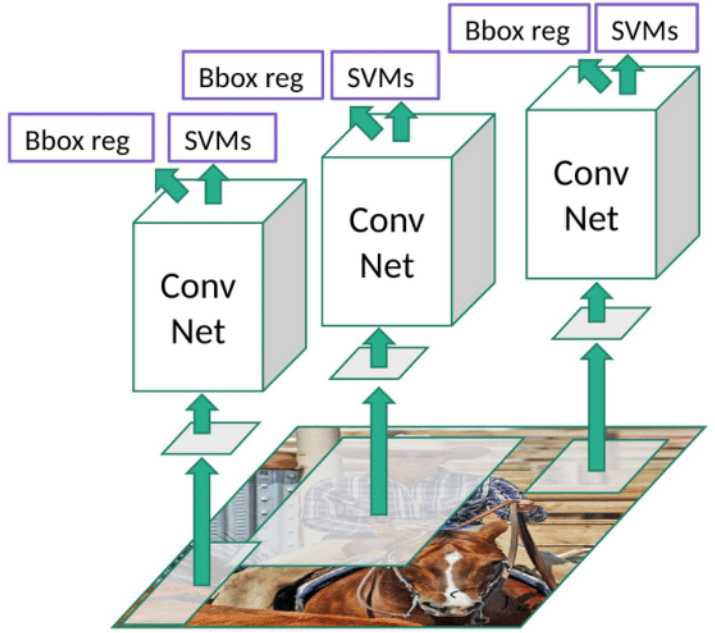
Regions of Interest (RoI) from a proposal method (~2k)

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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

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.
Slide copyright Ross Girshick, 2015; source. Reproduced with permission.

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

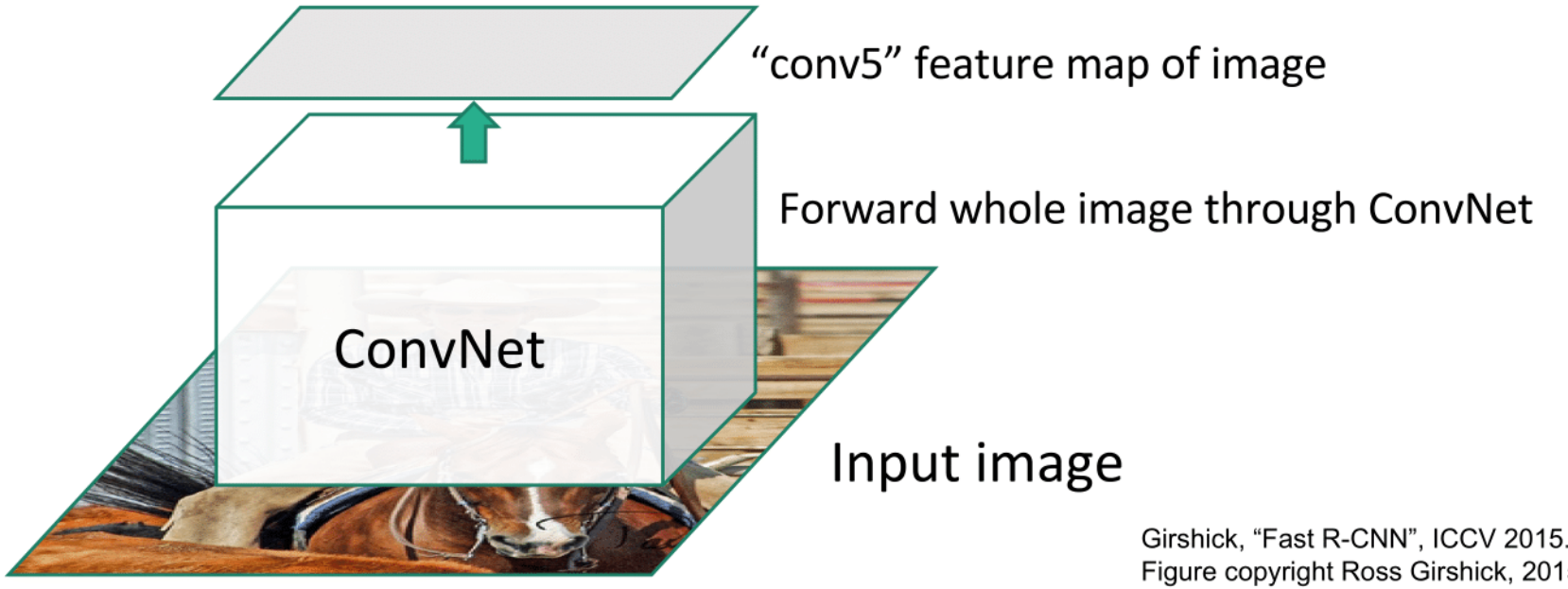
Fast R-CNN

Input image

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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

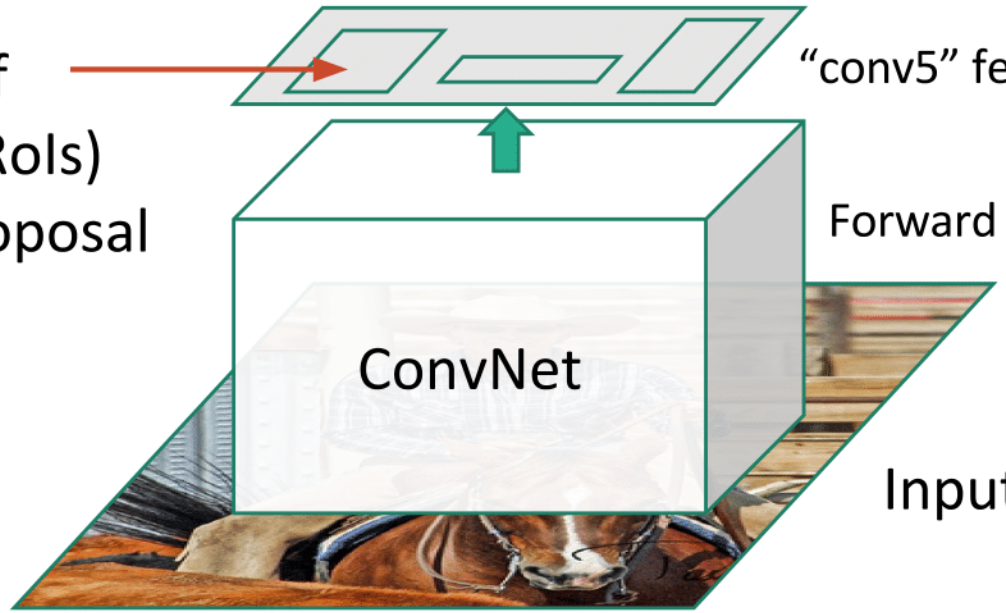
Fast R-CNN



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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

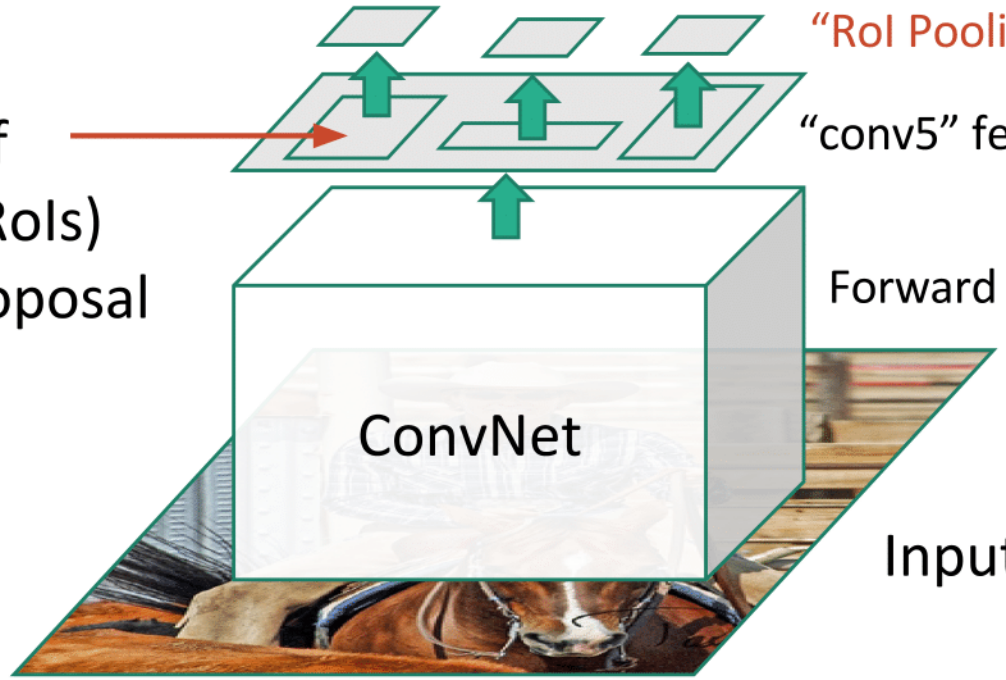
Fast R-CNN



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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

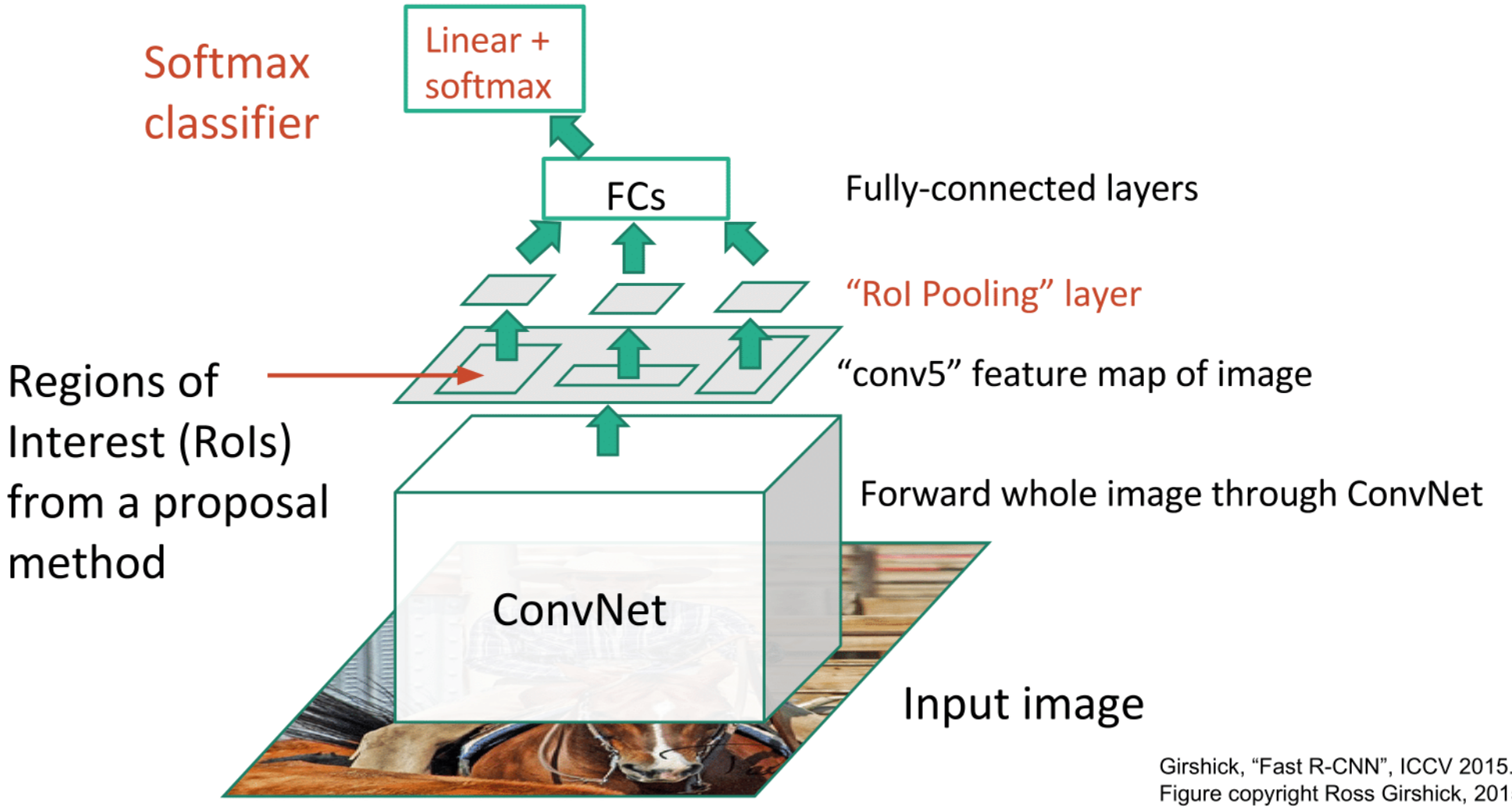
Fast R-CNN



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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

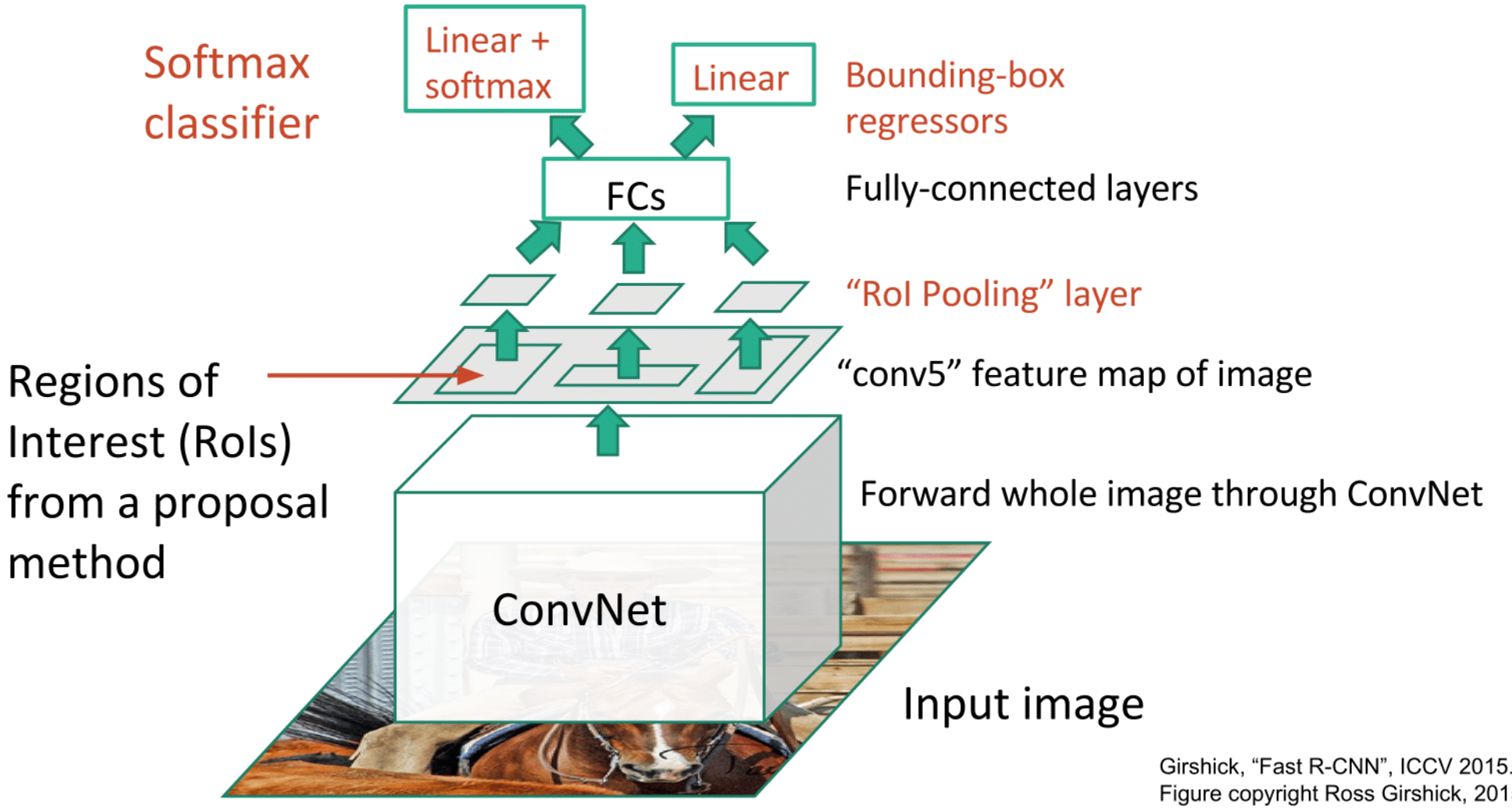
Fast R-CNN



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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

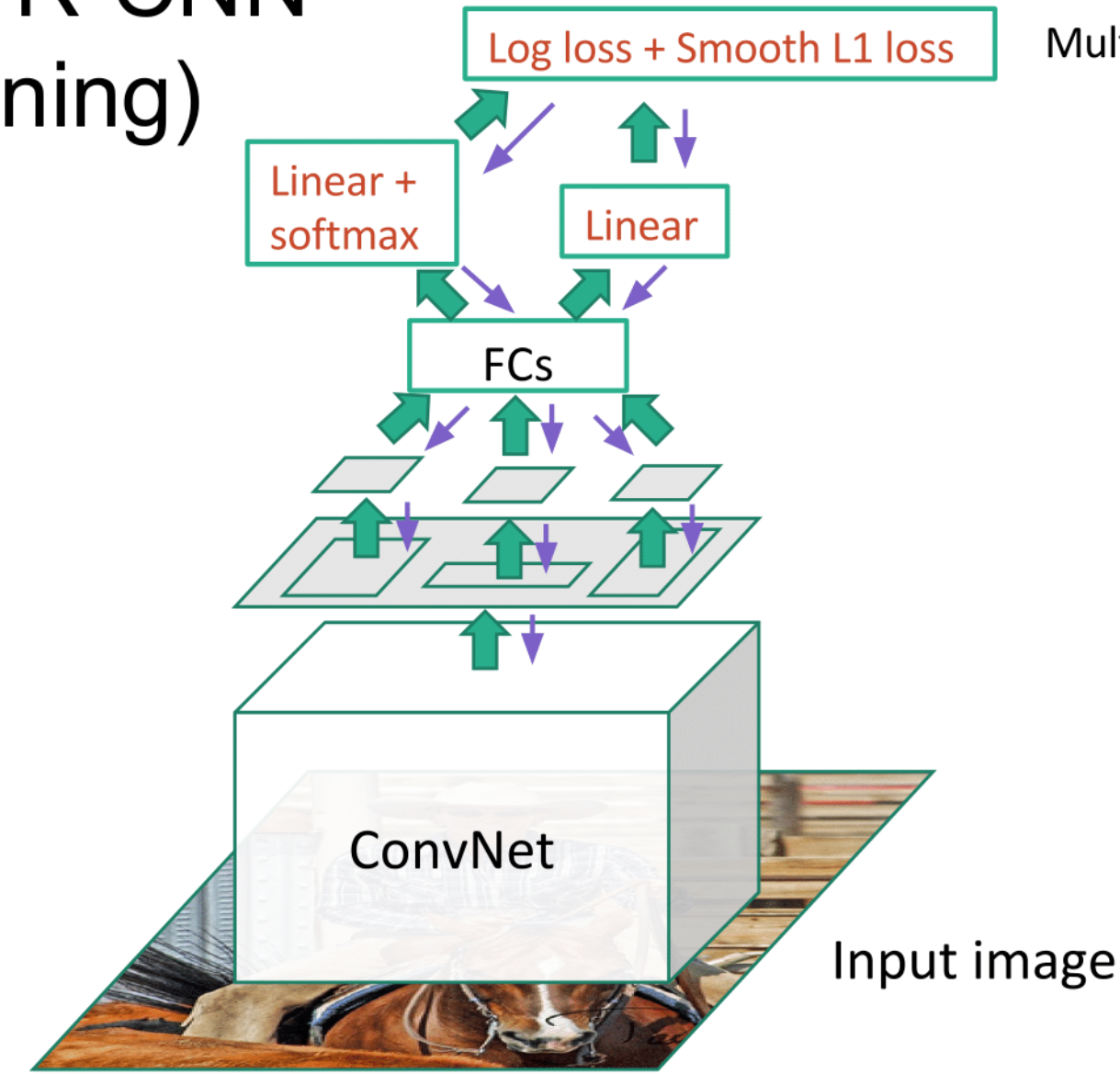
Fast R-CNN



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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Fast R-CNN (Training)

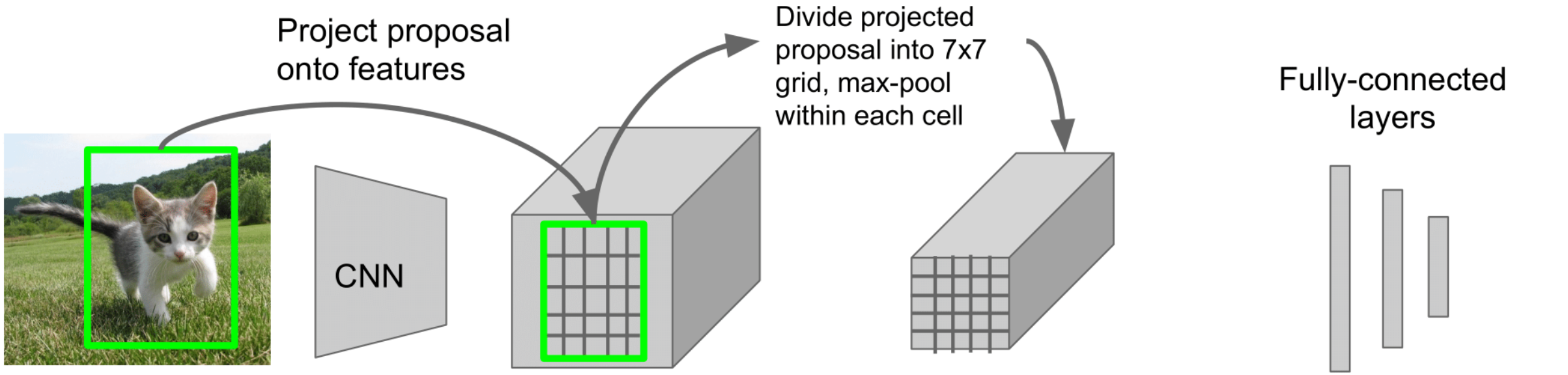


Multi-task loss

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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Faster R-CNN: RoI Pooling



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

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

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

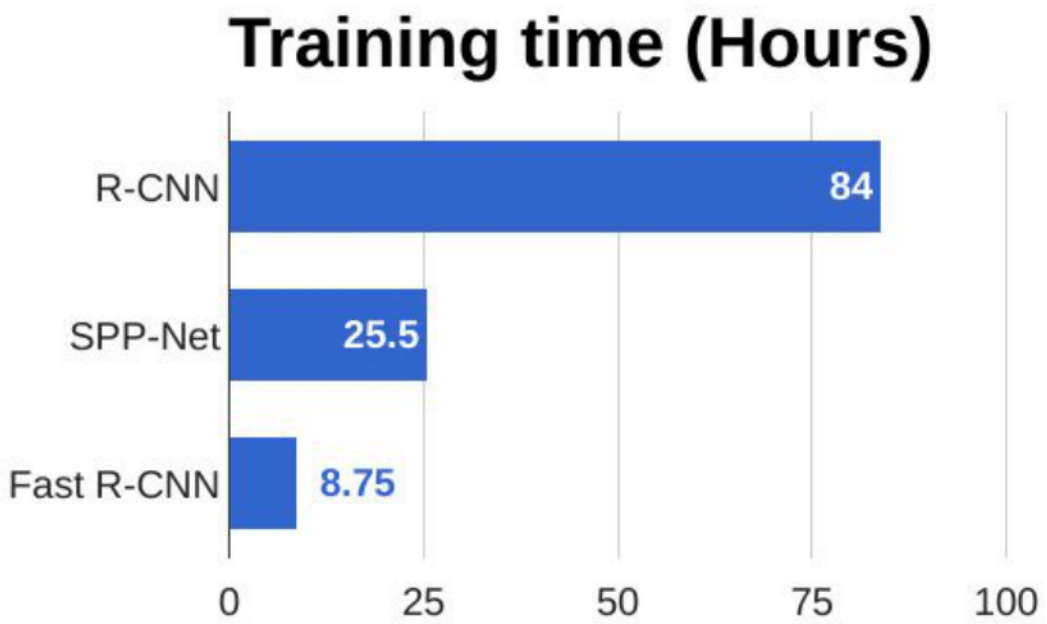
RoI conv features: 512 x 7 x 7 for region proposal

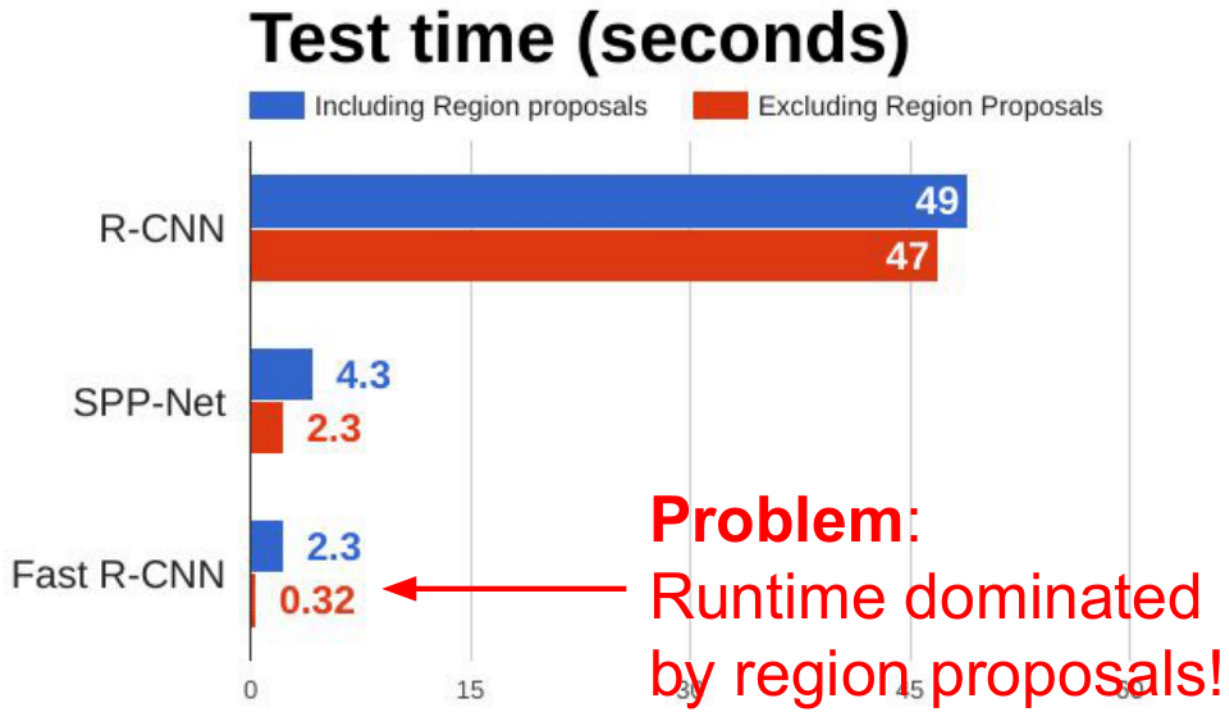
Fully-connected layers expect low-res conv features: 512 x 7 x 7

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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

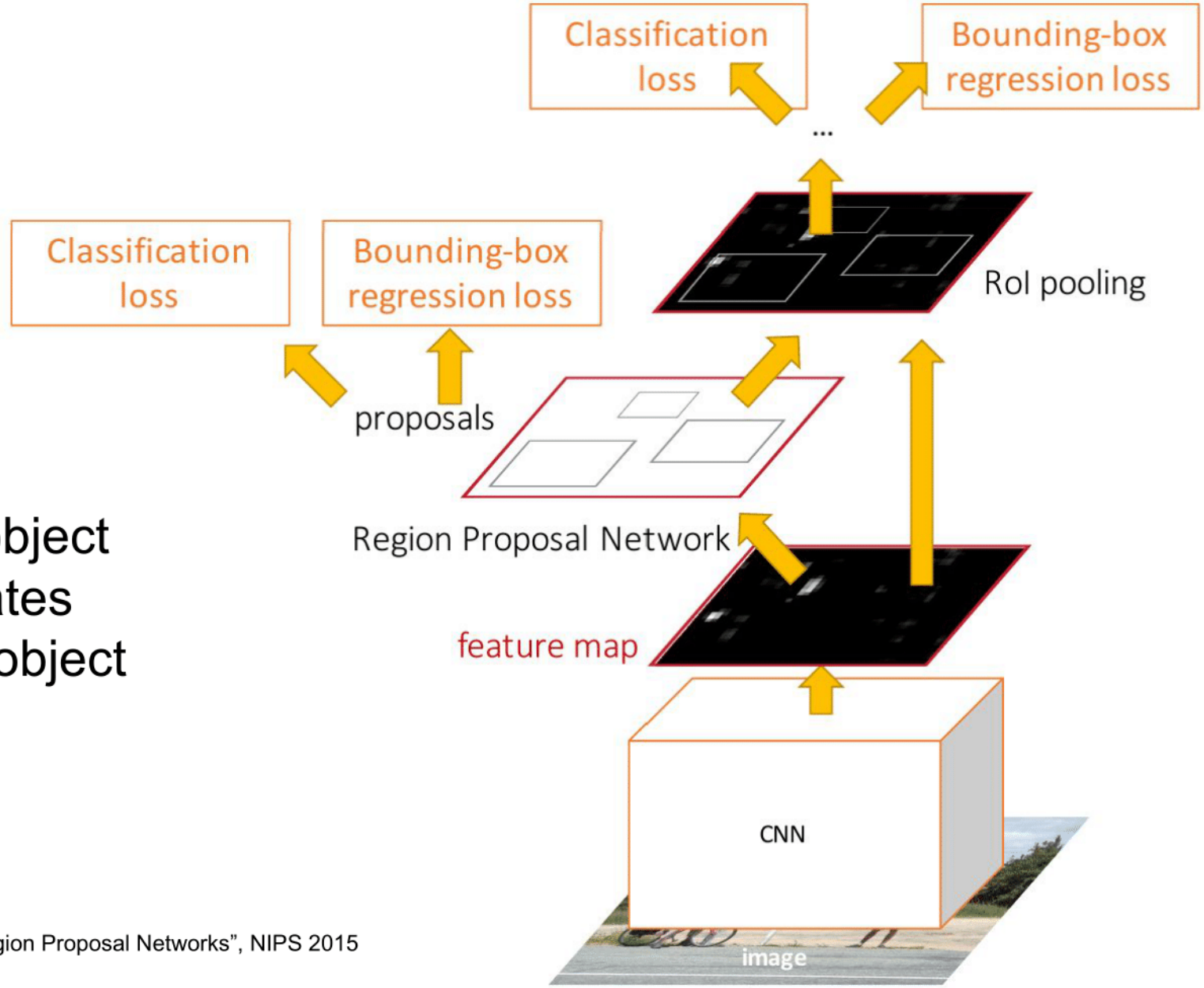
Faster R-CNN:

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

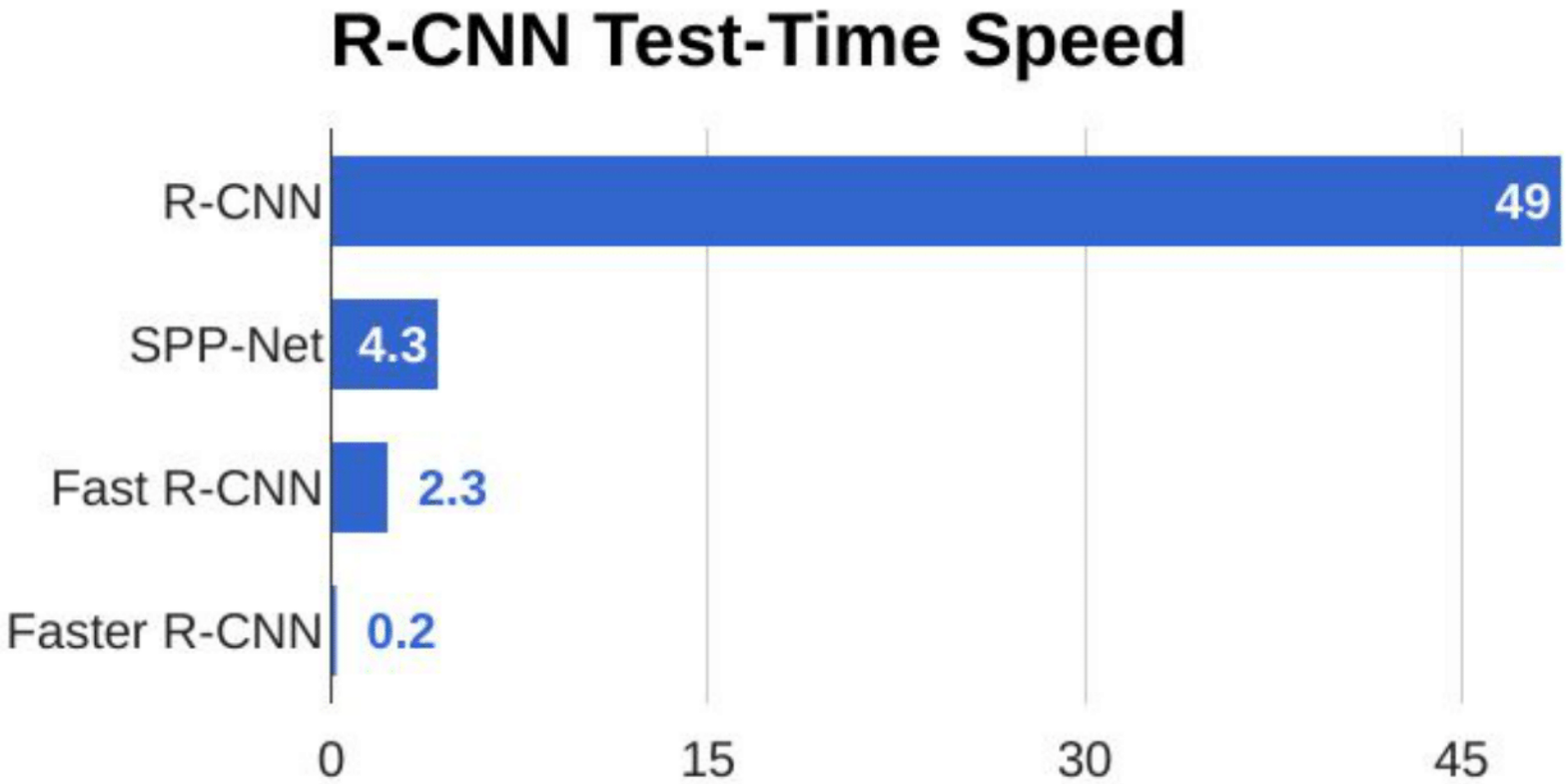


Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Faster R-CNN:

Make CNN do proposals!

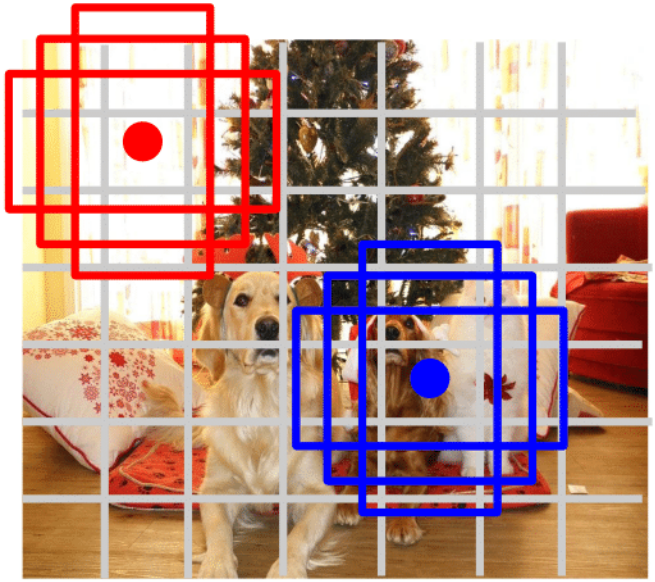


Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

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:

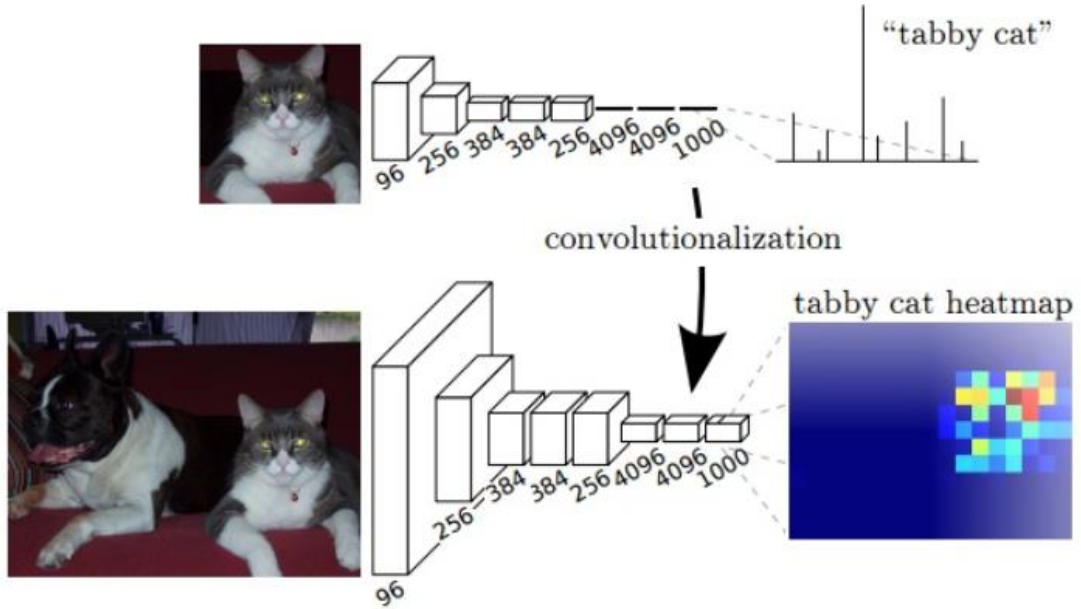
- Regress from each of the B base boxes to a final box with 5 numbers: (dx, dy, dh, dw, confidence)
- Predict scores for each of C classes (including background as a class)

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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

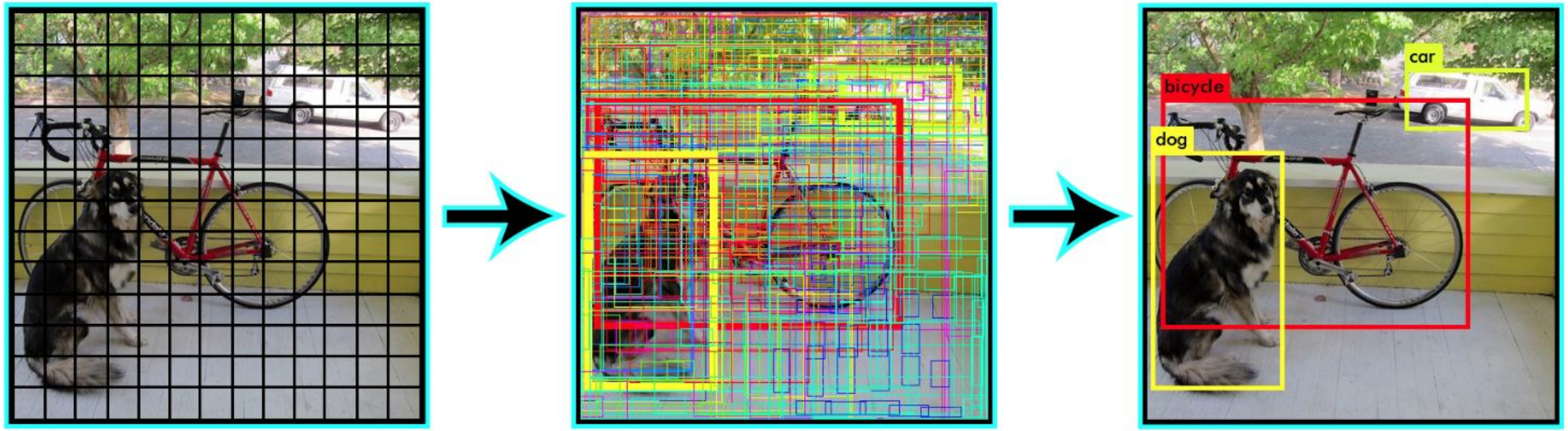
One Shot Detection



Для каждой ячейки в последнем conv слое предказываем координаты бокса и класс объекта с центром в ячейке.

slides by @Vladimir Iglovikov

One Shot Detector: YOLO



Для каждой ячейки в последнем conv слое предказываем координаты бокса и класс объекта с центром в ячейке.

slides by @Vladimir Iglovikov

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
…

Takeaways
Faster R-CNN is slower but more accurate

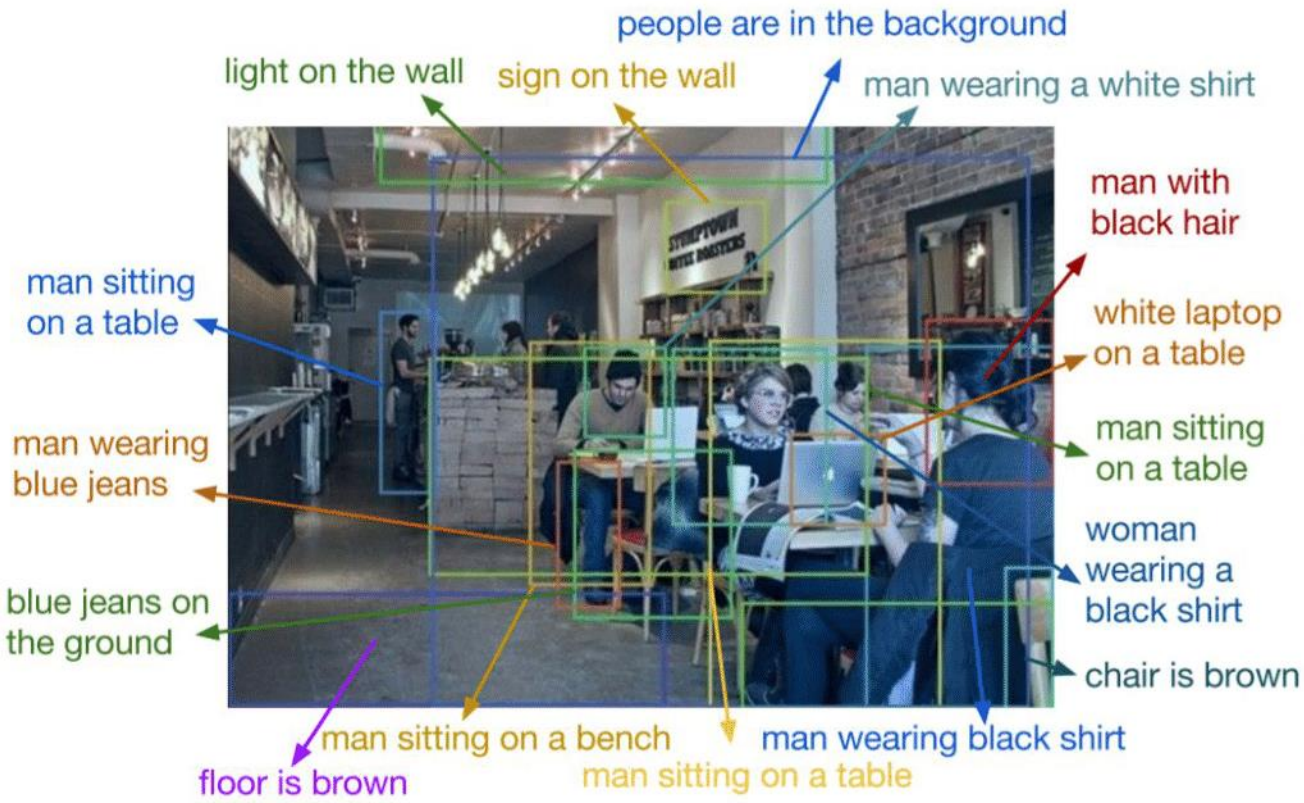
SSD is much faster but not as accurate

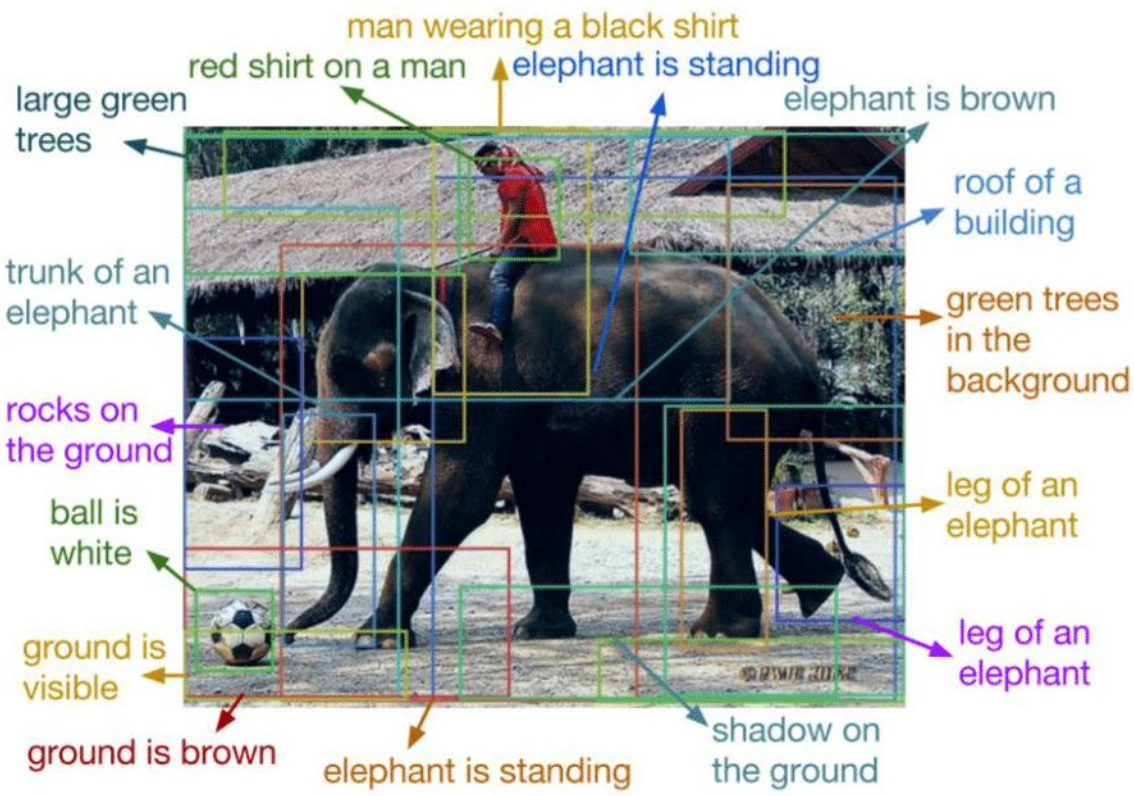
Huang et al, “Speed/accuracy trade-offs for modern convolutional object detectors”, CVPR 2017

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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Aside: Object Detection + Captioning = Dense Captioning

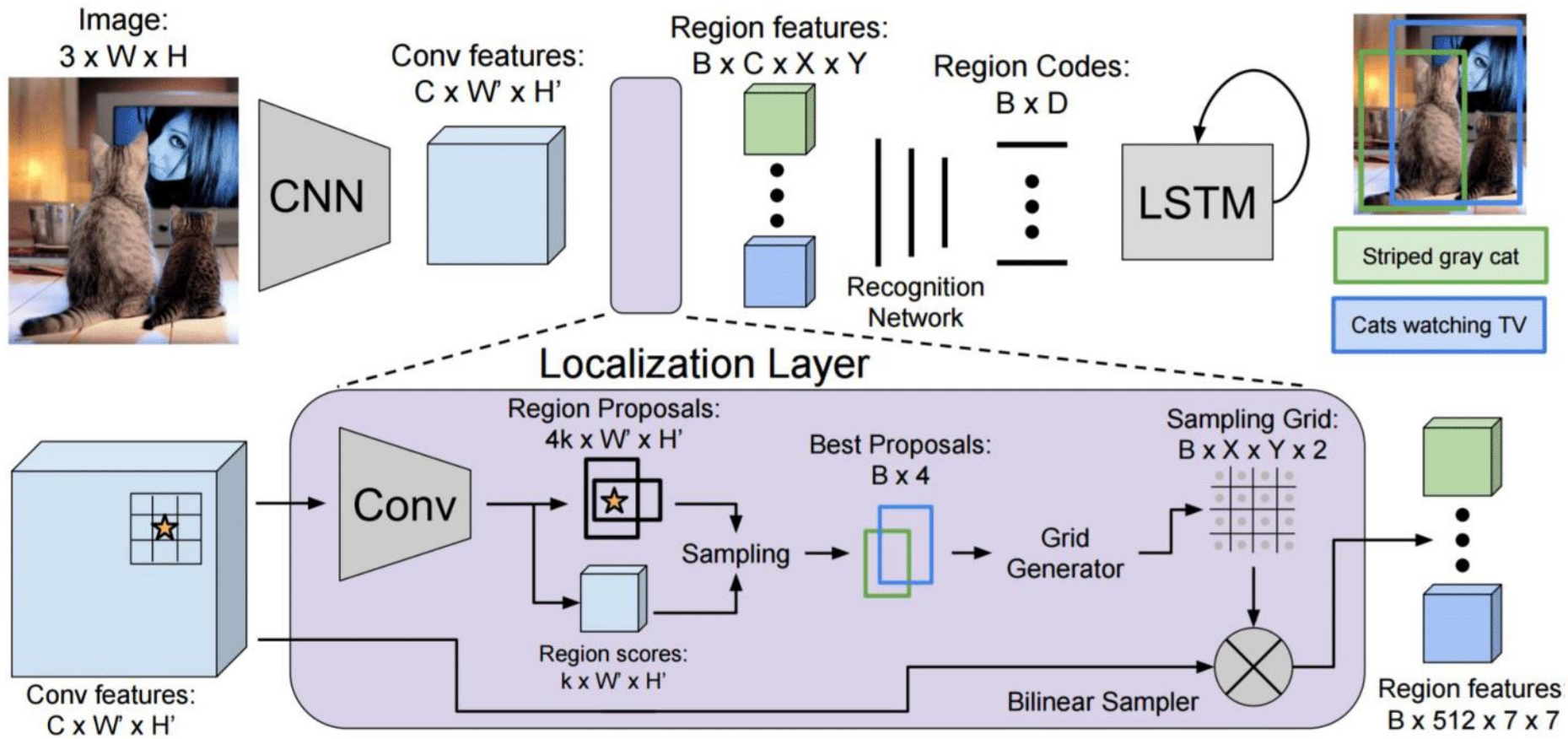




Johnson, Karpathy, and Fei-Fei, "DenseCap: Fully Convolutional Localization Networks for Dense Captioning", CVPR 2016
Figure copyright IEEE, 2016. Reproduced for educational purposes.

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

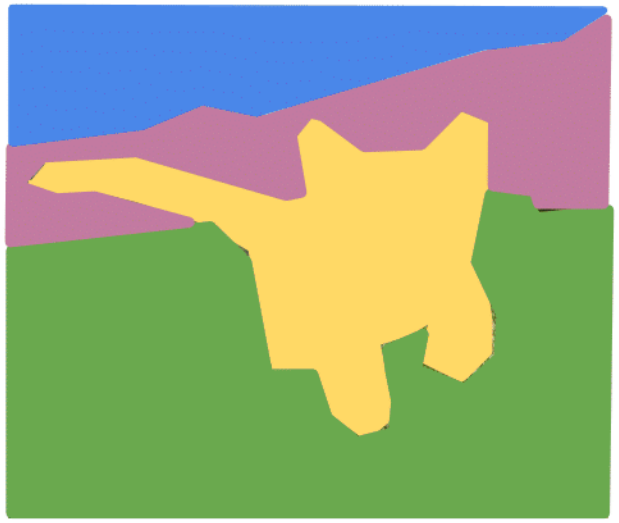
Aside: Object Detection + Captioning = Dense Captioning



Johnson, Karpathy, and Fei-Fei, "DenseCap: Fully Convolutional Localization Networks for Dense Captioning", CVPR 2016
Figure copyright IEEE, 2016. Reproduced for educational purposes.

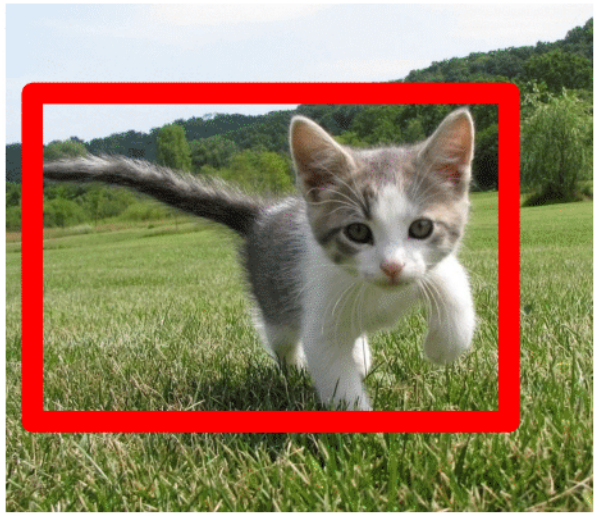
Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Instance Segmentation



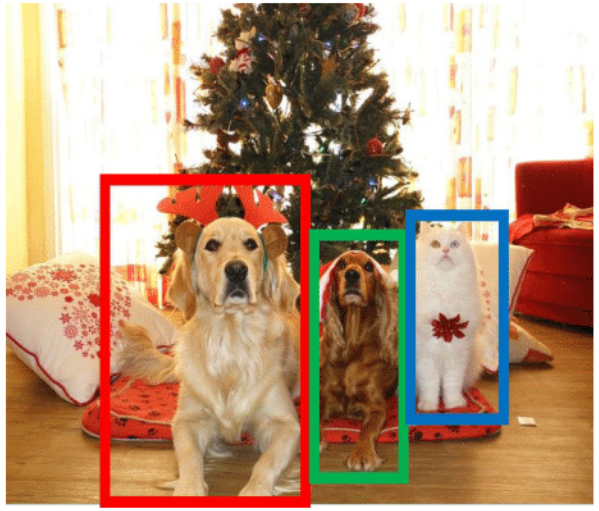
GRASS, CAT, TREE, SKY

No objects, just pixels

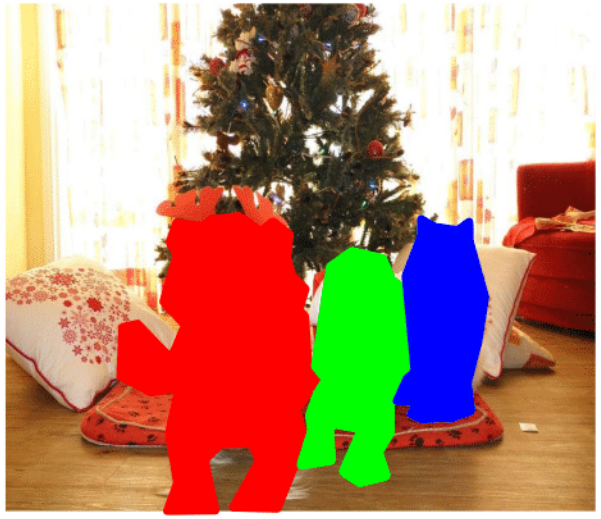


CAT

Single Object



DOG, DOG, CAT

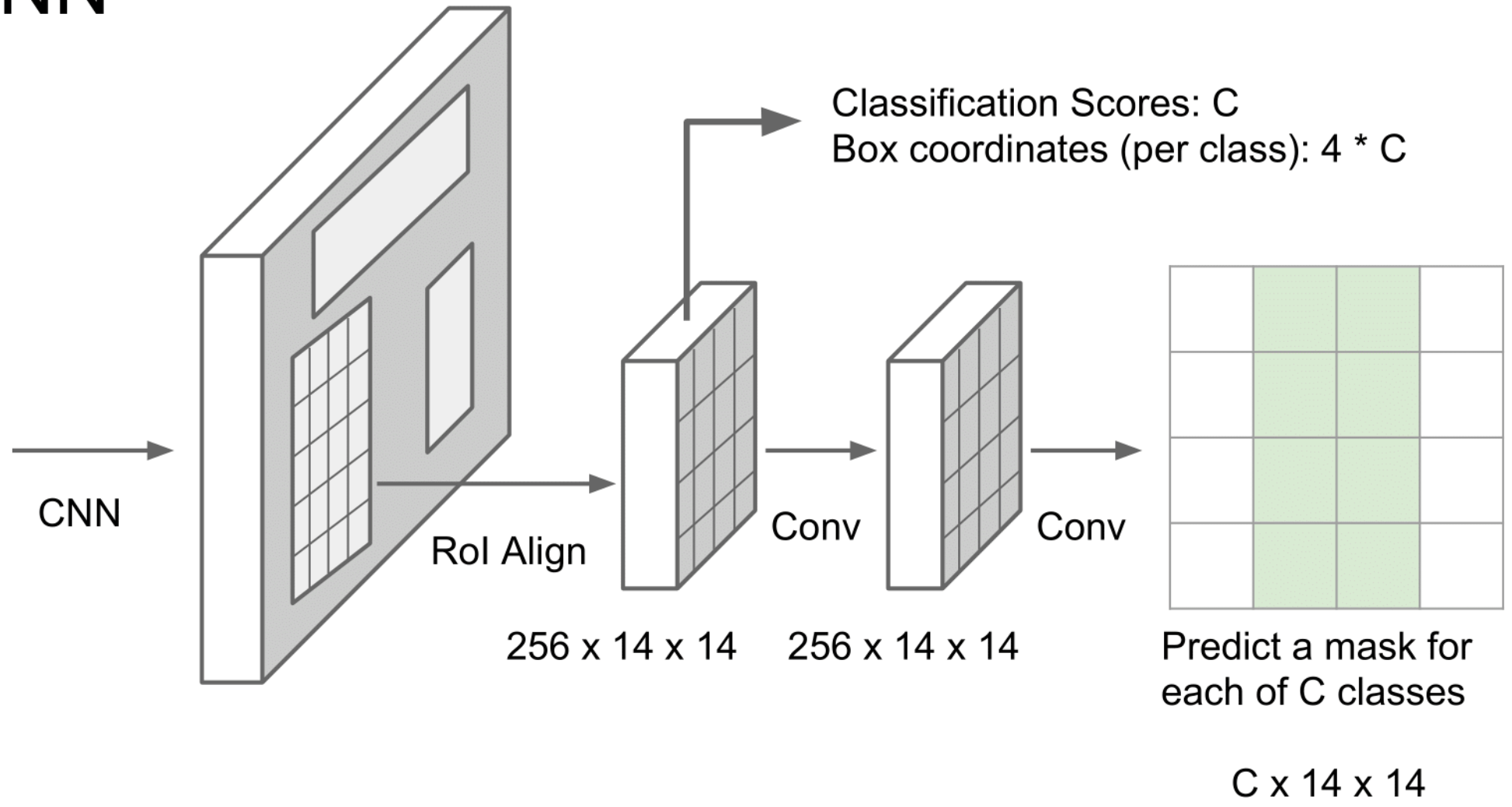


DOG, DOG, CAT

Multiple Object

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

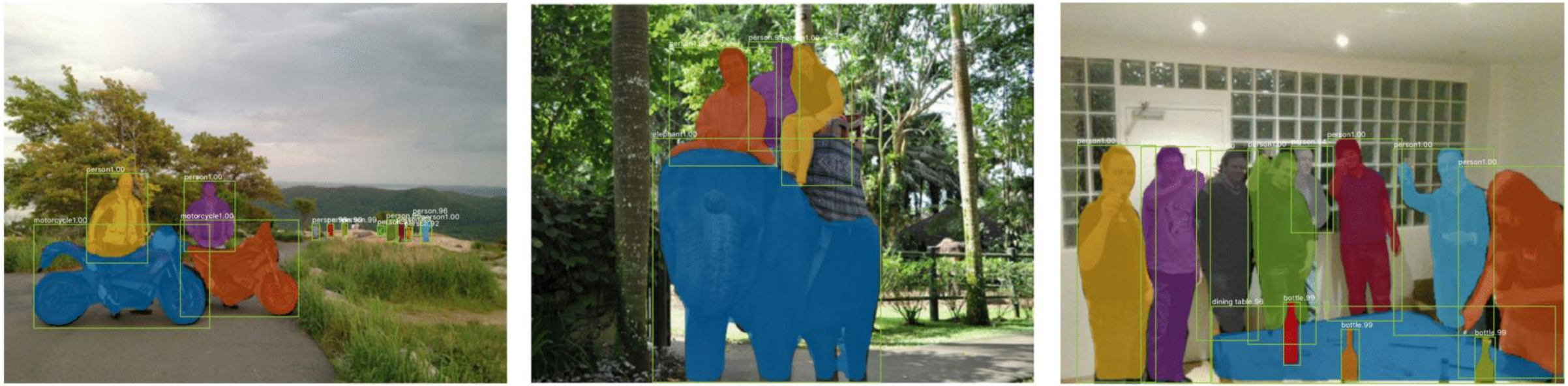
Mask R-CNN



He et al, “Mask R-CNN”, arXiv 2017

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Mask R-CNN: Very Good Results!

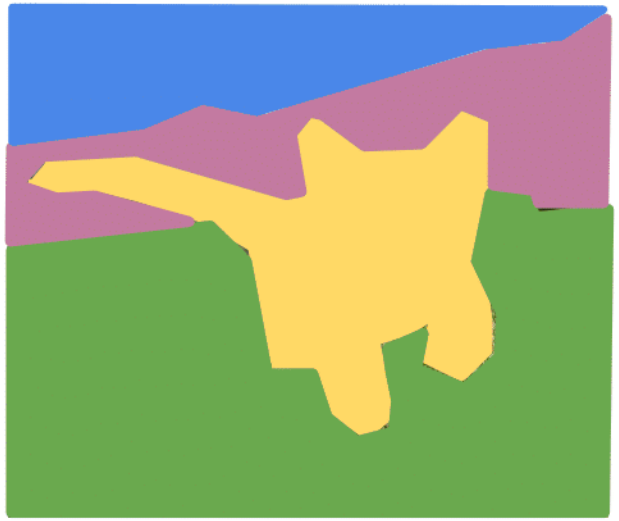


He et al, “Mask R-CNN”, arXiv 2017
Figures copyright Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, 2017.
Reproduced with permission.

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

Recap:

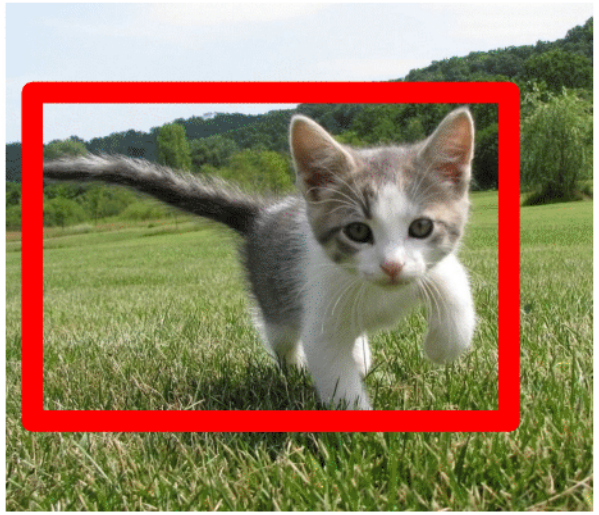
Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

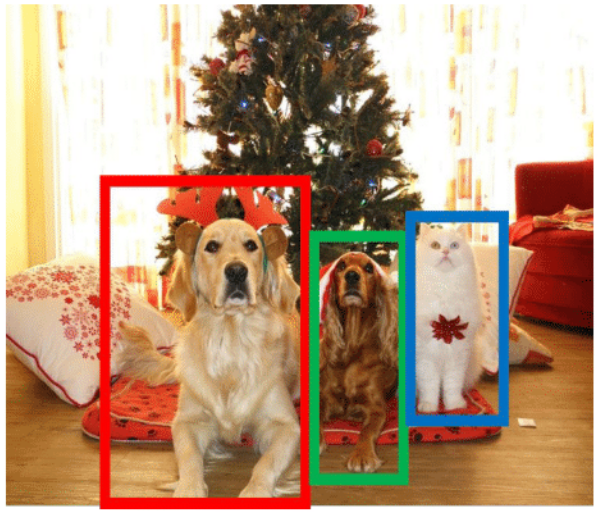
Classification + Localization



CAT

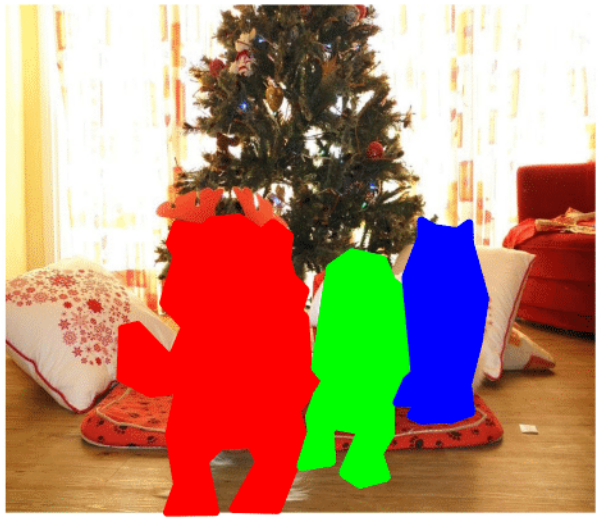
Single Object

Object Detection



DOG, DOG, CAT

Instance Segmentation



DOG, DOG, CAT

Multiple Object

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung