Сегментация и поиск объектов на изображениях

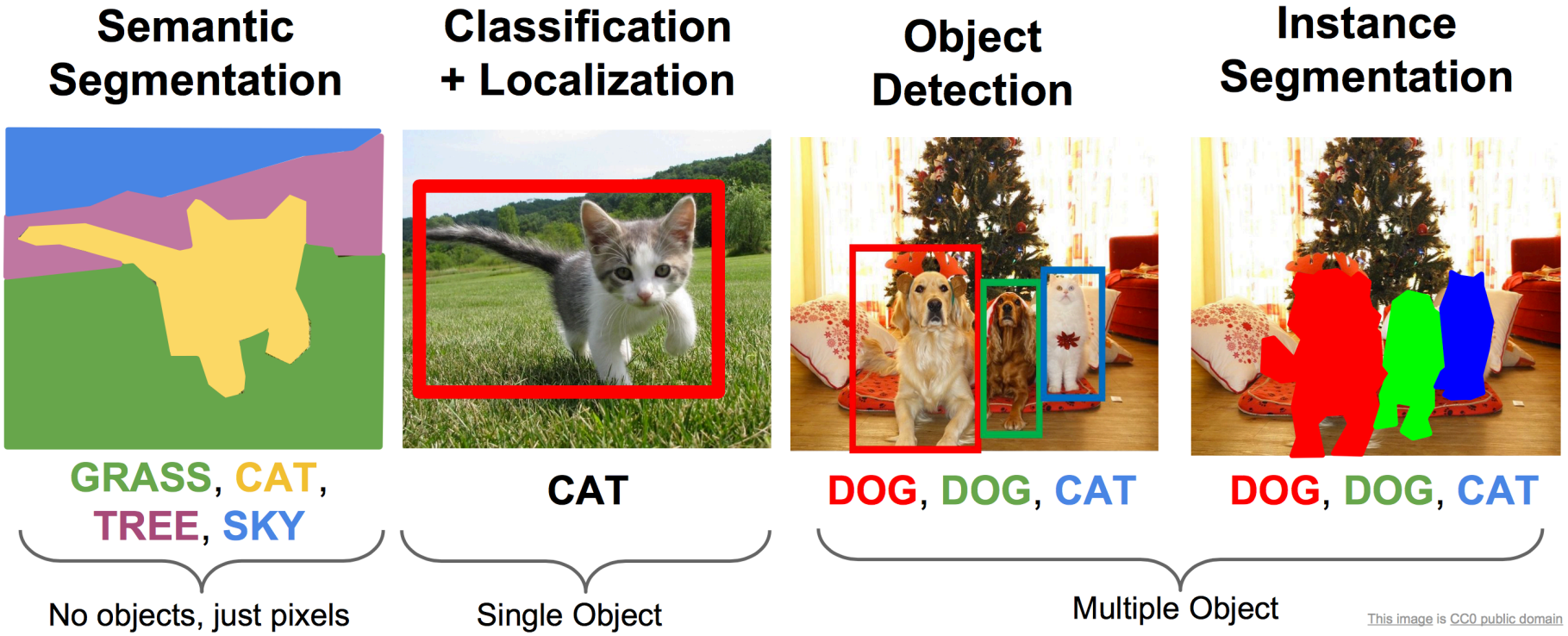
Белов Андрей

ФИВТ МФТИ, ABBYY

План лекции

- Обзор задач
- Датасеты
- Semantic Segmentation
- Object Localization
- Object Detection
- Instance Segmentation

Обзор задач



Источник: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf

Датасеты

MS COCO

http://cocodataset.org/#overview

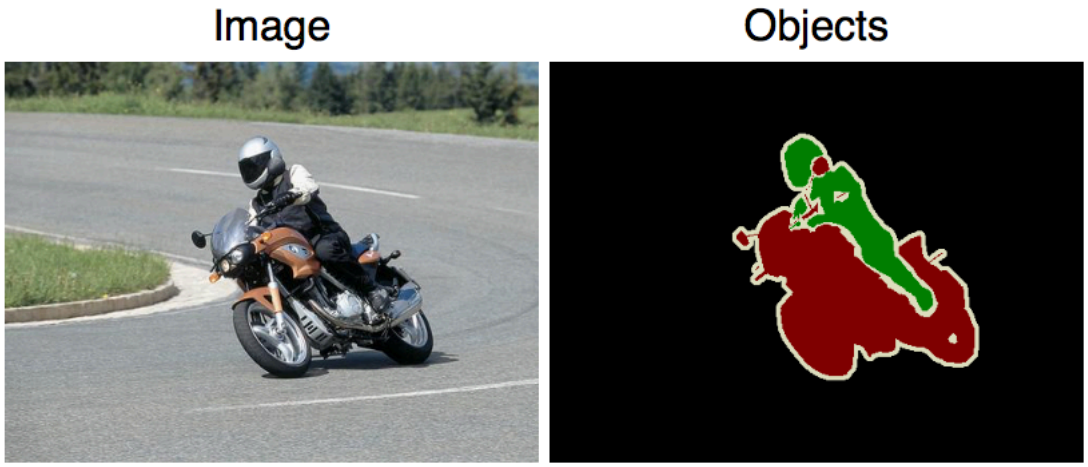
Instances:

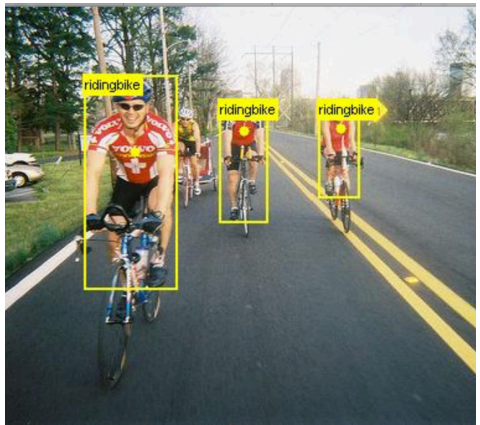
Stuff:

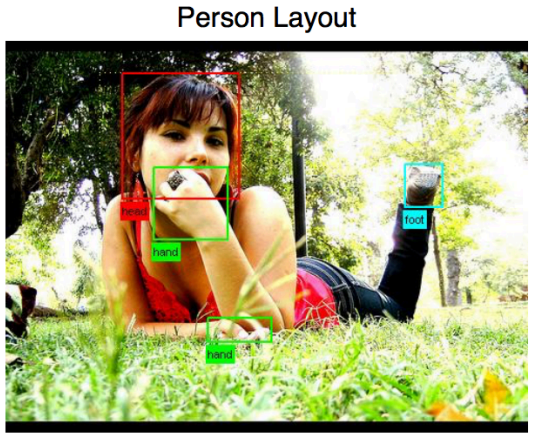
Keypoints:

PASCAL VOC

http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html







Датасеты

COCO

http://cocodataset.org/#overview

Сегментация + Keypoints для людей

Pascal Voc

http://host.robots.ox.ac.uk/pascal/VOC/

ImageNet

http://image-net.org/challenges/LSVRC/2017/index#comp

Object localization + Objects Detection (в основном используется для классификации)

Places

http://placeschallenge.csail.mit.edu

CityScapes

https://www.cityscapes-dataset.com

CamVid

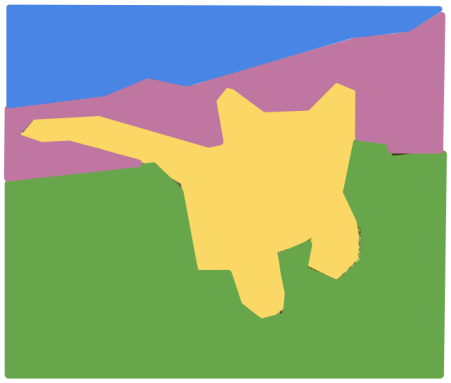
http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/

SpaceNet

https://spacenetchallenge.github.io

И еще тонны на Kaggle и других площадках

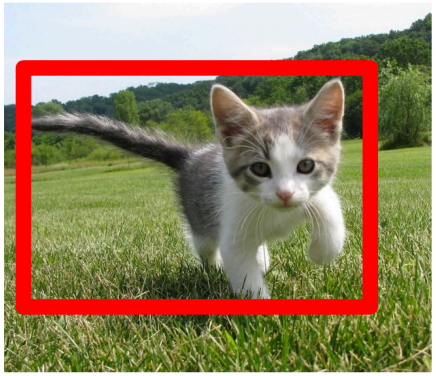
Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

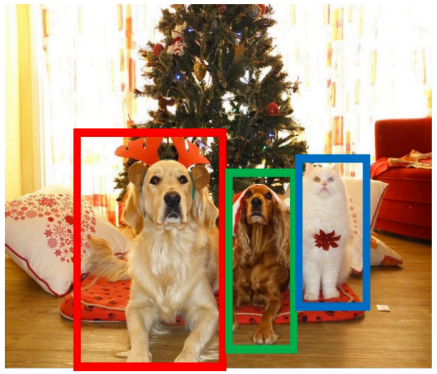
Classification + Localization



CAT

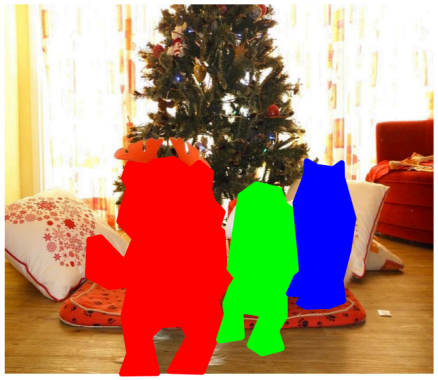
Single Object

Object Detection



DOG, DOG, CAT

Instance Segmentation



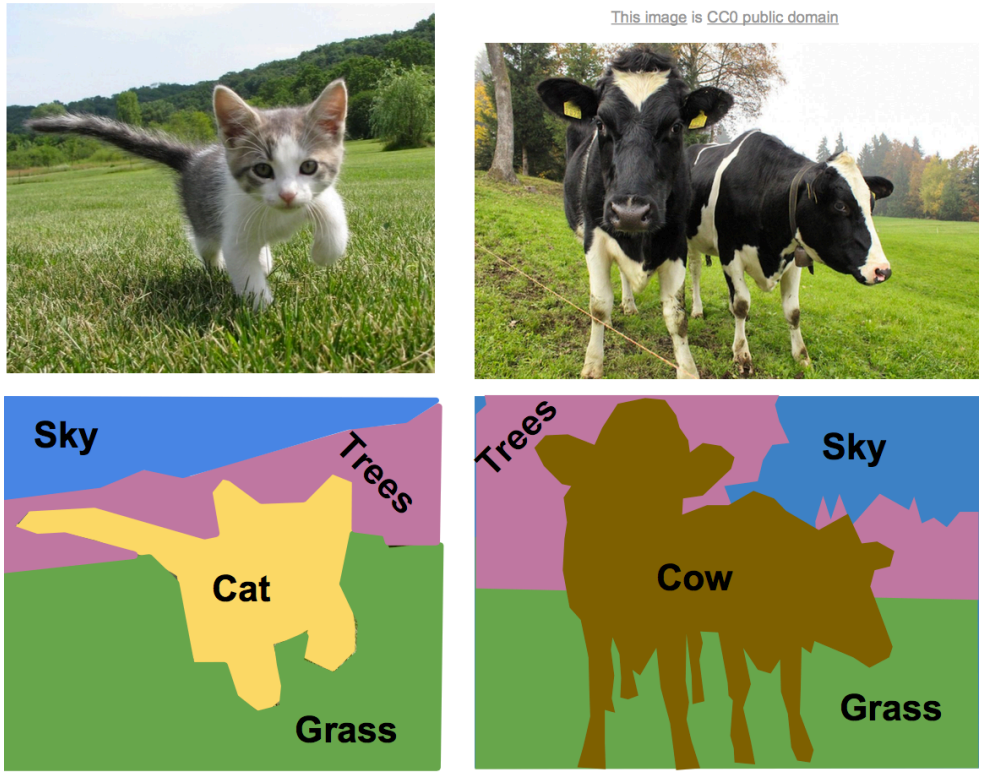
DOG, DOG, CAT

Multiple Object

Semantic Segmentation

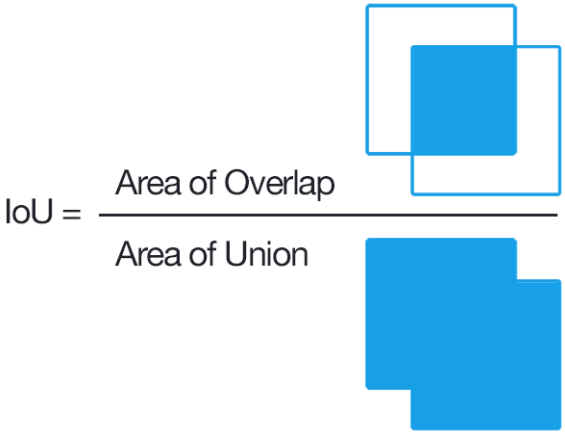
Размечаем каждый пиксель изображения

Не разделяем разные объекты одного класса



Основные метрики

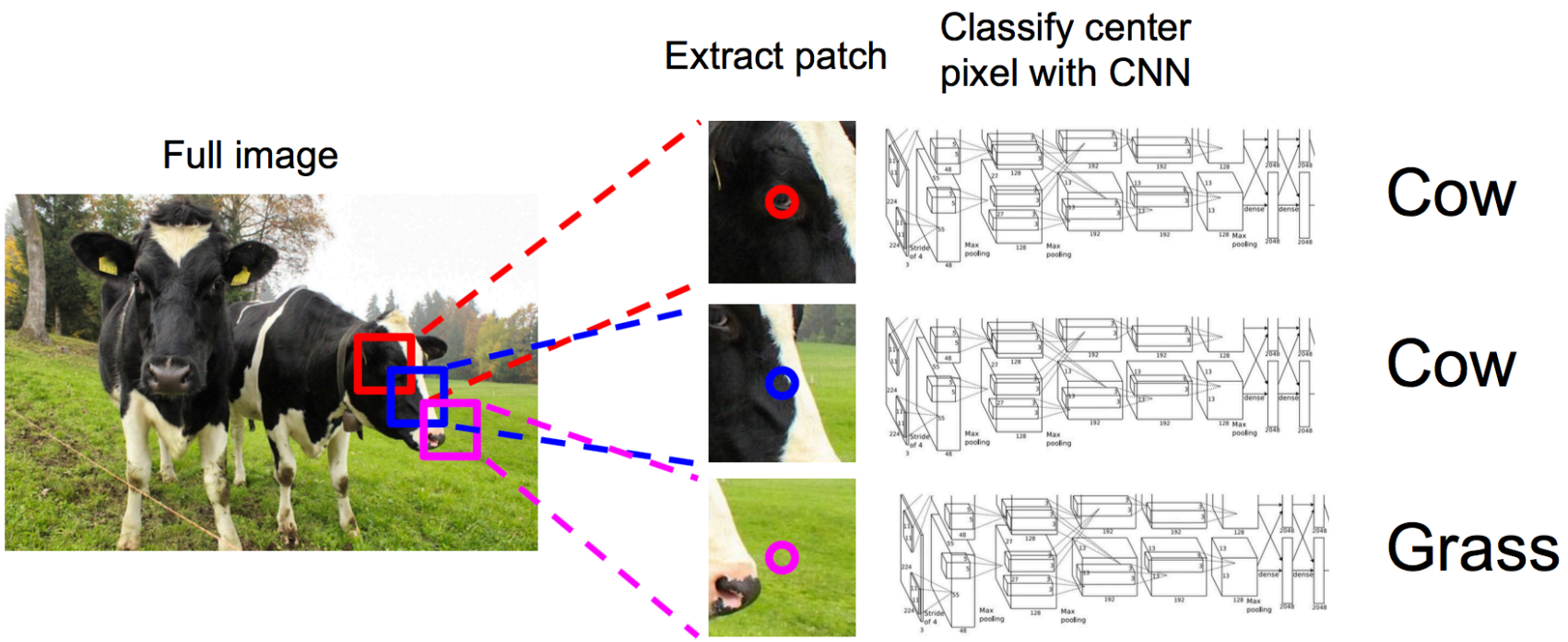
- Intersection over Union



- Dice coefficient

$$QS = \frac{2|X \cap Y|}{|X| + |Y|}$$

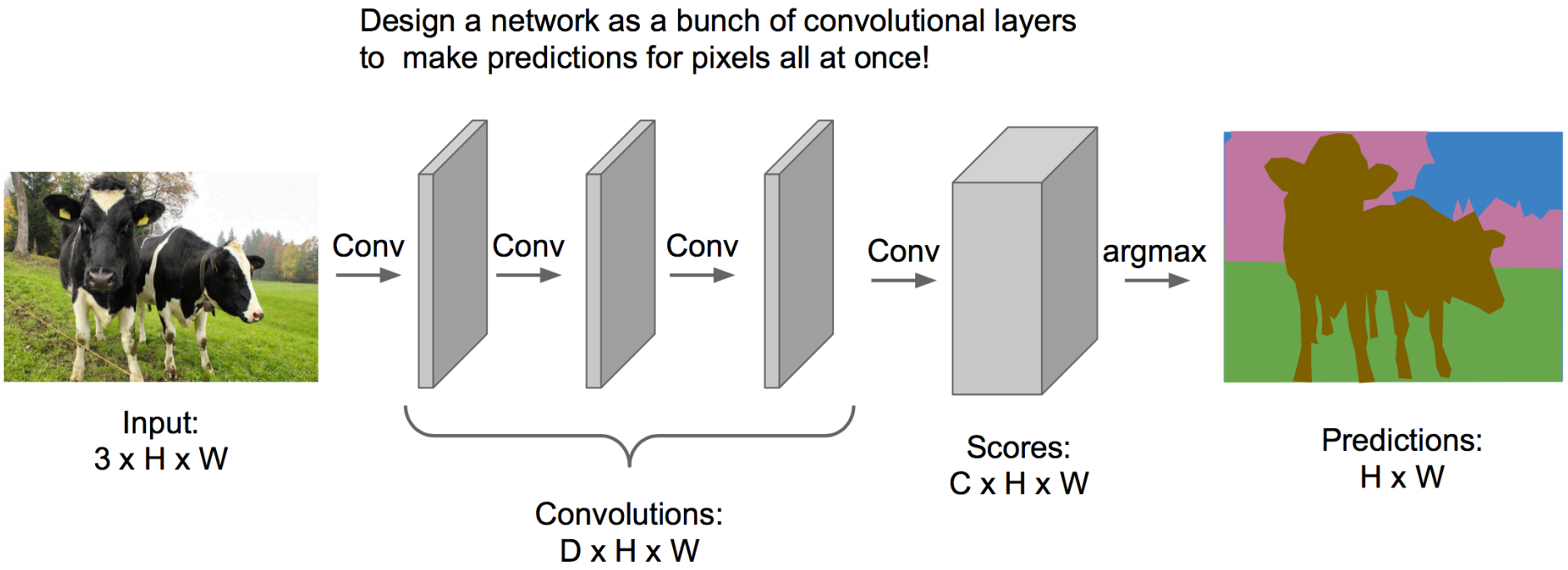
Сегментация: идея 1



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

Неэффективно, т.к. признаки для пересекающихся фрагментов каждый раз вычисляются заново.

Сегментация: идея 2

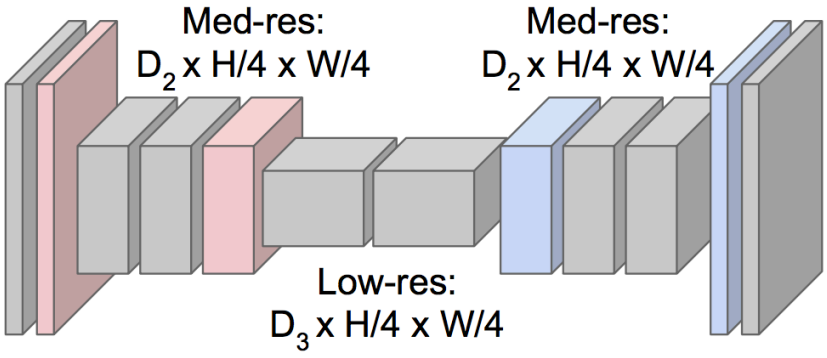


Очень много вычислений, если применять свертки на полном разрешении изображения.

Сегментация: идея 3

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

Input:
3 x H x W



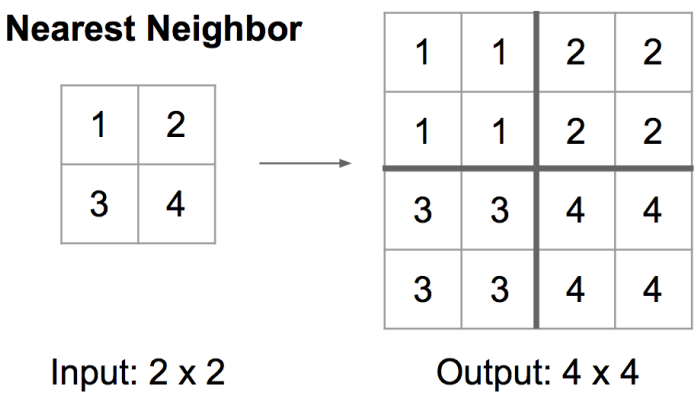
High-res:
D_1 x H/2 x W/2

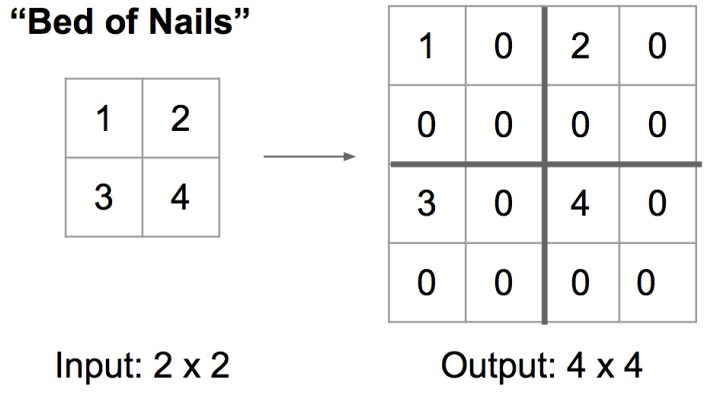
High-res:
D_1 x H/2 x W/2

Predictions:
H x W

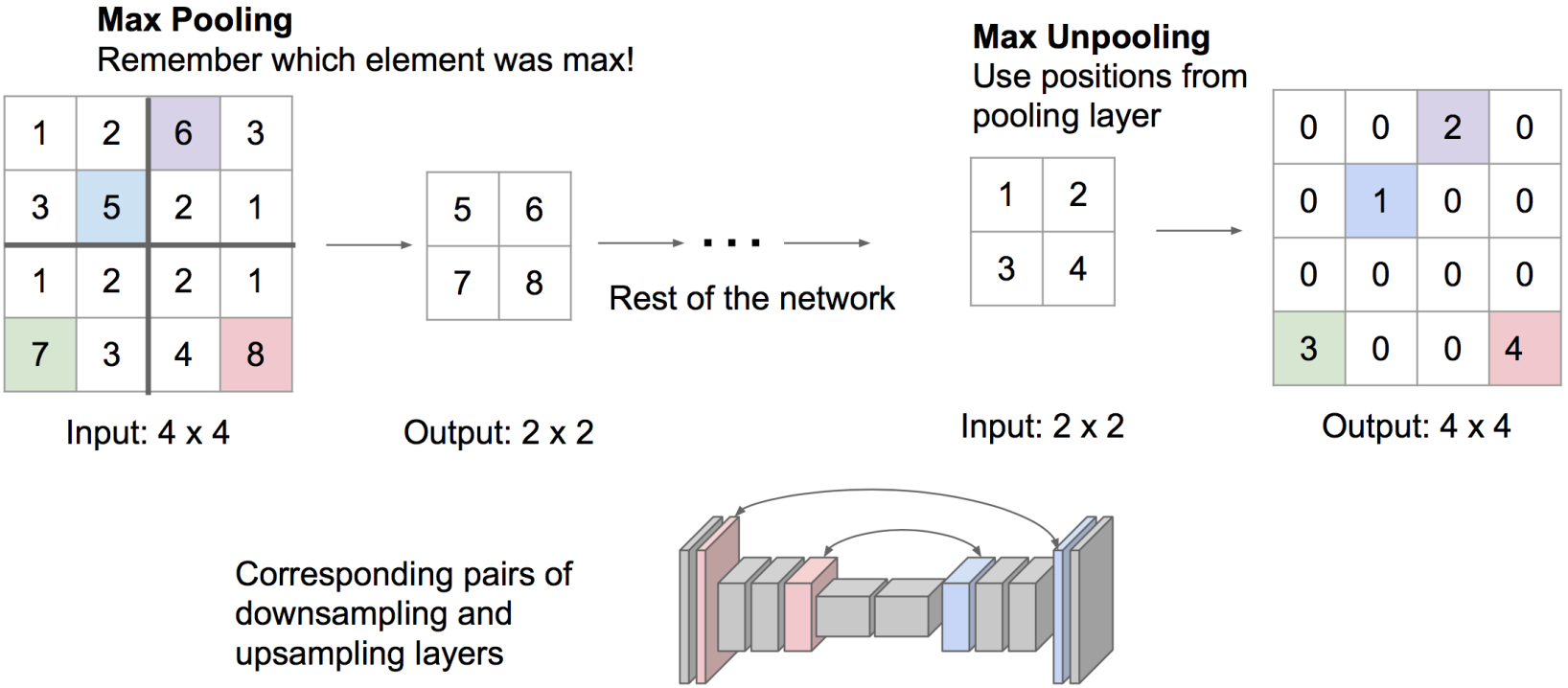
Downsampling – pooling или strided convolutions
Но как делать upsampling?

Unpooling

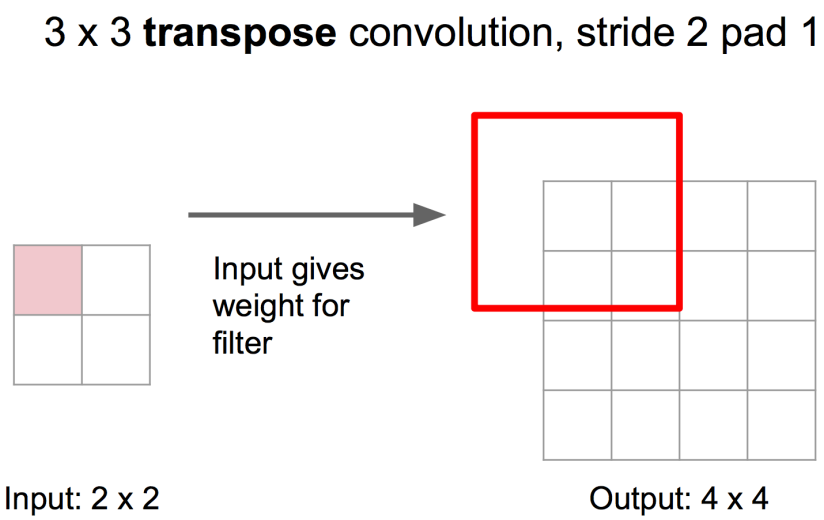

Nearest Neighbor
1 2
3 4
1 1 2 2
1 1 2 2
3 3 4 4
3 3 4 4
Input: 2 x 2
Output: 4 x 4


“Bed of Nails”
1 2
3 4
1 0 2 0
0 0 0 0
3 0 4 0
0 0 0 0
Input: 2 x 2
Output: 4 x 4

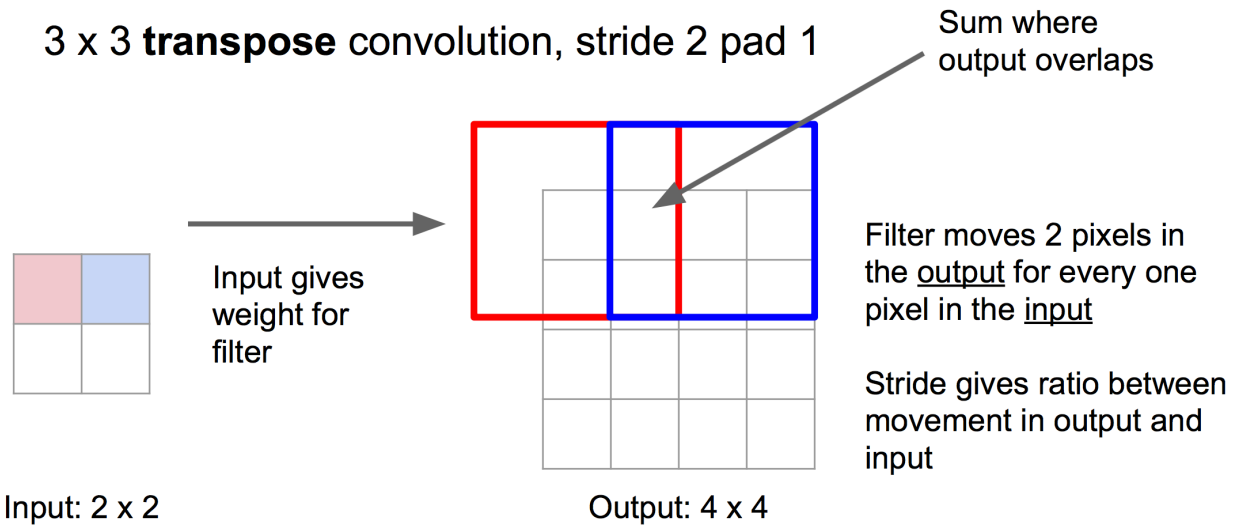
Max-unpooling



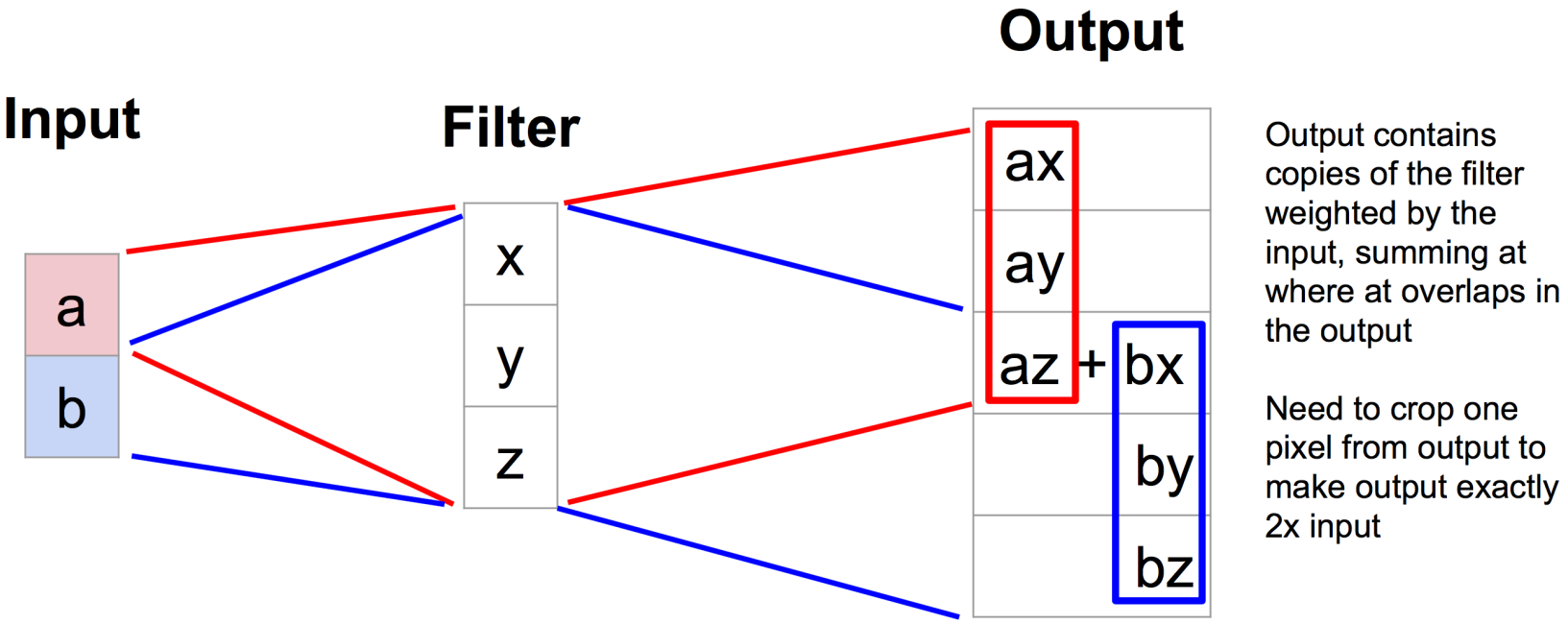
Transposed convolutions



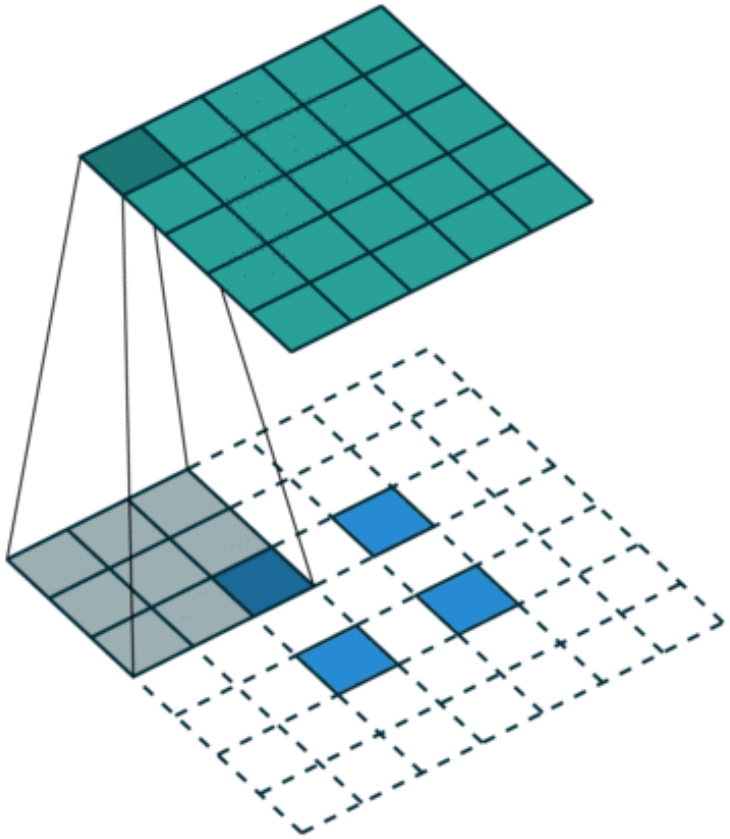
Transposed convolutions



Transposed convolutions. Одномерный пример



Суммирование может приводить к появлению решетки, поэтому иногда используют свертки kernel 2 stride 2 или kernel 4 stride 2



Архитектуры: U-Net

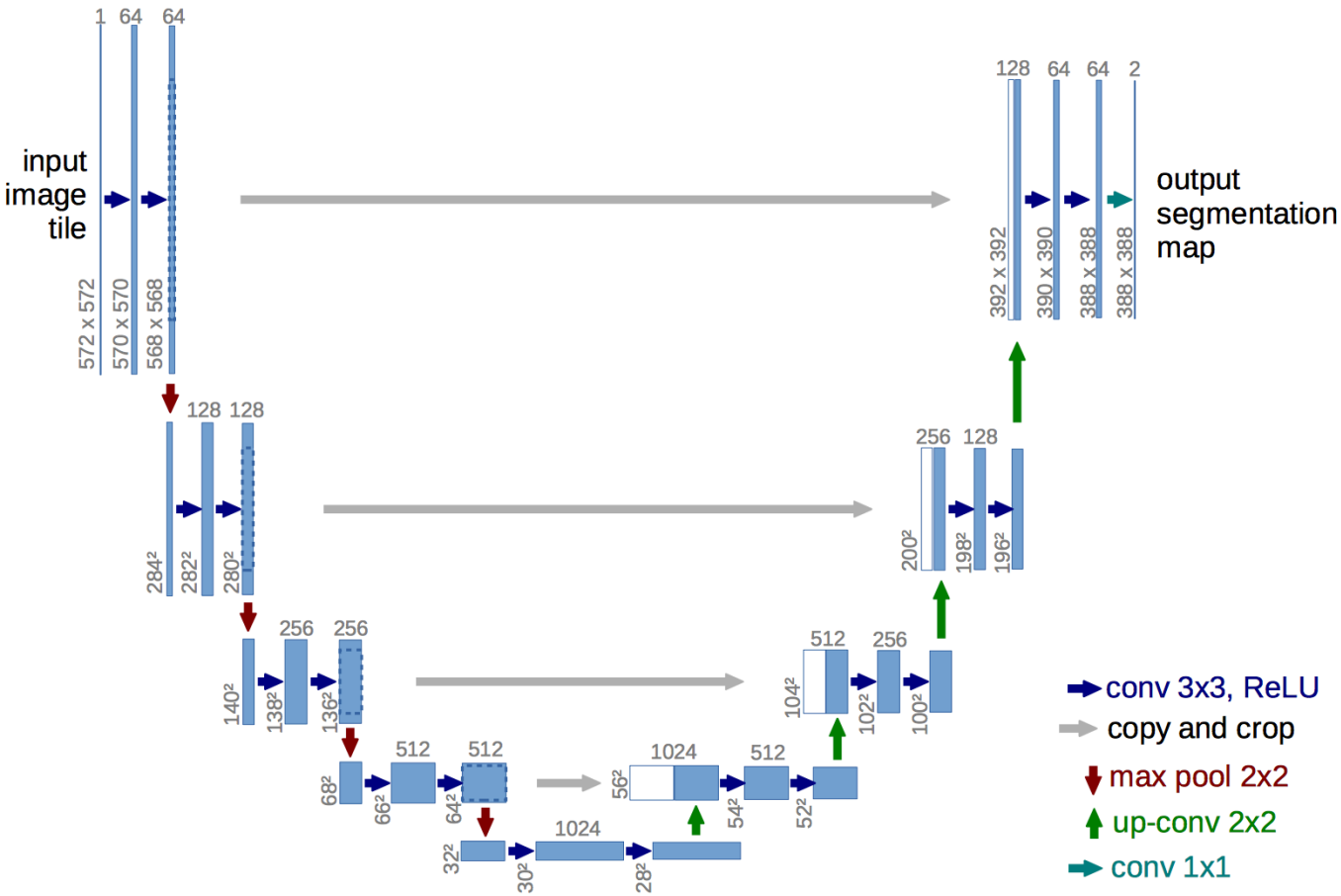


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

https://arxiv.org/abs/1505.04597

FCN - Fully convolutional networks for semantic segmentation

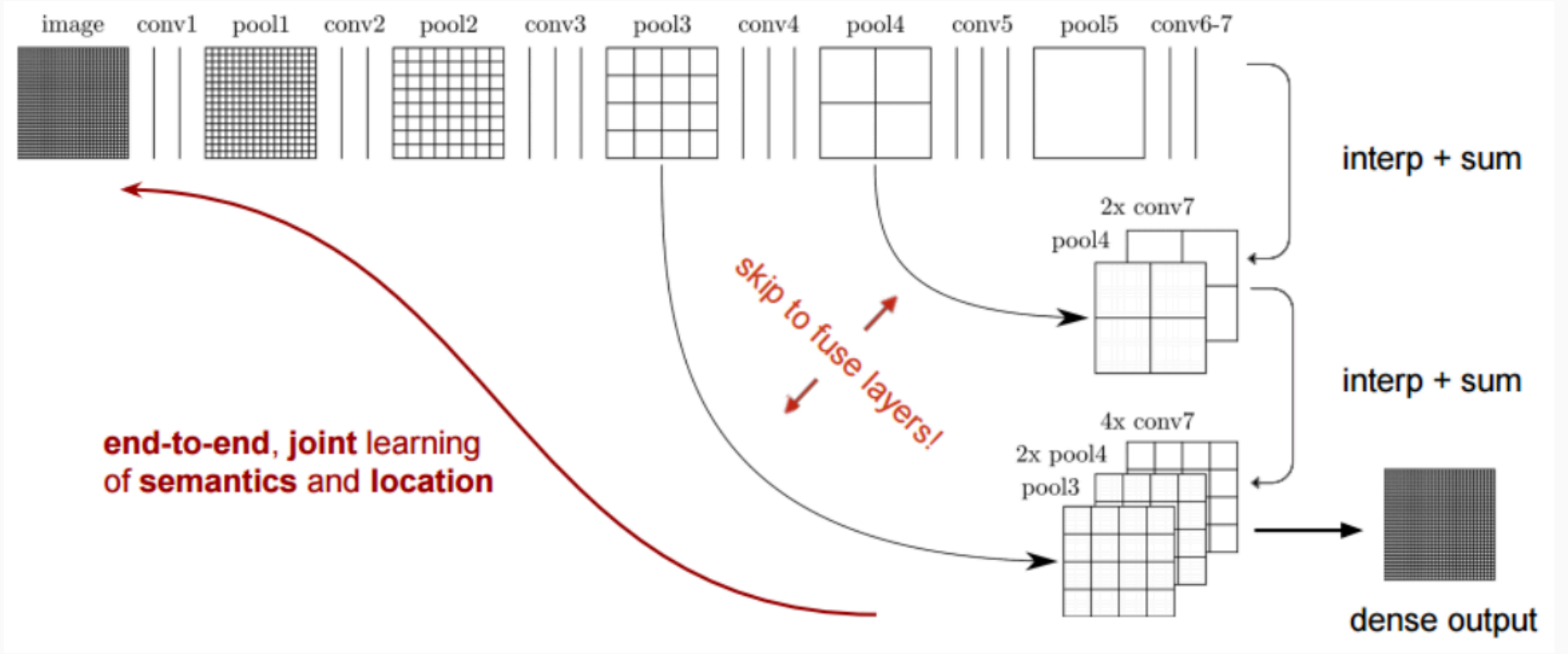


Figure : The FCN-32s Architecture

https://arxiv.org/abs/1411.4038

SegNet

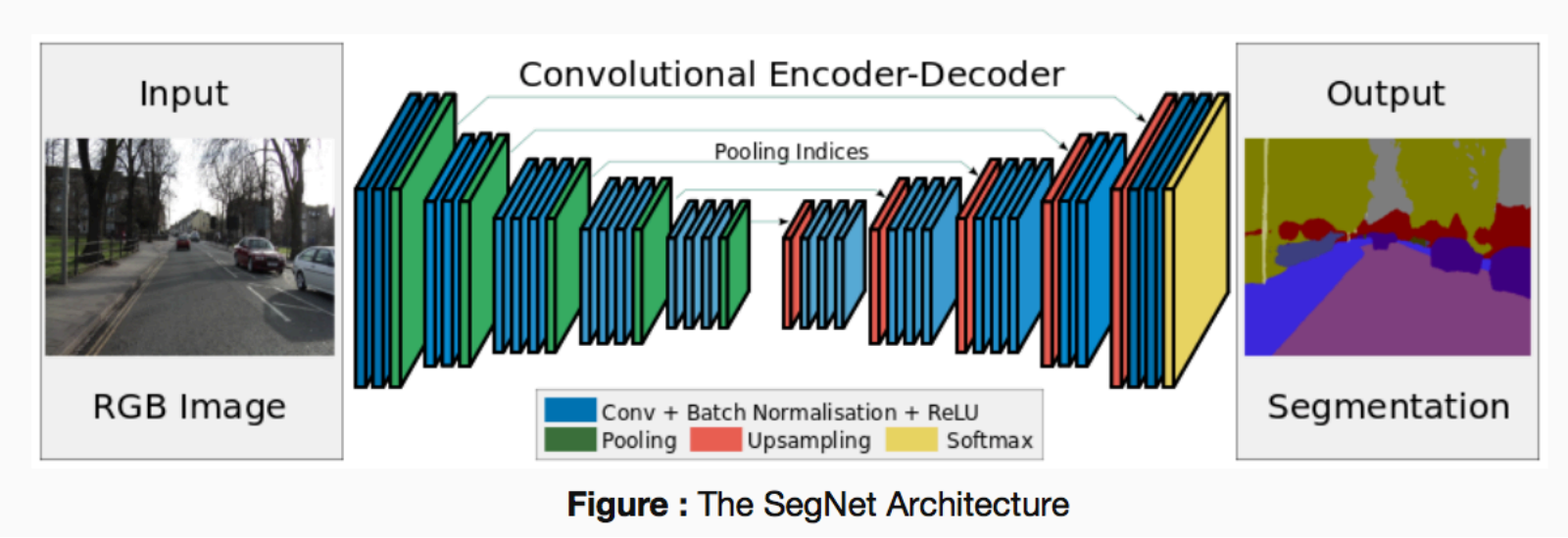
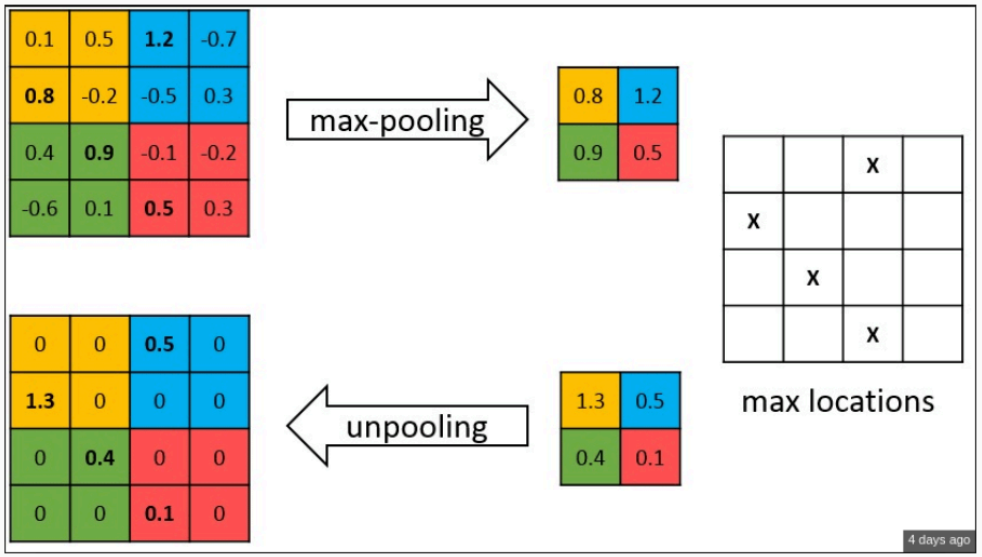


Figure : The SegNet Architecture

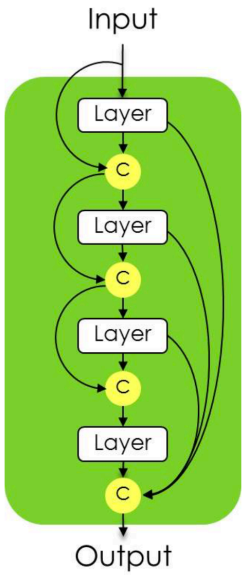
Для повышения размерности используется max-unpooling

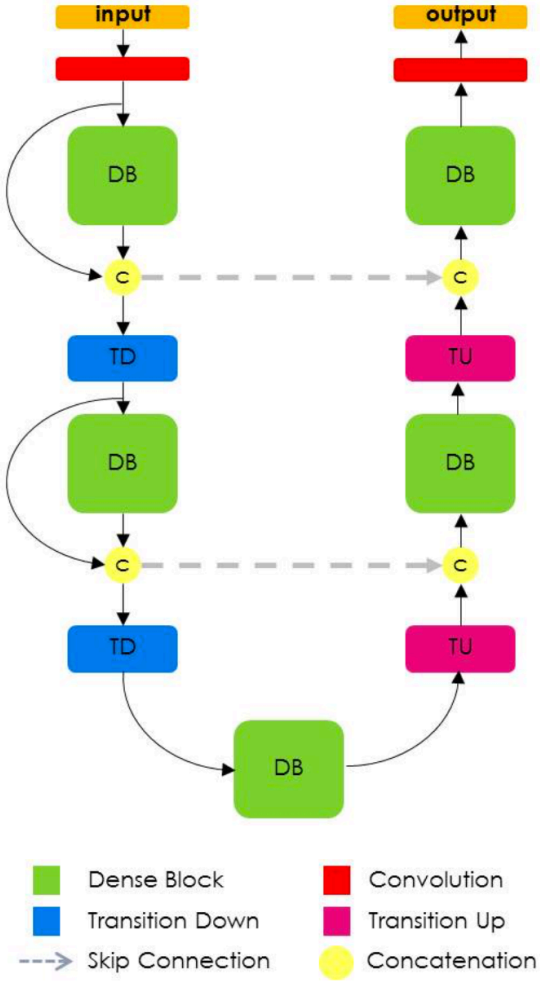


https://arxiv.org/abs/1511.00561

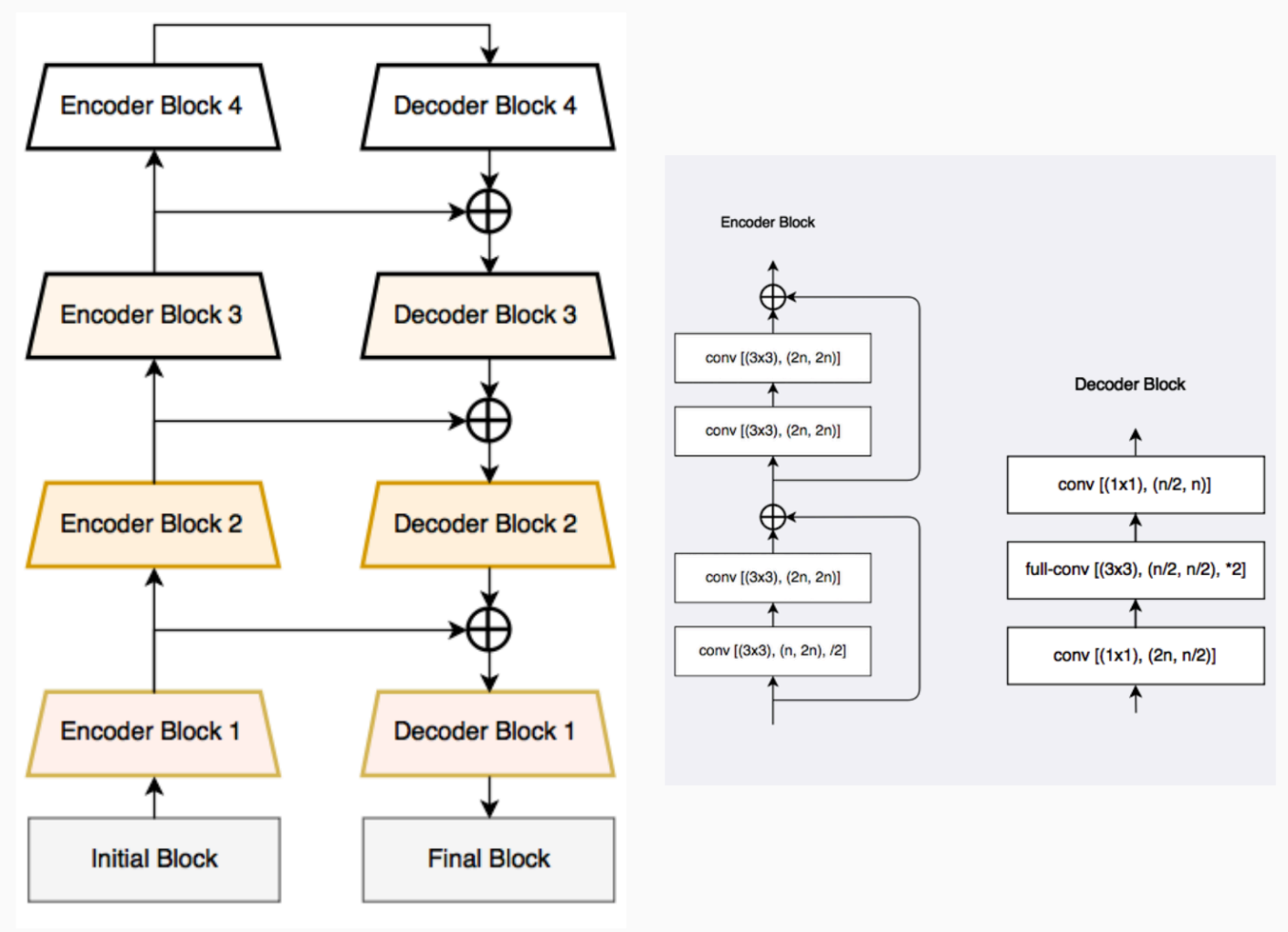
Fully Convolutional DenseNet

В качестве основы для декодера и энкодера используется dense-блок, как в архитектуре DenseNet

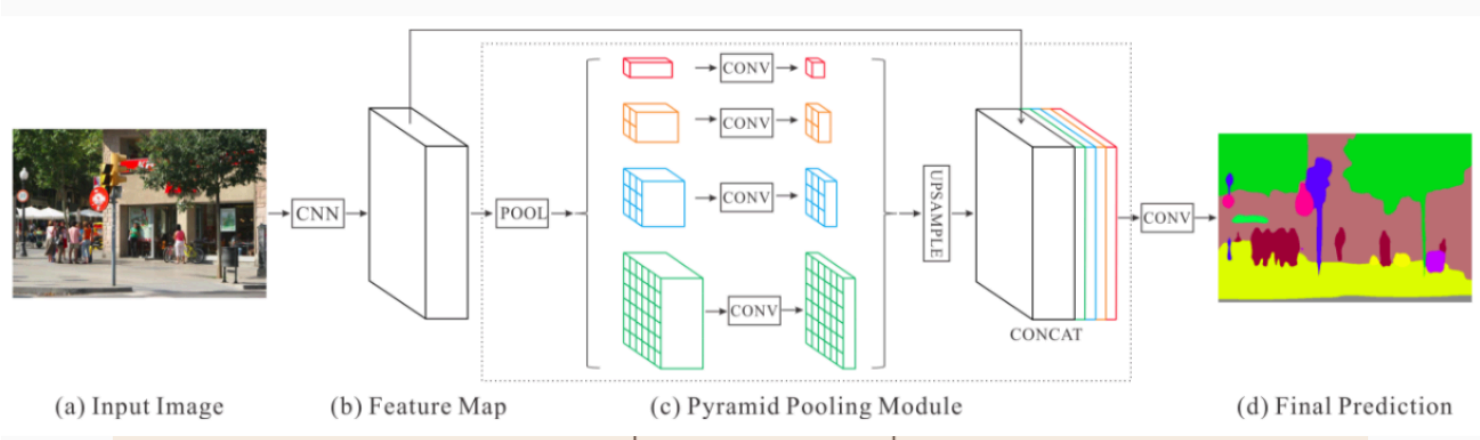


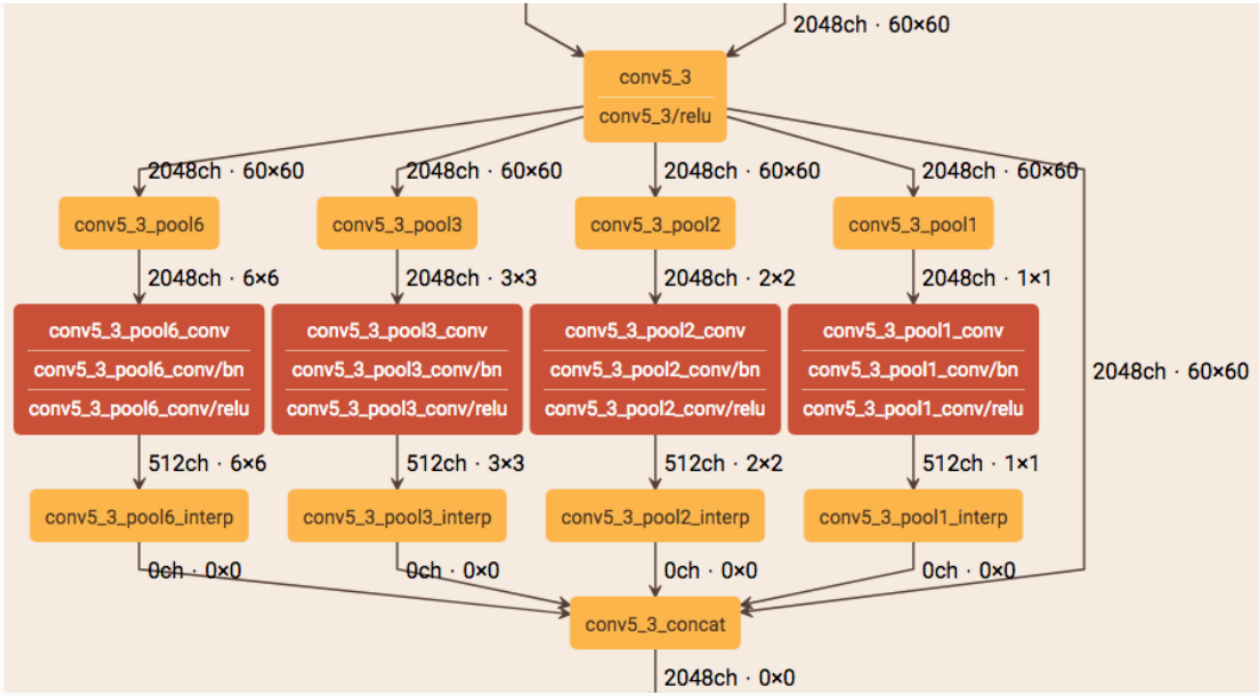


LinkNet (и ENet)



PSP-Net

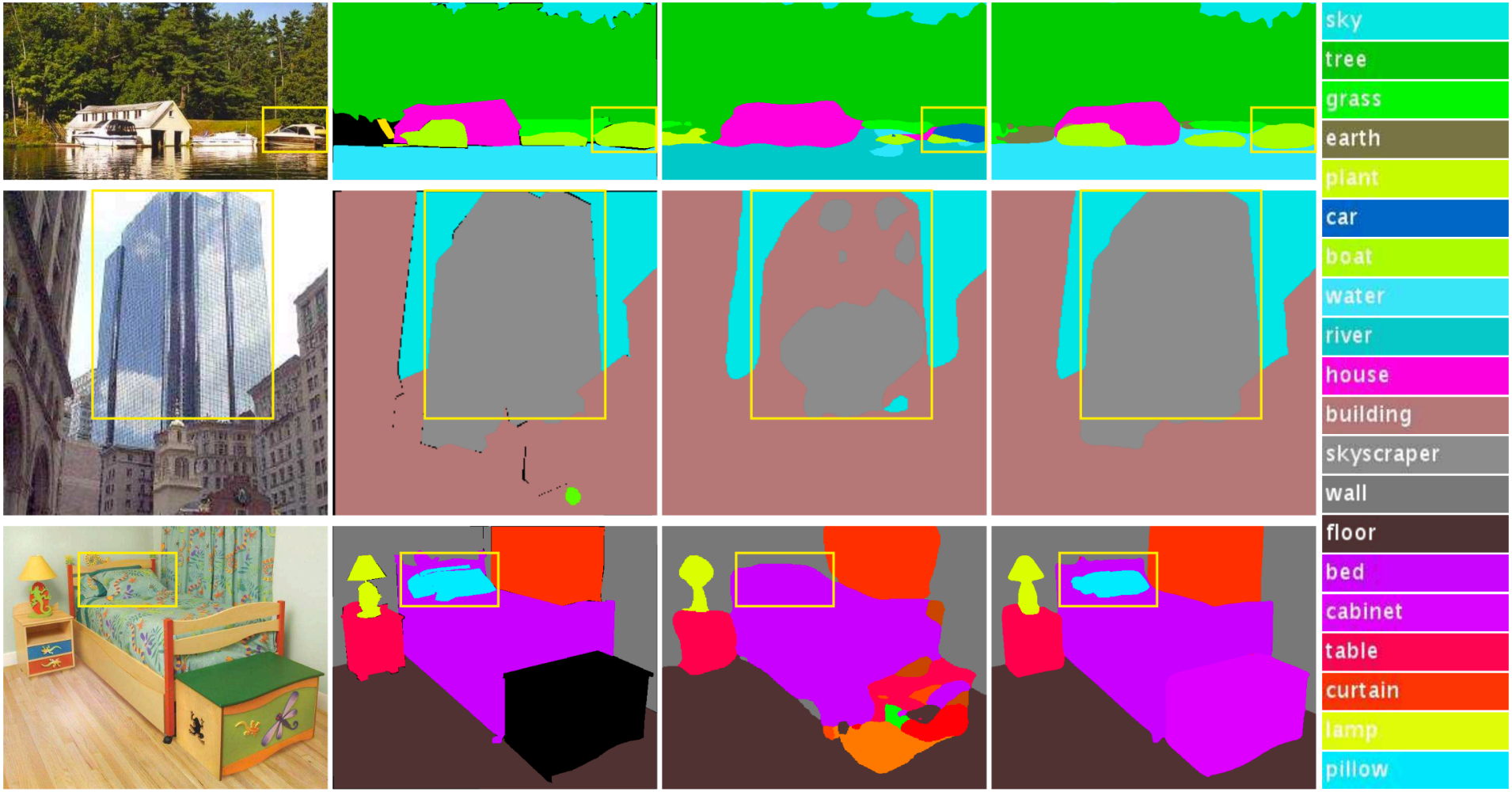




Top : The PSPNet Architecture

Bottom : The Spatial Pyramid Pooling in visualized in detail using netscope

PSP-net. Результаты

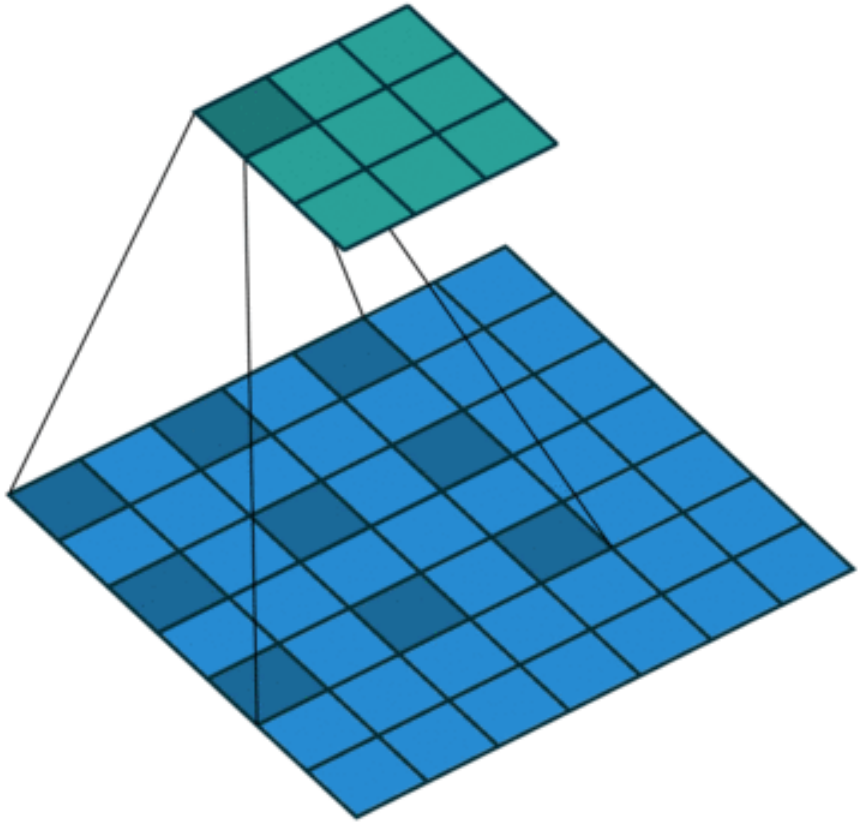


(a) Image (b) Ground Truth (c) FCN (d) PSPNet (e) ColorMap

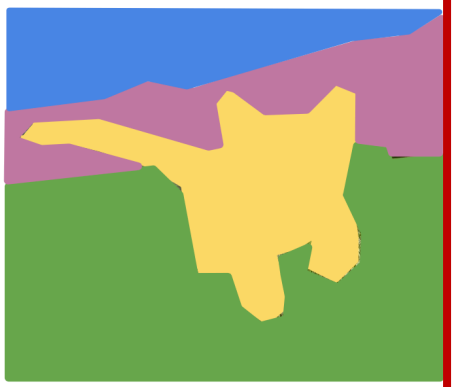
Функции потерь

- crossentropy
- crossentropy – a * soft dice
- crossentropy – a * log(soft dice)
- фантазии на тему (y_true – y_pred)

Dilated (Atrous) convolutions



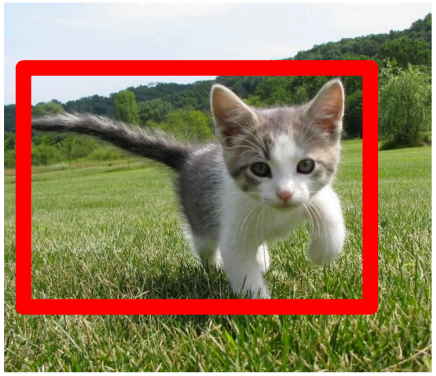
Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

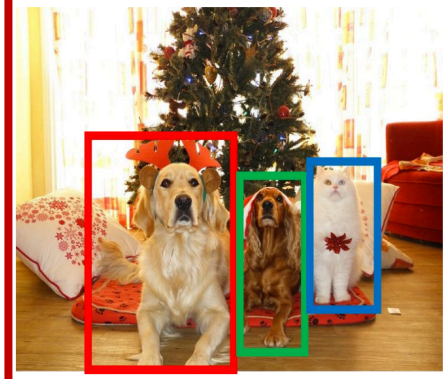
Classification + Localization



CAT

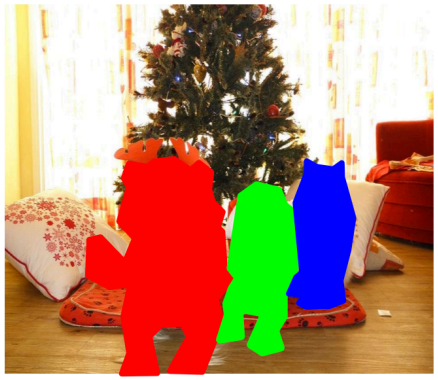
Single Object

Object Detection



DOG, DOG, CAT

Instance Segmentation

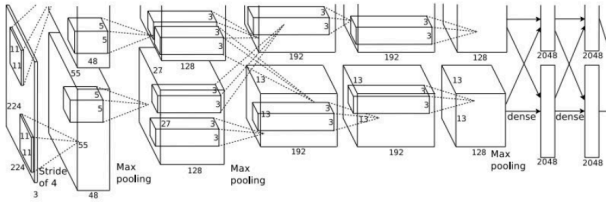


DOG, DOG, CAT

Multiple Object

Classification + Localization

This image is CC0 public domain



Fully Connected: 4096 to 1000

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

Softmax Loss

Correct label: Cat

Vector: 4096

Fully Connected: 4096 to 4

Box Coordinates (x, y, w, h)

L2 Loss

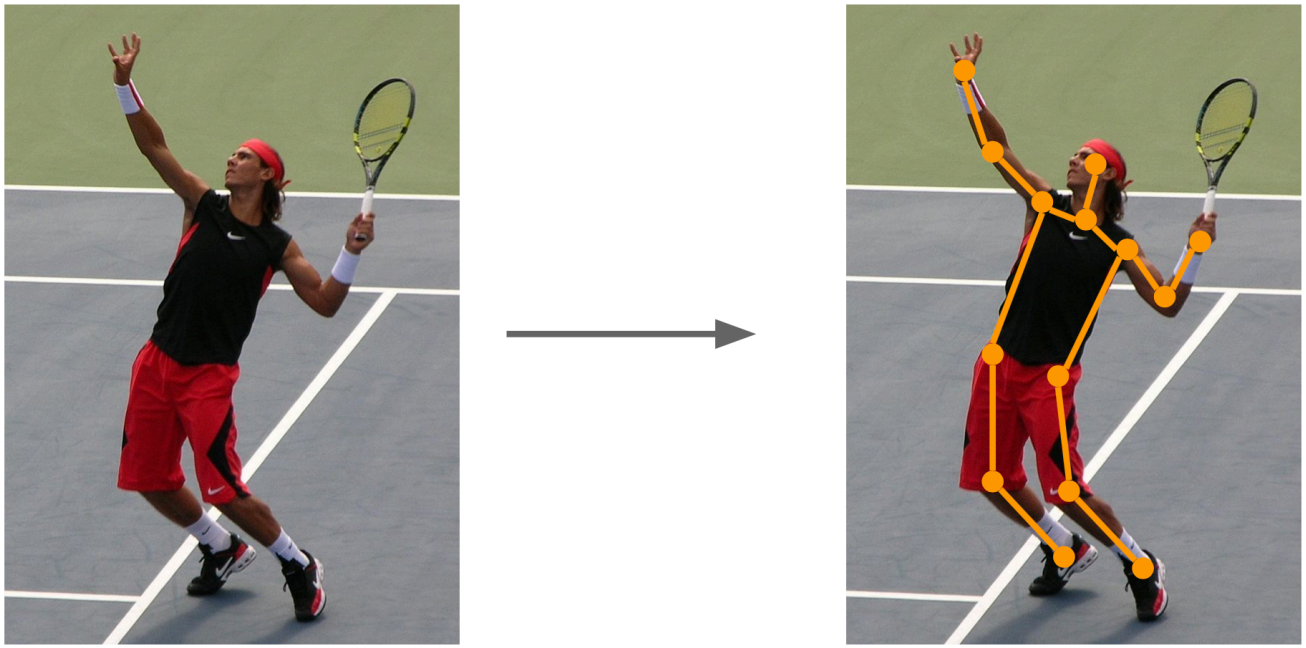
Correct box: (x', y', w', h')

Multitask Loss

+ → **Loss**

Treat localization as a regression problem!

Human pose estimation

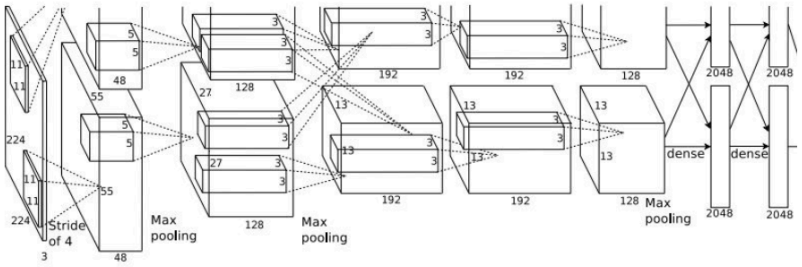


Represent pose as a set of 14 joint positions:

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

Human pose estimation



Left foot: (x, y)

Right foot: (x, y)

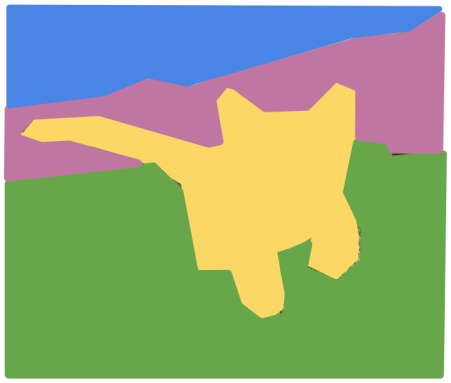
…

Head top: (x, y)

Vector:
4096

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

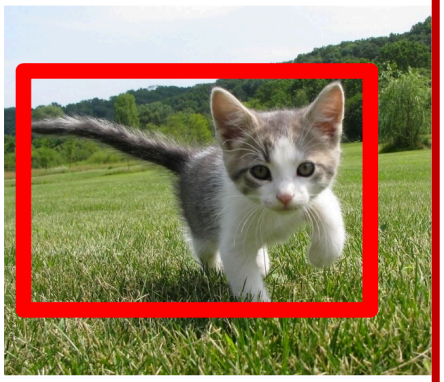
Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

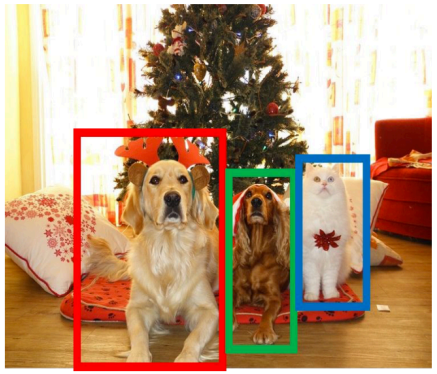
Classification + Localization



CAT

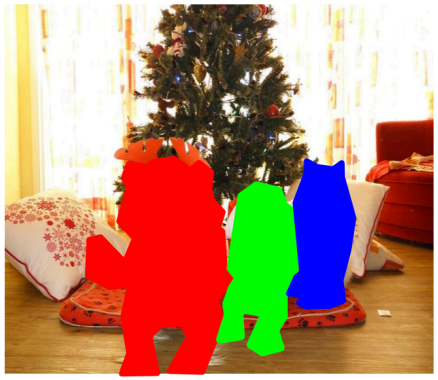
Single Object

Object Detection



DOG, DOG, CAT

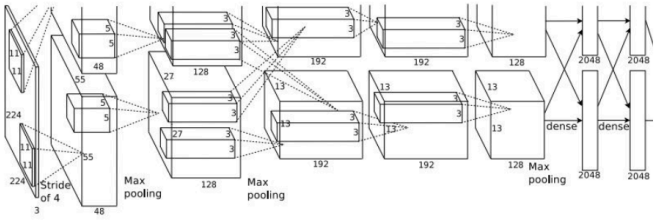
Instance Segmentation



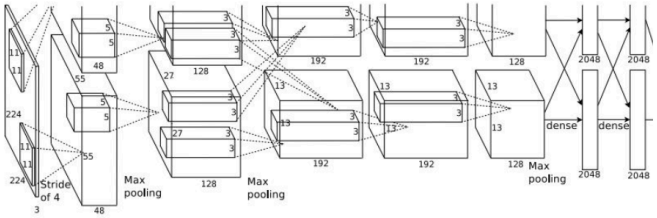
DOG, DOG, CAT

Multiple Object

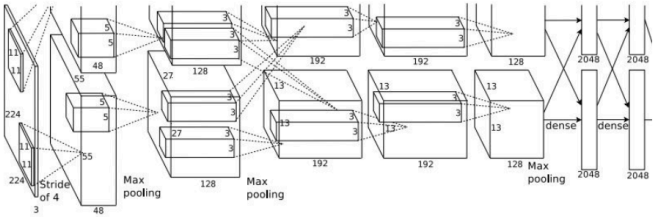
Почему не подойдет подход с регрессией?



CAT: (x, y, w, h)



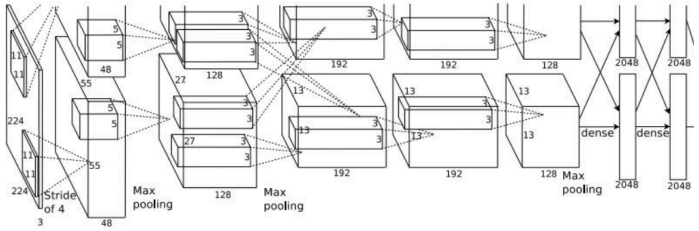
DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)



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

Object Detection as Classification: Sliding Window

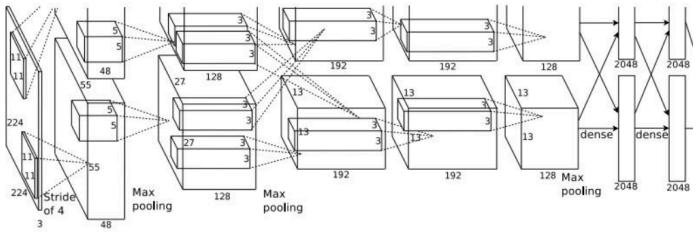
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? NO
Background? YES

Object Detection as Classification: Sliding Window

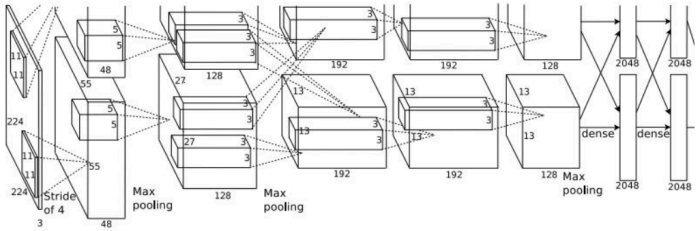
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

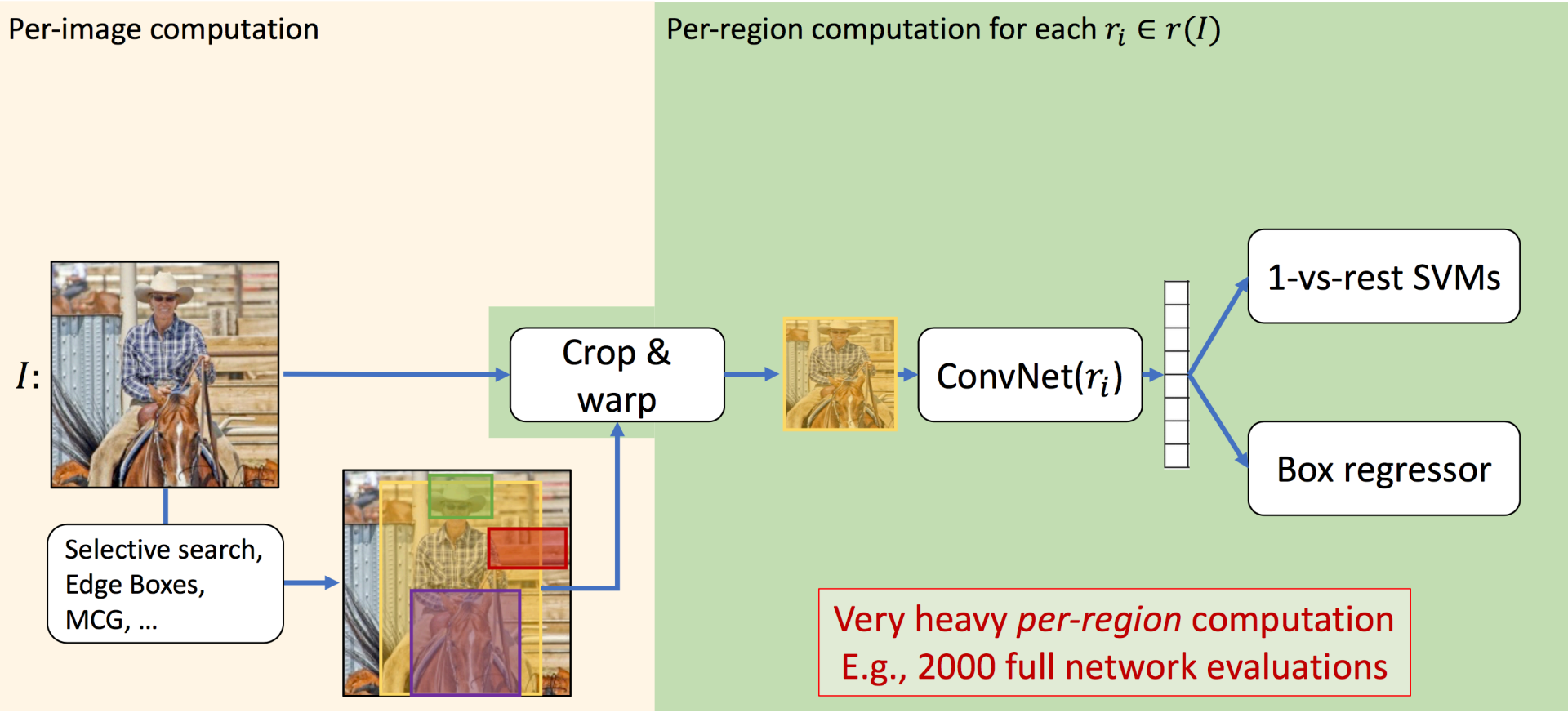
Object Detection as Classification: Sliding Window

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

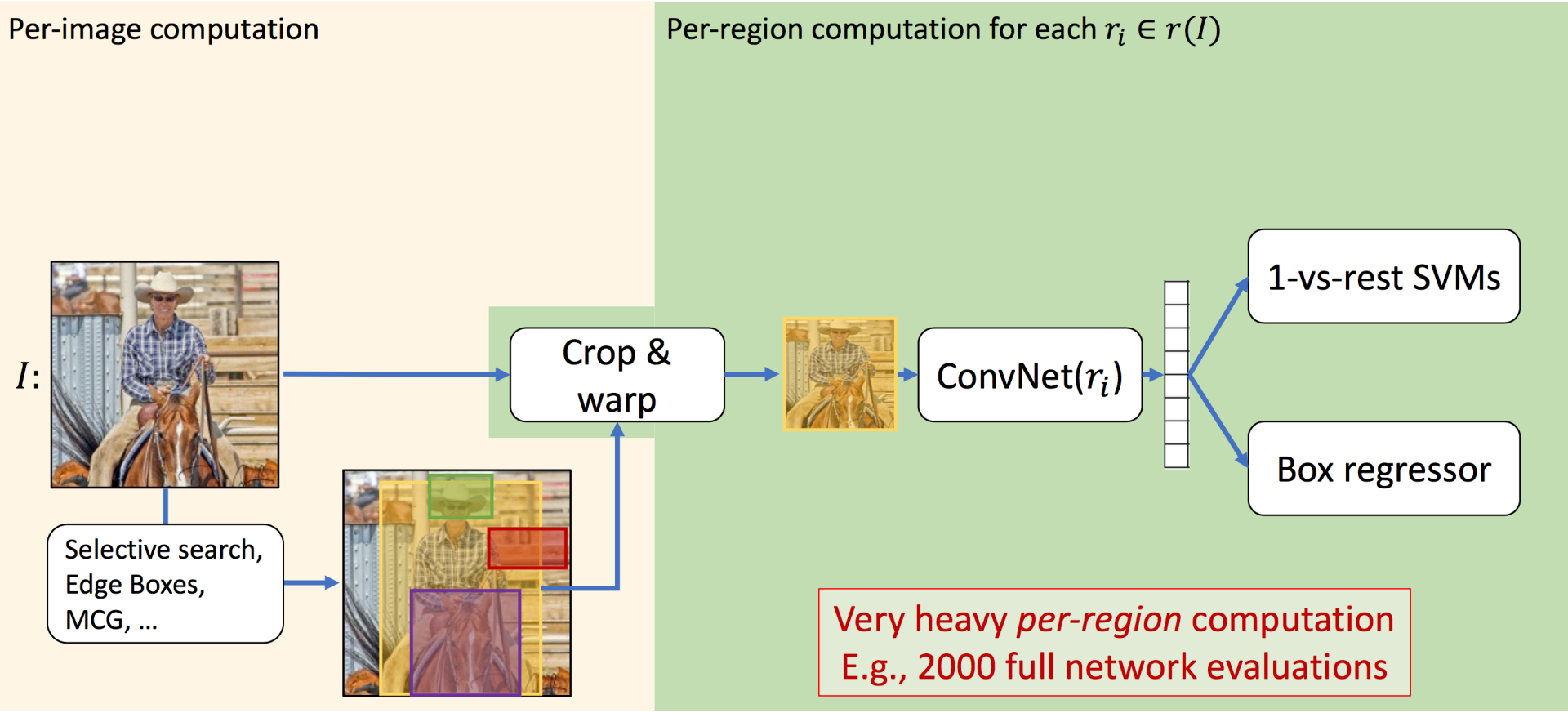


Dog? YES
Cat? NO
Background? NO

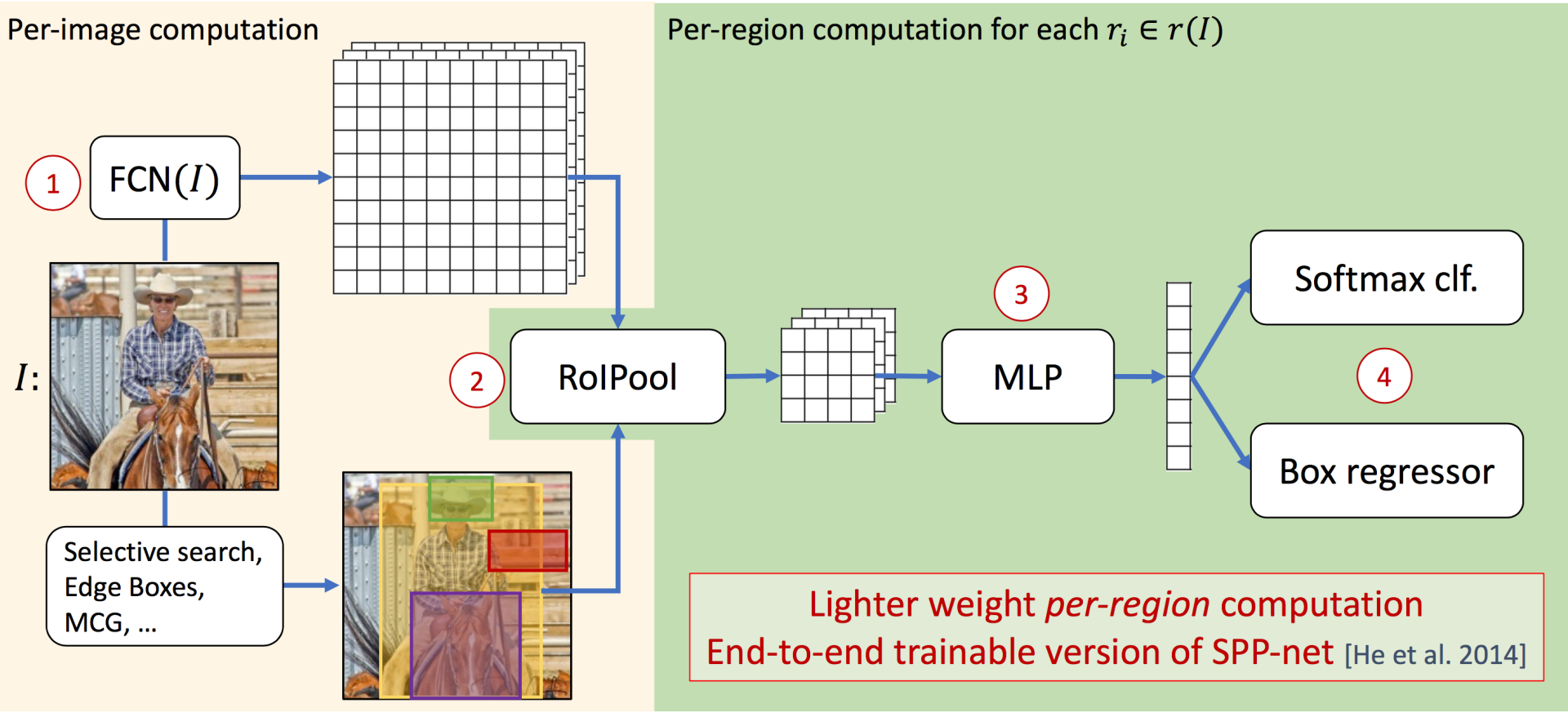
"Slow" R-CNN


Per-image computation
Per-region computation for each $r_i \in r(I)$
I:
Selective search, Edge Boxes, MCG, …
Crop & warp
ConvNet(r_i)
1-vs-rest SVMs
Box regressor
Very heavy *per-region* computation
E.g., 2000 full network evaluations

"Slow" R-CNN

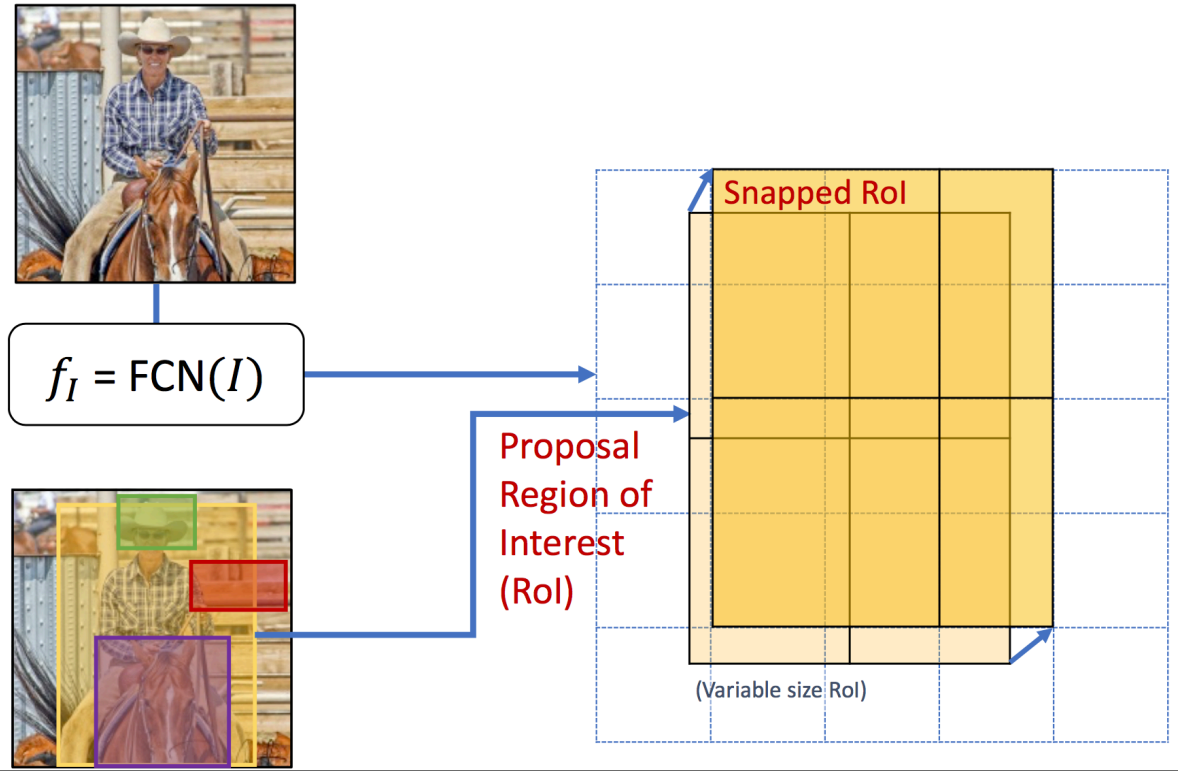

Per-image computation
Per-region computation for each $r_i \in r(I)$
I:
Selective search, Edge Boxes, MCG, …
Crop & warp
ConvNet(r_i)
1-vs-rest SVMs
Box regressor
Very heavy *per-region* computation
E.g., 2000 full network evaluations

Generalized R-CNN → Fast R-CNN



He, Zhang, Ren, Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. ECCV 2014.

RoIPool (on each Proposal)



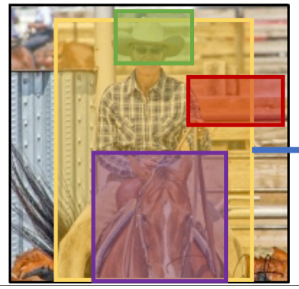
Key innovation in SPP-net
[He et al. 2014]

He, Zhang, Ren, Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. ECCV 2014.

RoIPool (on each Proposal)

Transform arbitrary size proposal into a fixed-dimensional representation (e.g., 2x2)

f_I = FCN(I)



Proposal Region of Interest (RoI)

Snapped RoI

(Variable size RoI)

RoIPool transform

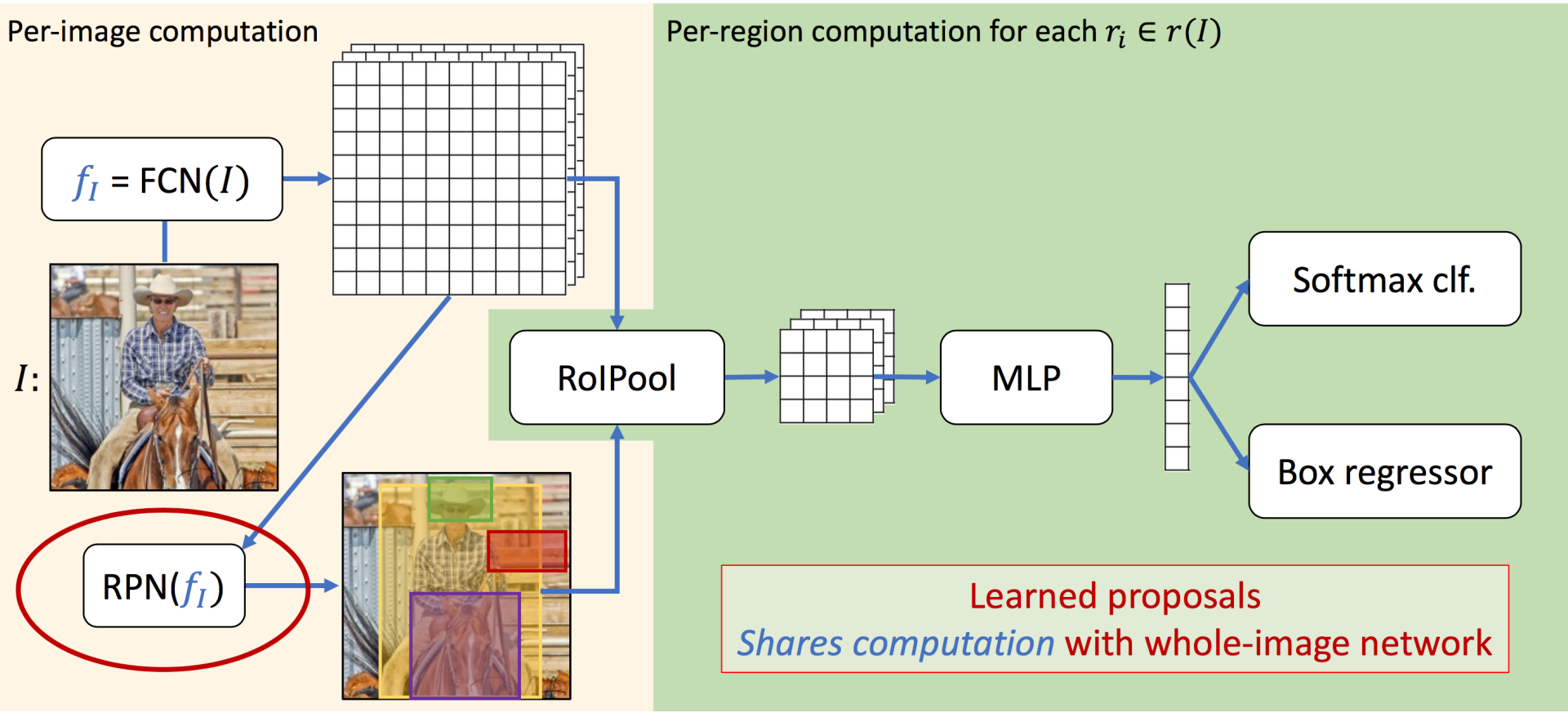
(Fixed dimensional representation)

MLP

Feature value is max over input cells

Ren, He, Girshick, Sun. Faster R-CNN:
Towards Real-Time Object Detection. NIPS 2015.

Faster R-CNN



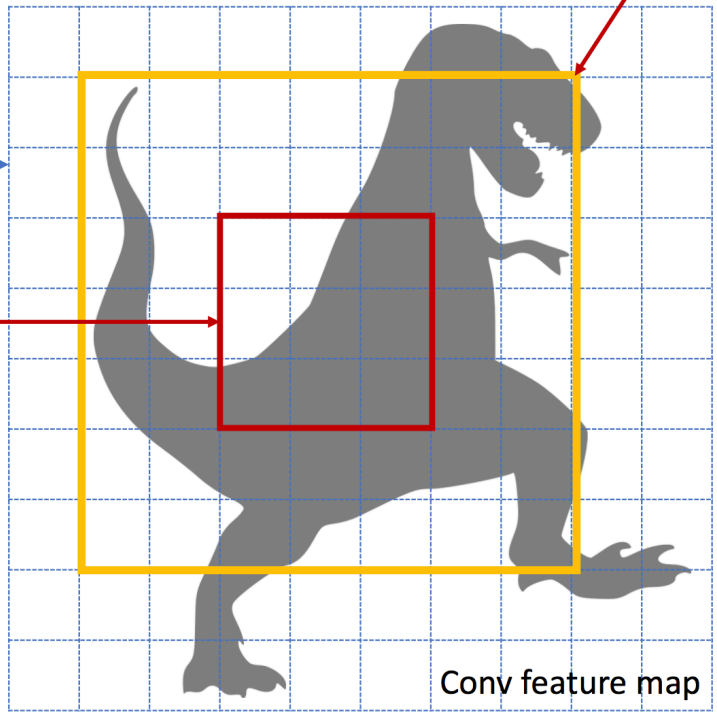
RPN: Anchor Box

f_I = FCN(I)

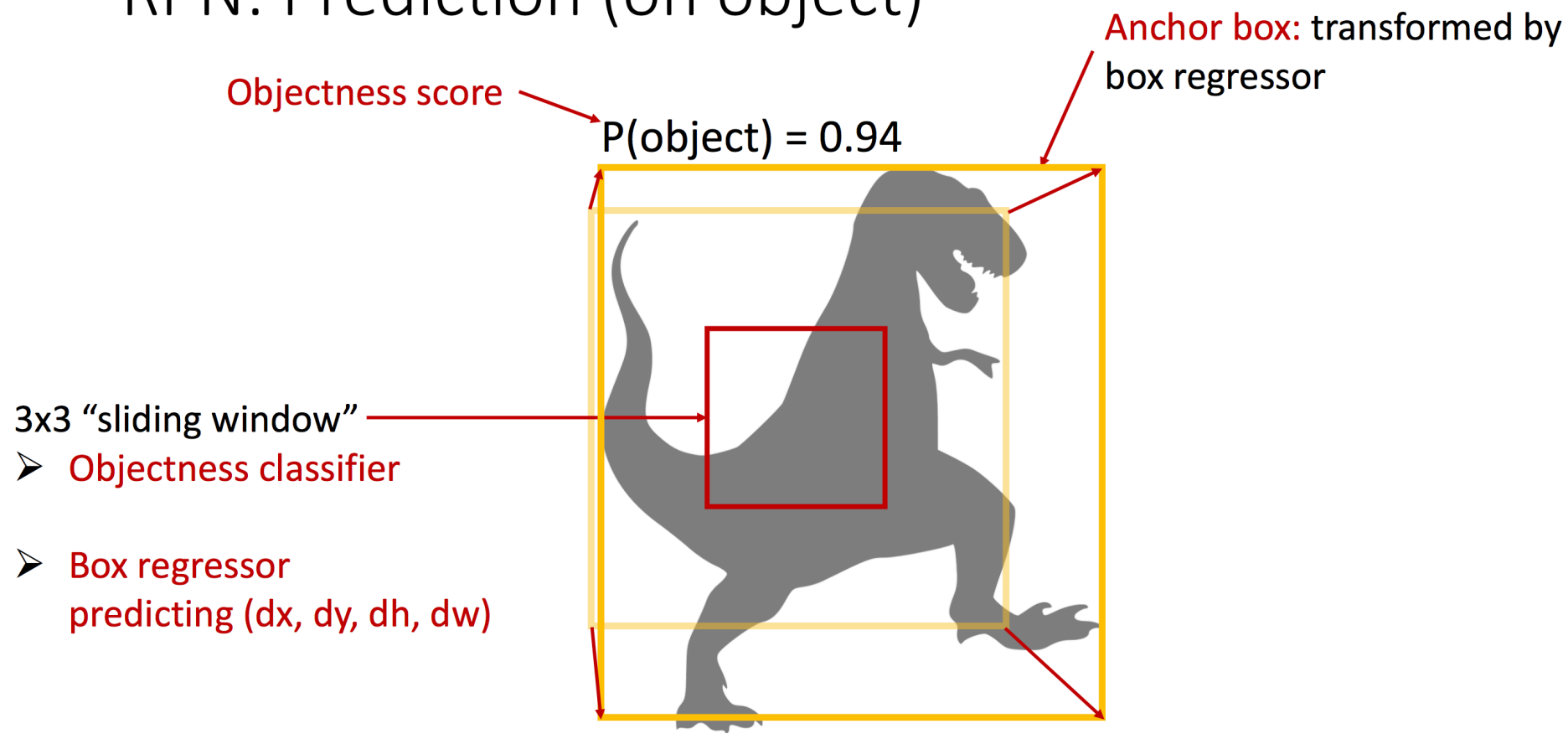
Anchor box: predictions are w.r.t. this box, *not the 3x3 sliding window*

3x3 “sliding window”

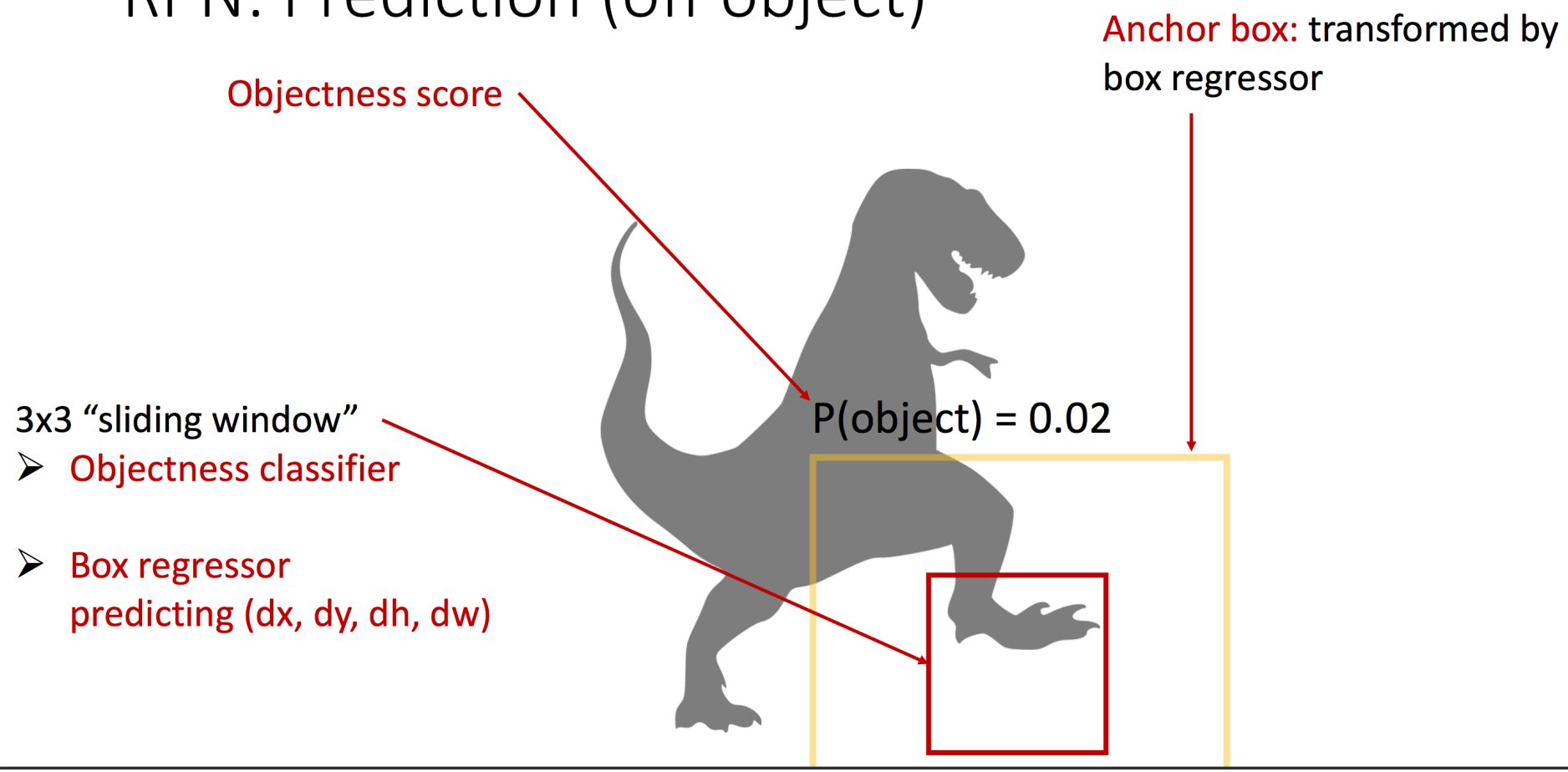
- Objectness classifier
- Box regressor predicting (dx, dy, dh, dw)



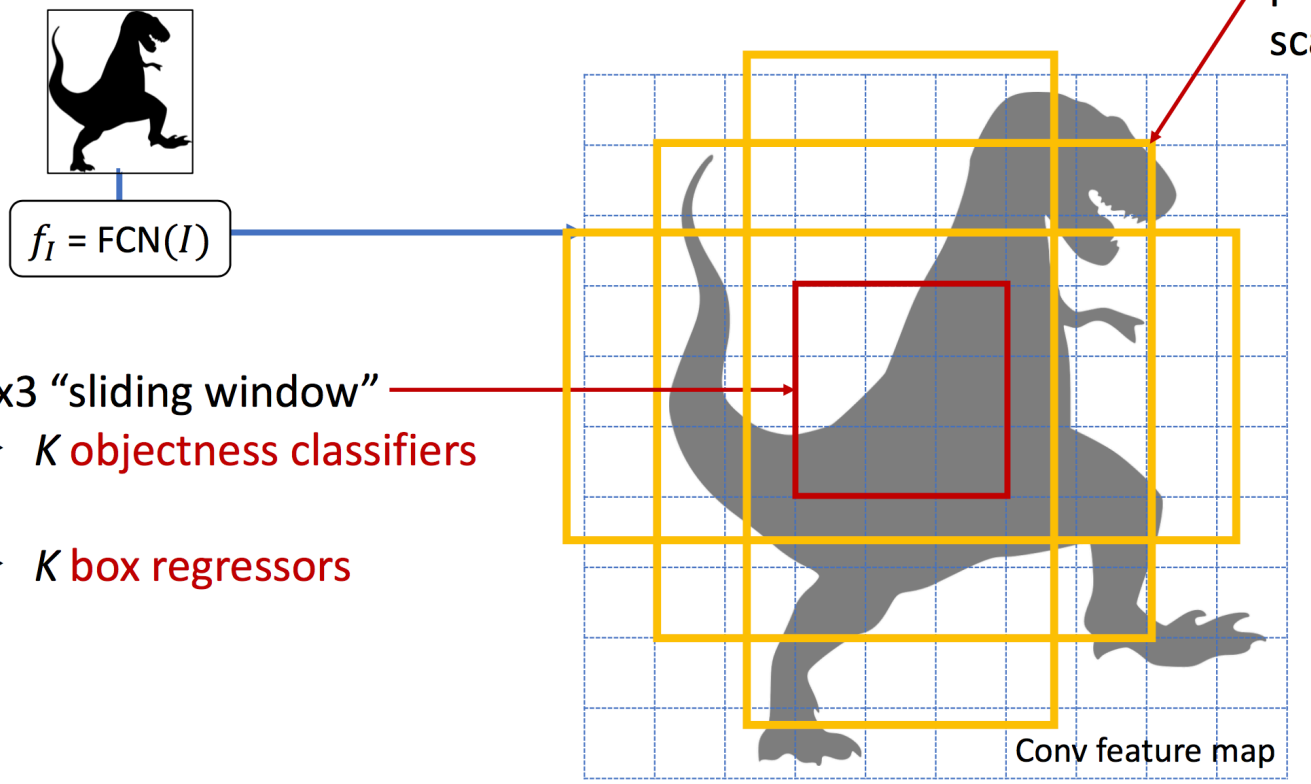
RPN: Prediction (on object)



RPN: Prediction (off object)

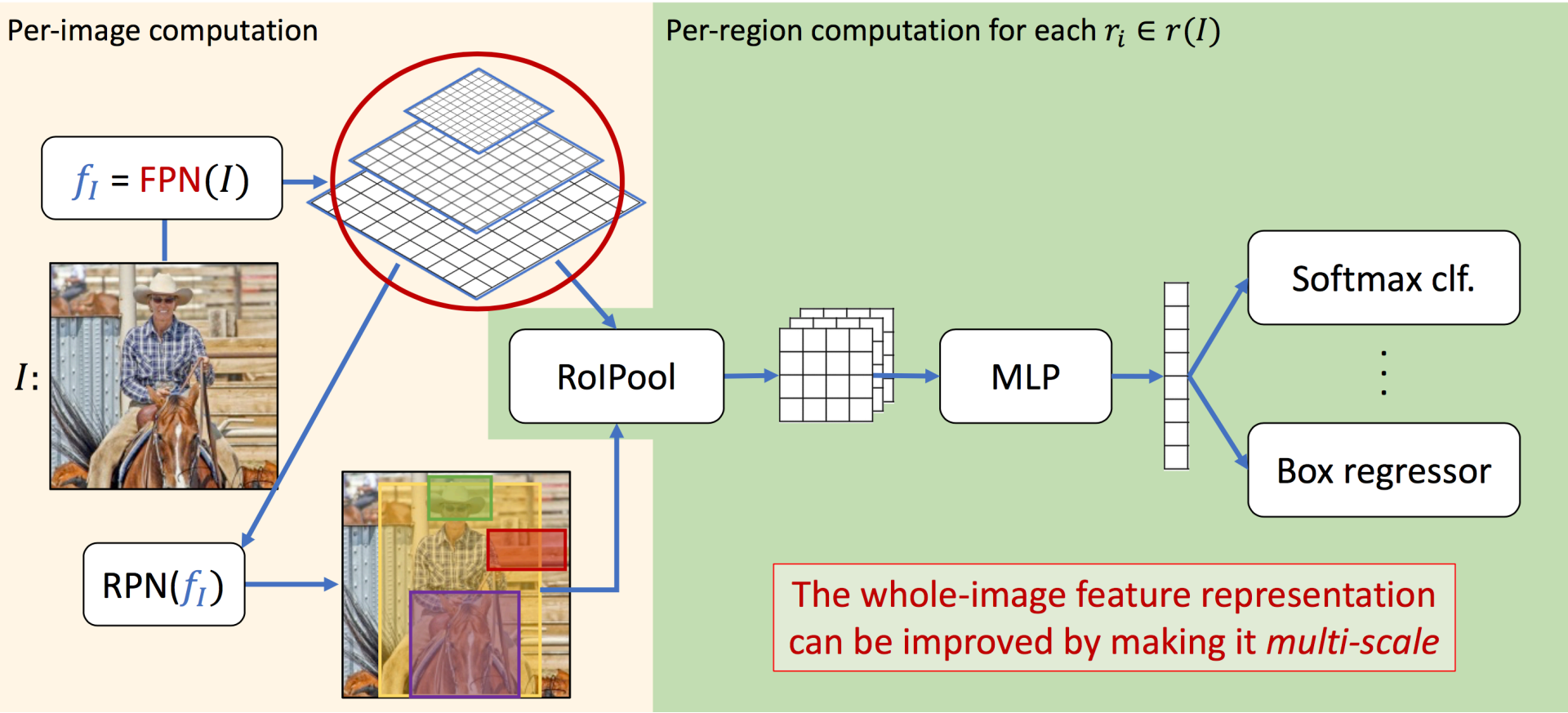


RPN: Multiple Anchors



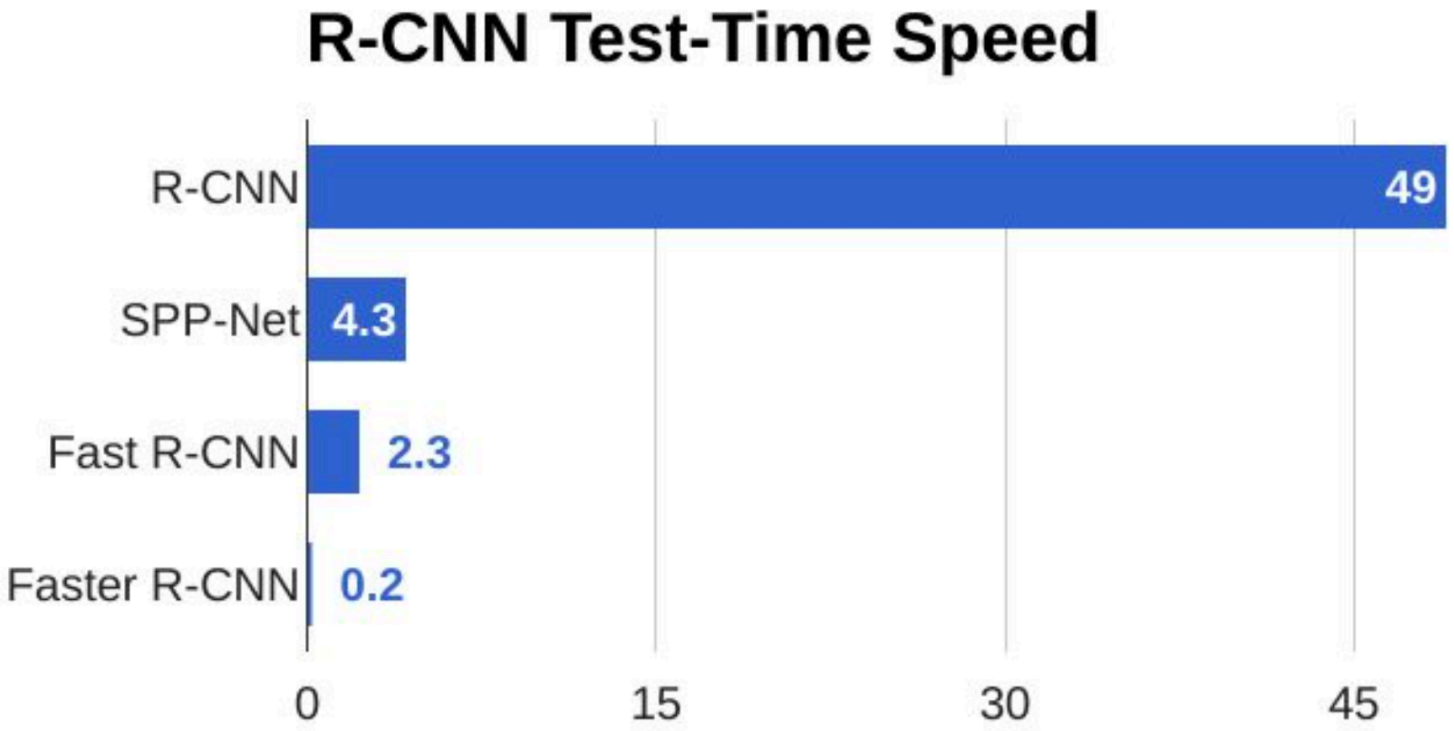
Lin et al. Feature Pyramid Networks for Object Detection. CVPR 2017.

Faster R-CNN with a Feature Pyramid Network



Faster R-CNN:

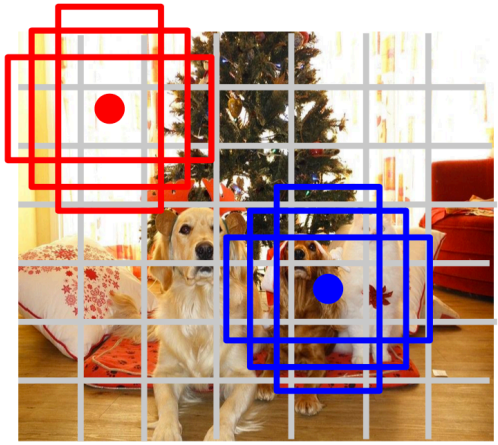
Make CNN do proposals!



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

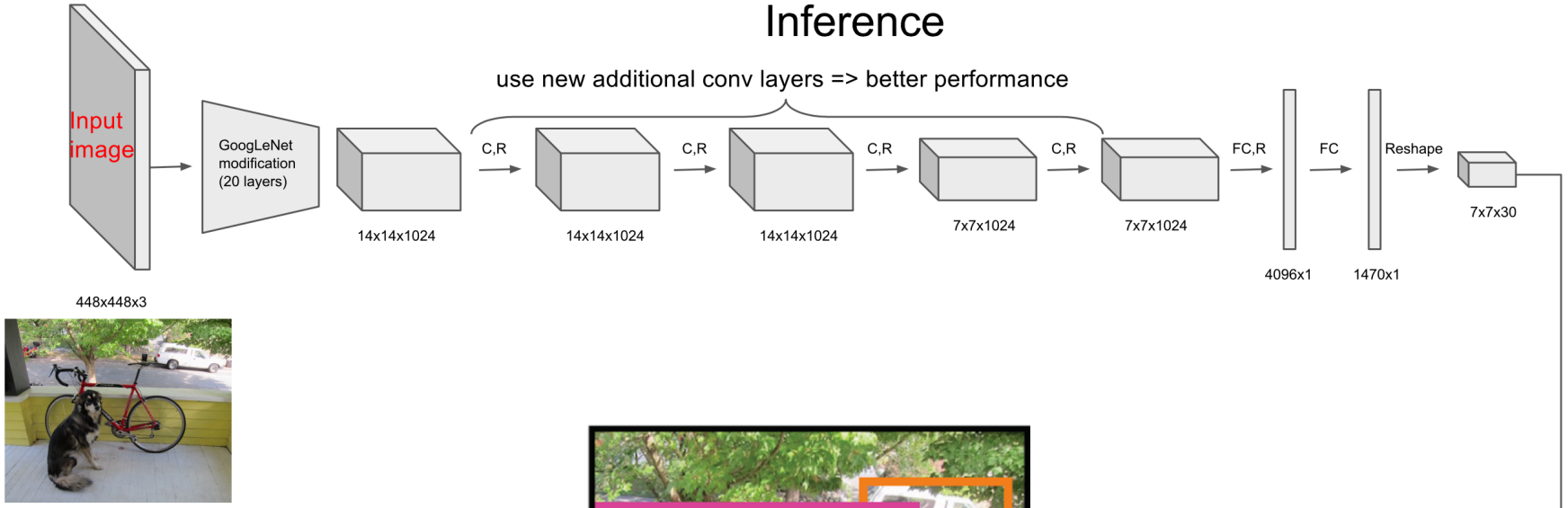
Within each grid cell:

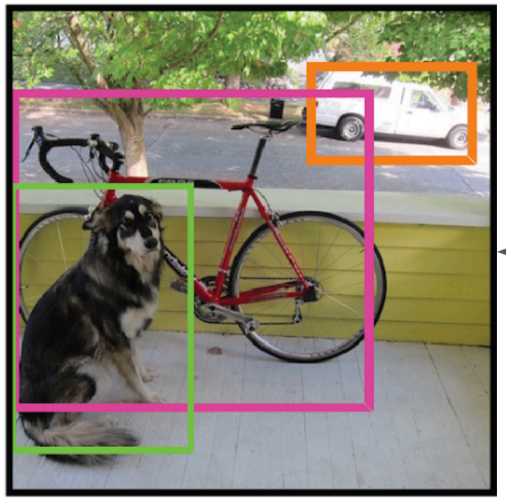
- Regress from each of the B base boxes to a final box with 5 numbers: (dx, dy, dh, dw, confidence)
- Predict scores for each of C classes (including background as a class)

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

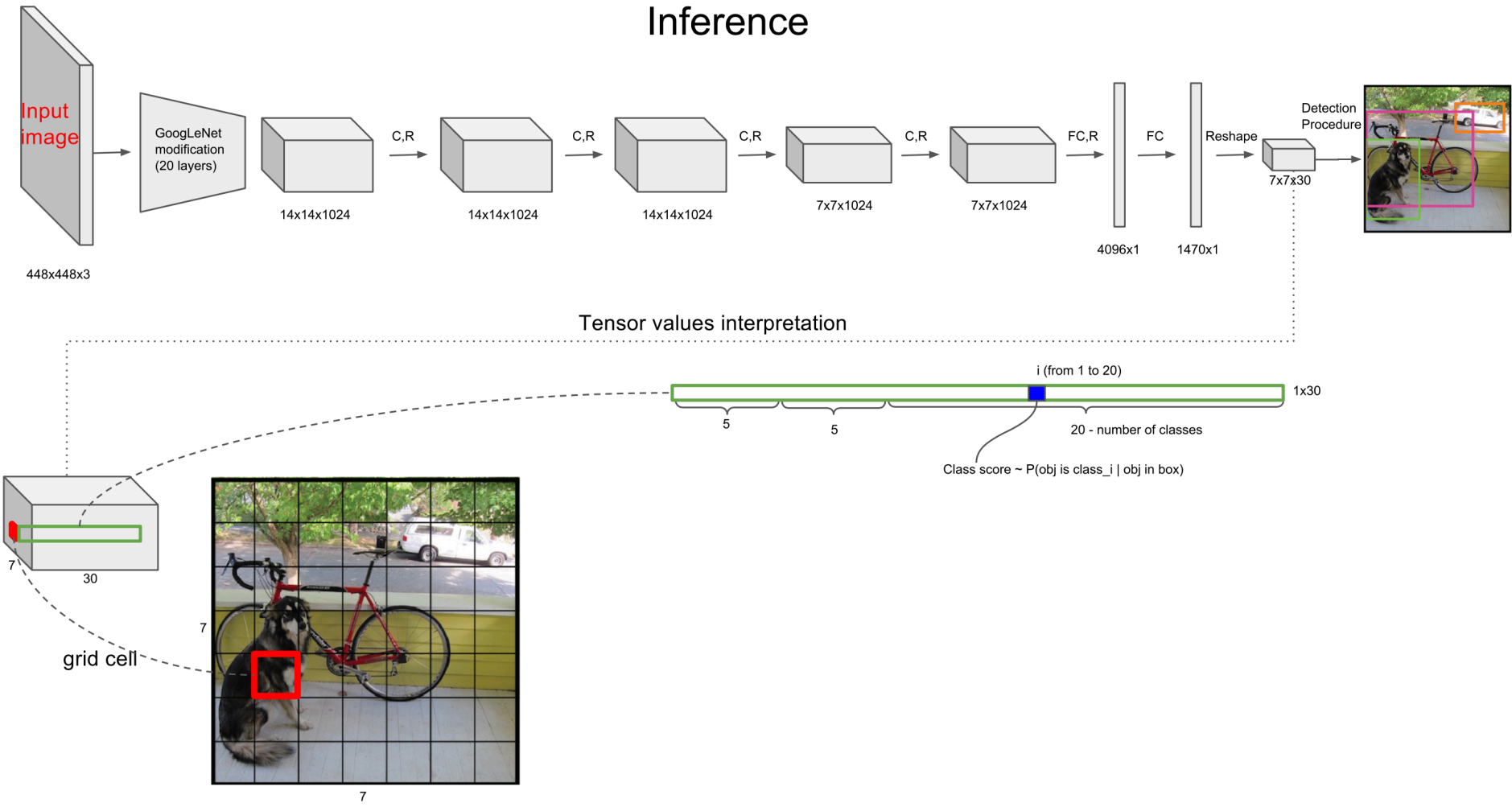
Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016

Yolo

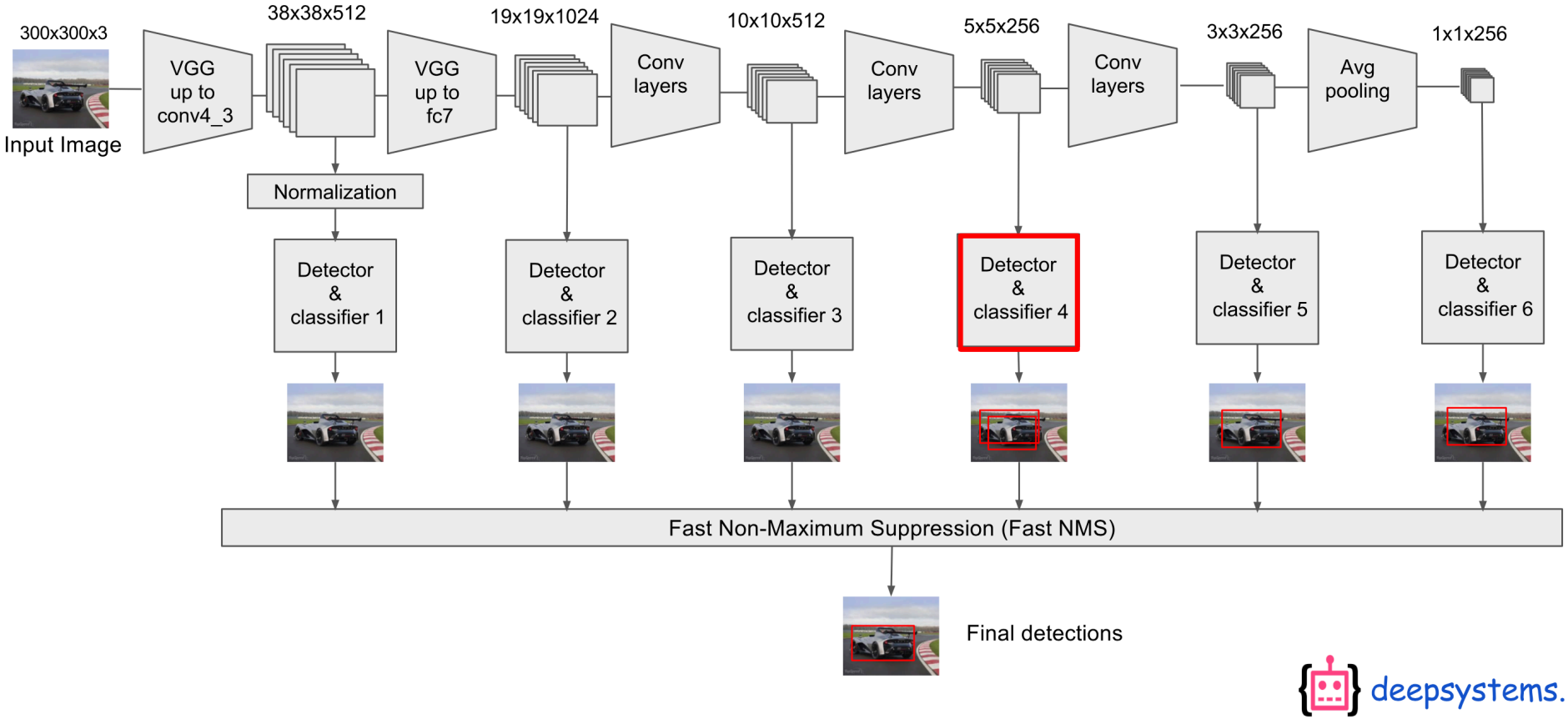

Inference
use new additional conv layers => better performance
Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Detection Procedure



Yolo

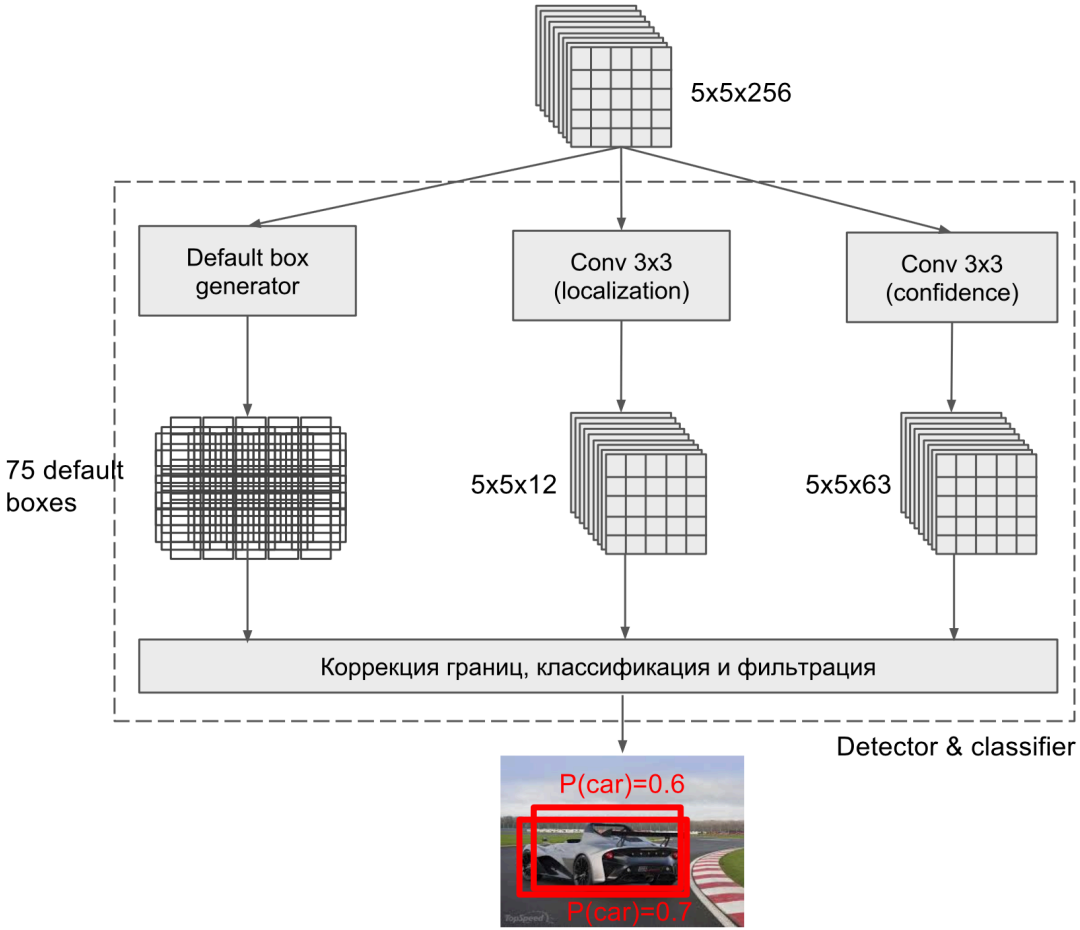

Inference
Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Detection Procedure
Tensor values interpretation
i (from 1 to 20)
1x30
5
5
20 - number of classes
Class score ~ P(obj is class_i | obj in box)
7
7
30
grid cell
7
7

Архитектура SSD 300



deepsystems.io

Архитектура SSD 300. Detector & classifier

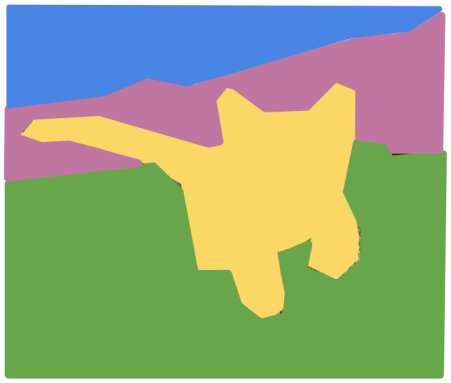


Пусть заданы следующие параметры:

- Размер исходного изображения (300 x 300)
- Размерность feature maps (5 x 5 x 256)
- #default boxes = 3

deepsystems.io

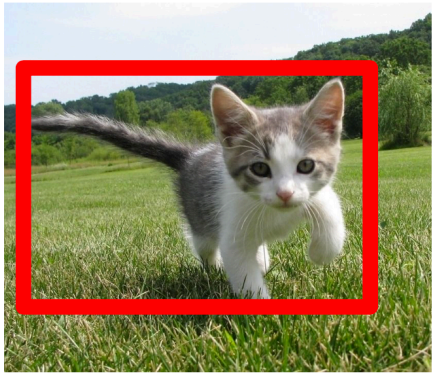
Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

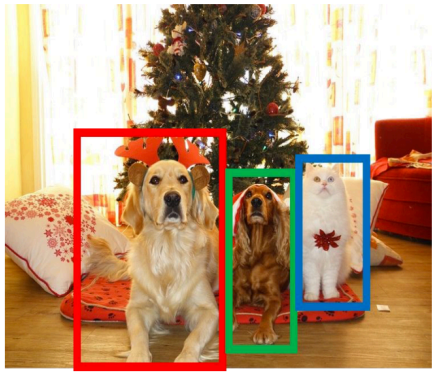
Classification + Localization



CAT

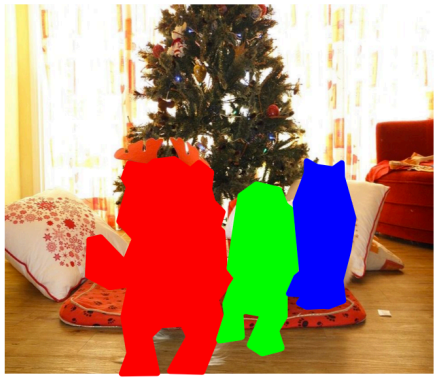
Single Object

Object Detection



DOG, DOG, CAT

Instance Segmentation

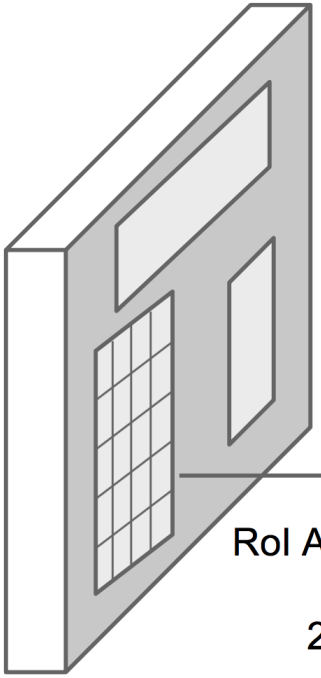


DOG, DOG, CAT

Multiple Object

Mask R-CNN

CNN



Classification Scores: C
Box coordinates (per class): 4 * C

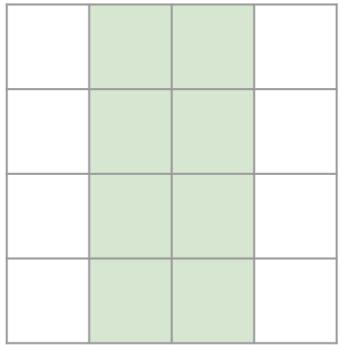
RoI Align

256 x 14 x 14

Conv

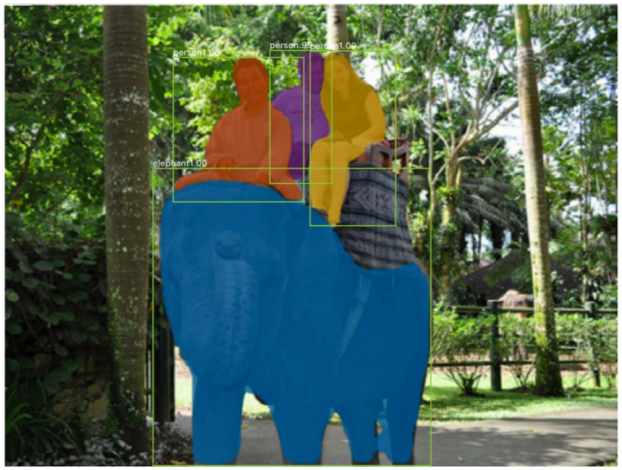
256 x 14 x 14

Conv



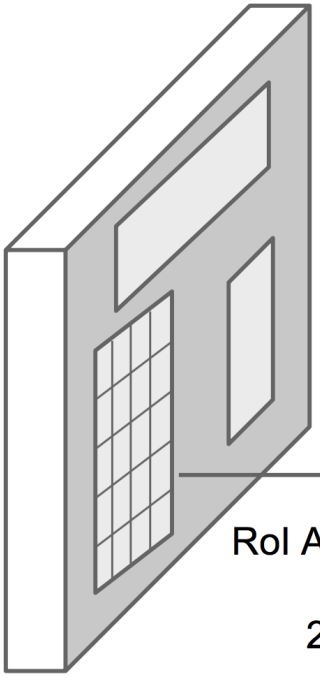
Predict a mask for each of C classes

C x 14 x 14



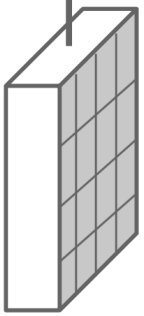
Mask R-CNN
Also does pose

CNN



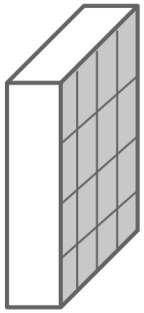
RoI Align

Classification Scores: C
Box coordinates (per class): 4 * C
Joint coordinates



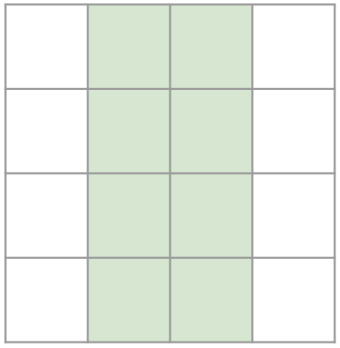
256 x 14 x 14

Conv

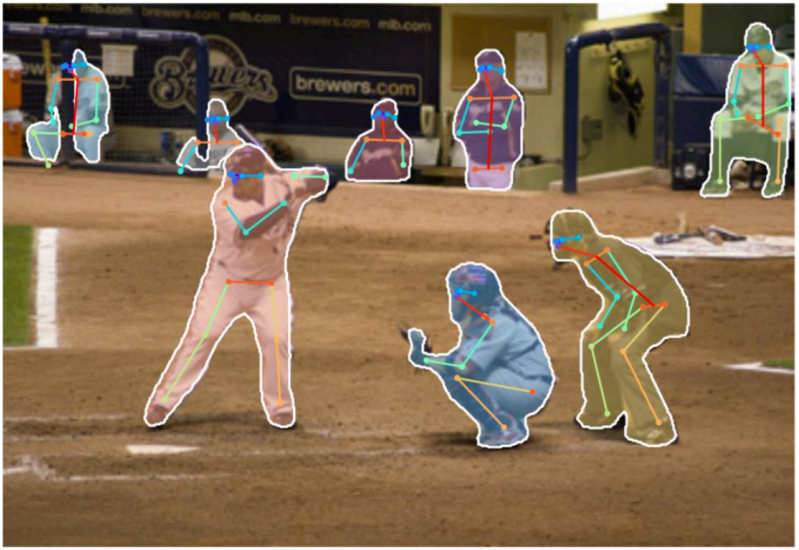


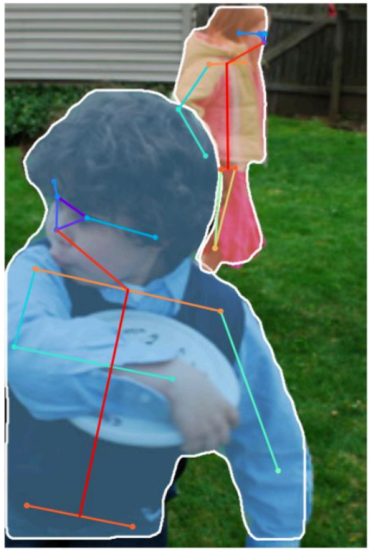
256 x 14 x 14

Conv



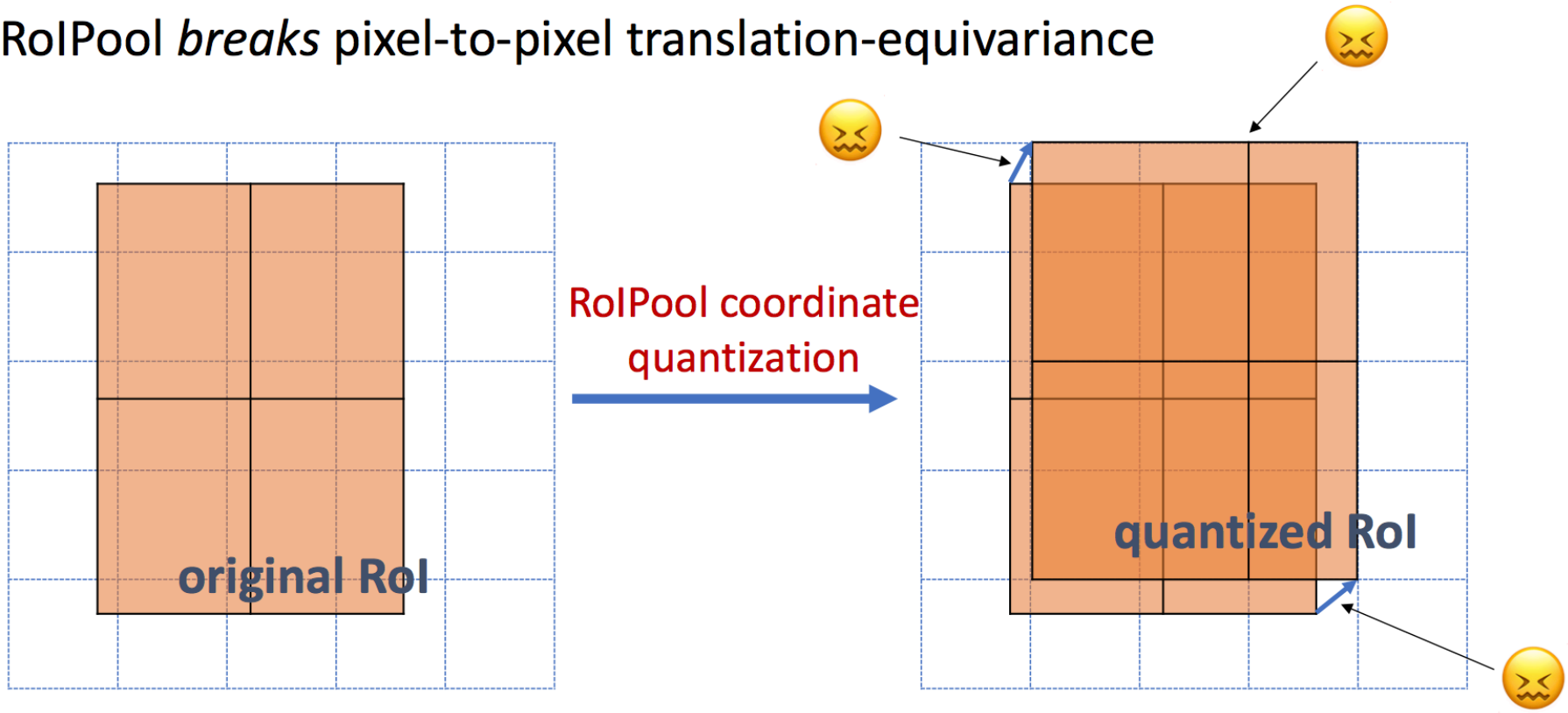
Predict a mask for each of C classes





RoIAlign vs. RoIPool

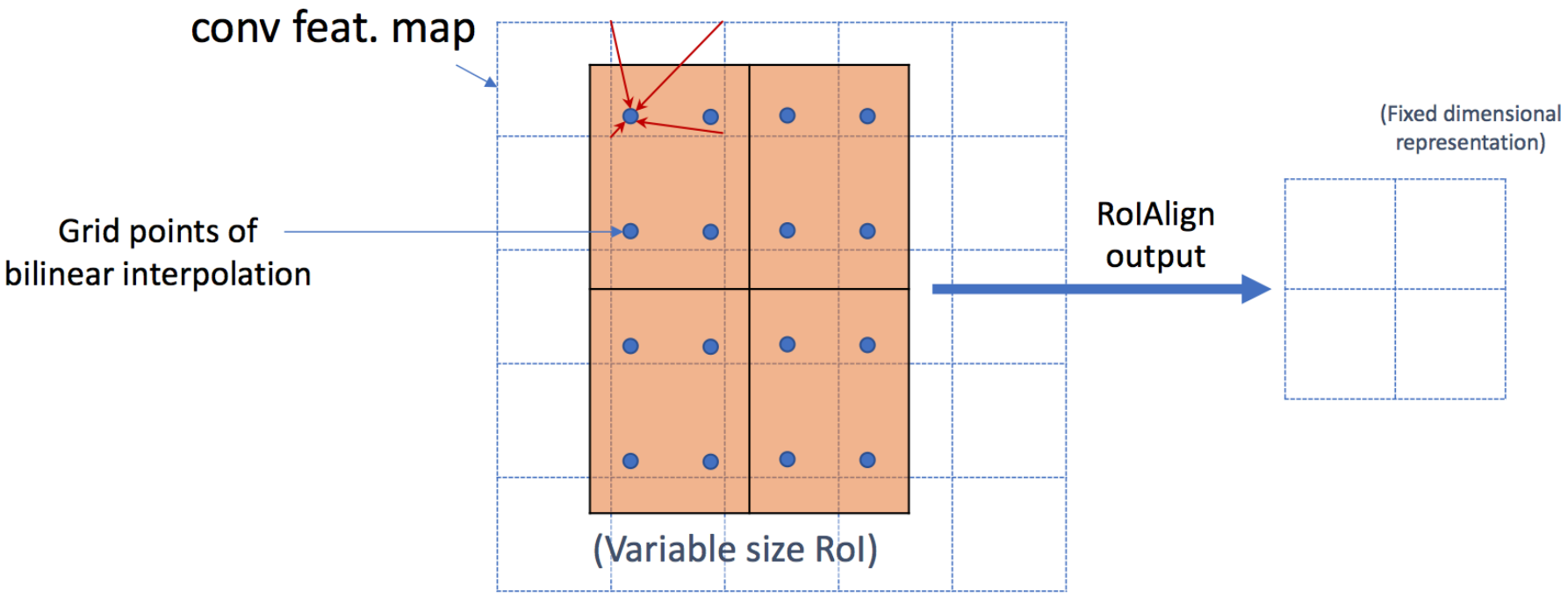
- RoIPool *breaks* pixel-to-pixel translation-equivariance



RoIAlign

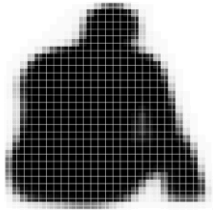
FAQs: how to sample grid points within a cell?

- 4 regular points in 2x2 sub-cells
- other implementation could work



Validation image with box detection shown in red

28x28 soft prediction from Mask R-CNN
(enlarged)

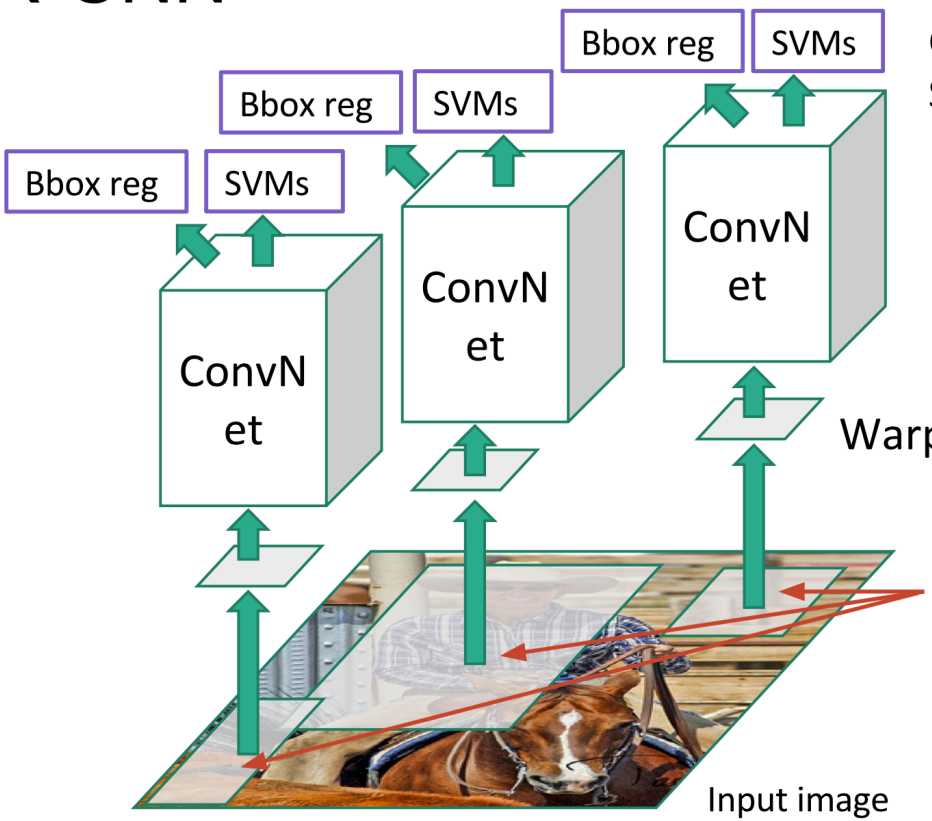


Soft prediction resampled to image coordinates
(bilinear and bicubic interpolation work equally well)

Final prediction (threshold at 0.5)

Спасибо за внимание

R-CNN



Linear Regression for bounding box offsets

Classify regions with SVMs

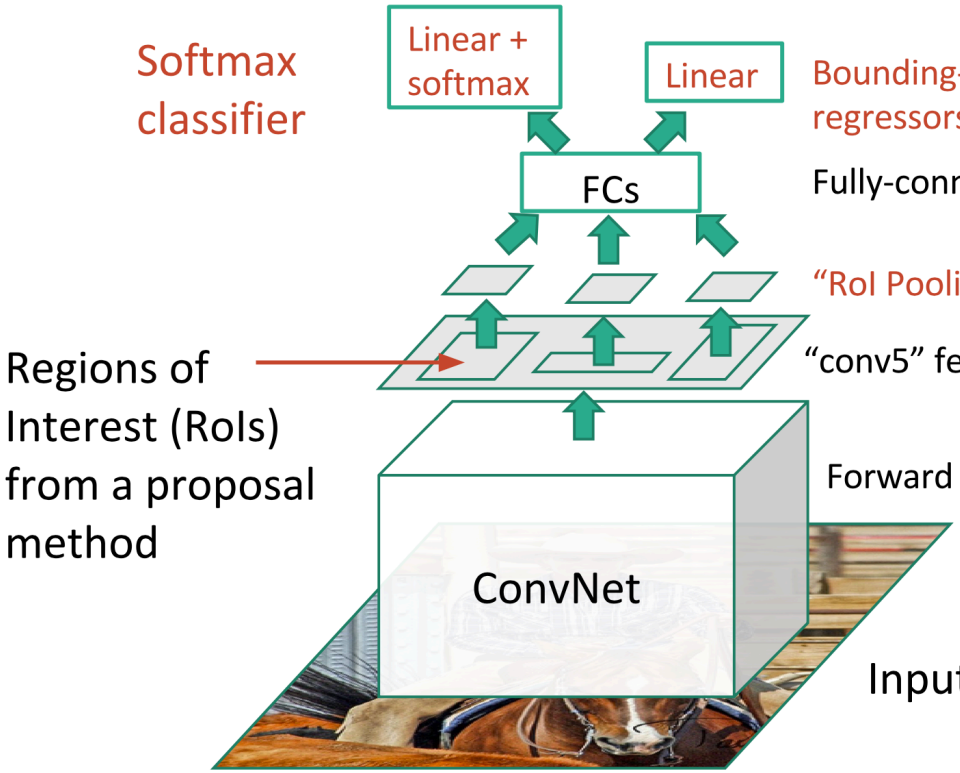
Forward each region through ConvNet

Warped image regions

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

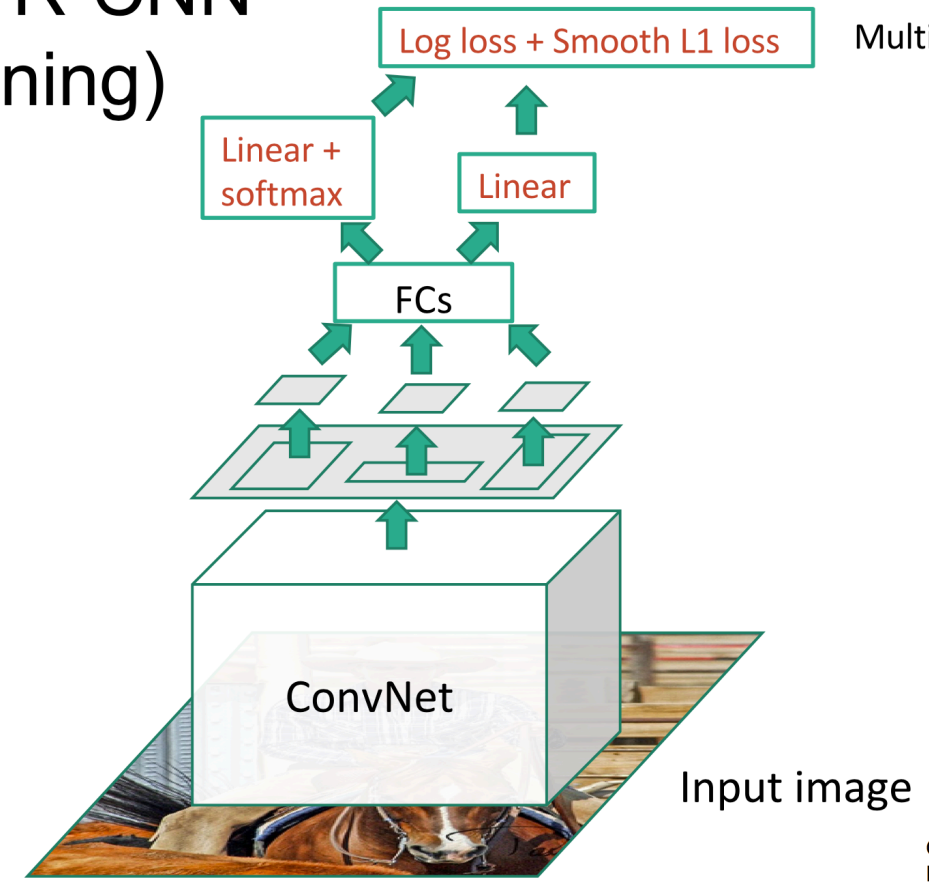
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

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.

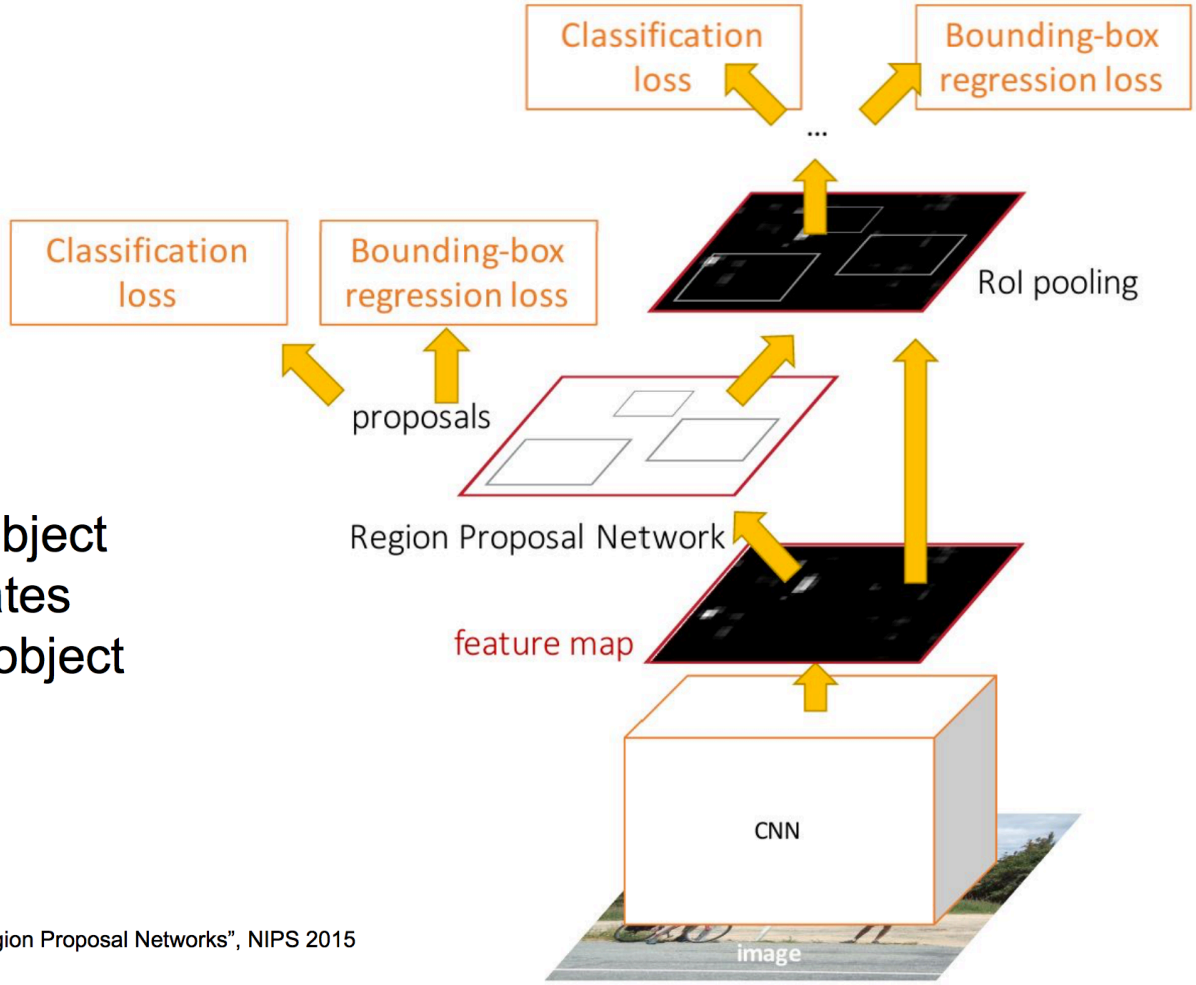
Faster R-CNN:

Make CNN do proposals!

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

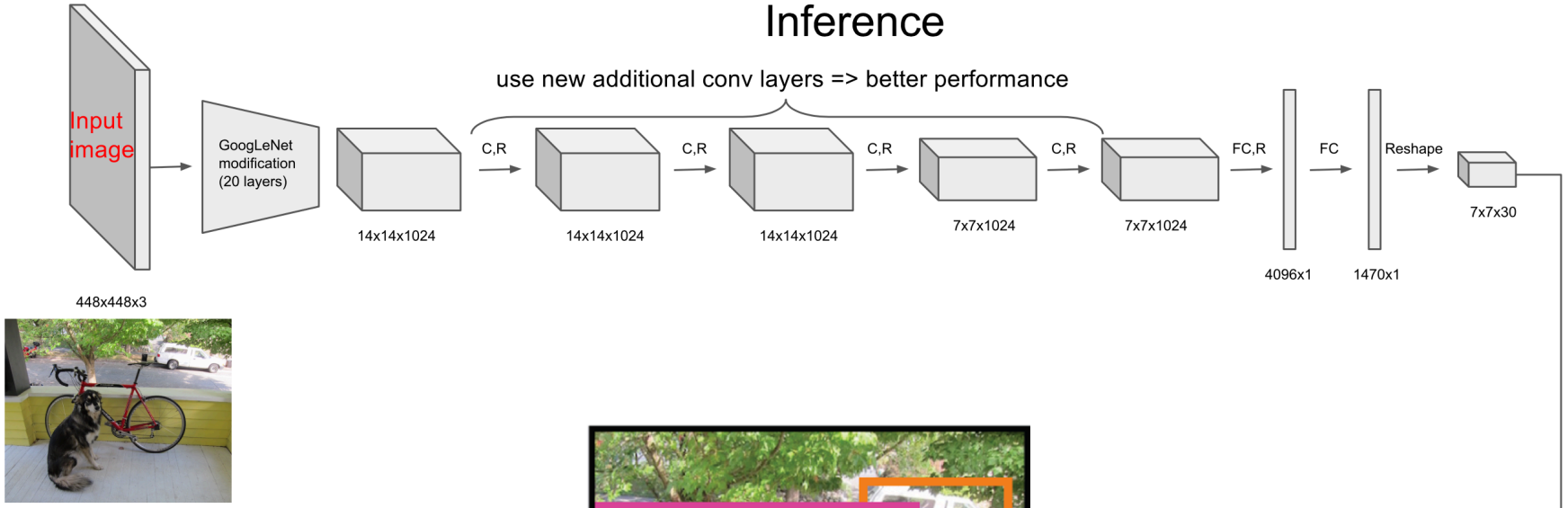
Jointly train with 4 losses:

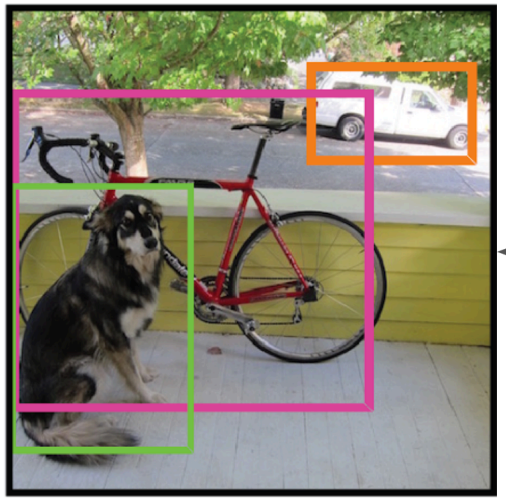
1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates



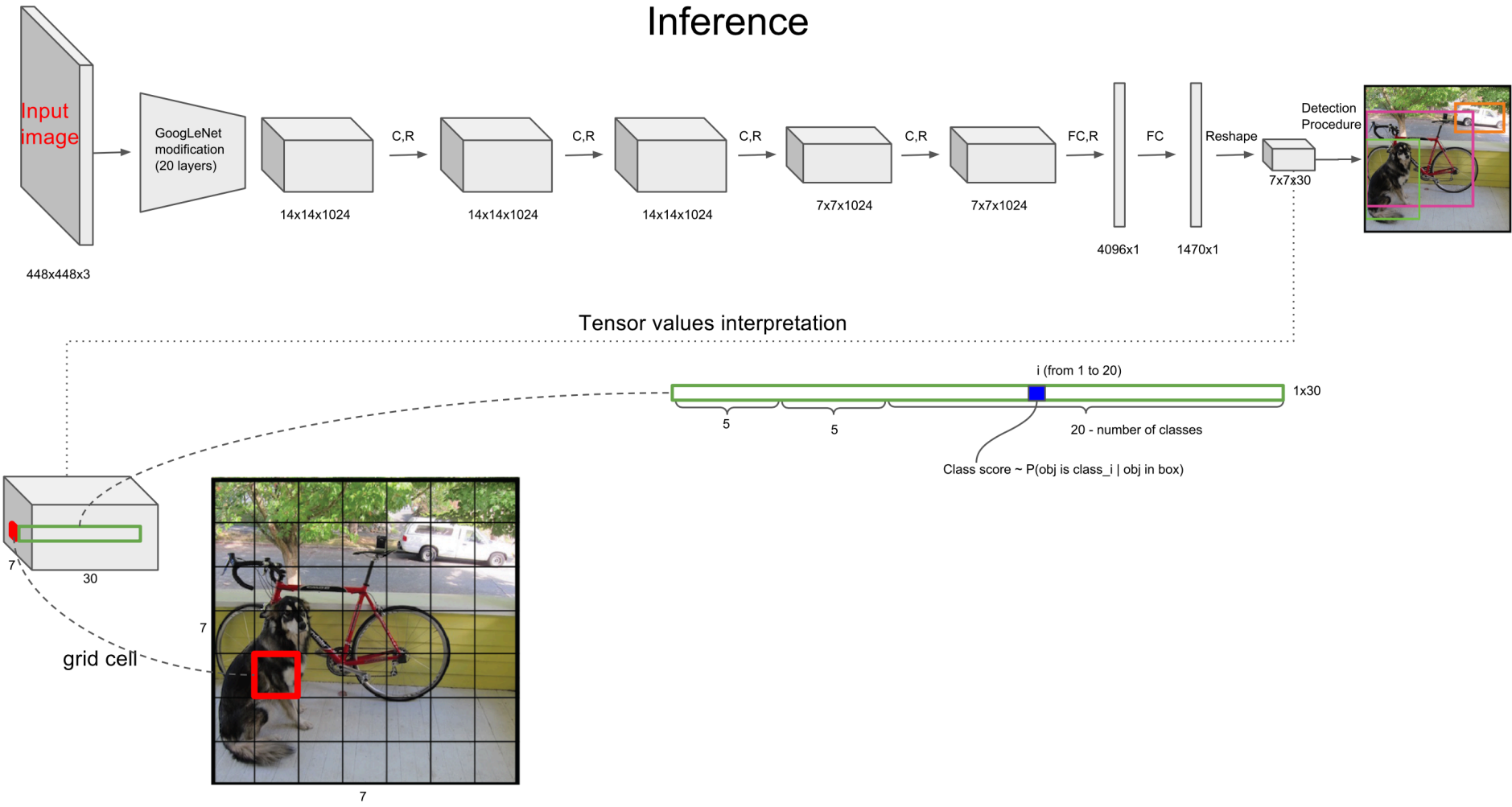
Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

Yolo


Inference
use new additional conv layers => better performance
Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Detection Procedure



Yolo


Inference
Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Detection Procedure
Tensor values interpretation
i (from 1 to 20)
1x30
5
5
20 - number of classes
Class score ~ P(obj is class_i | obj in box)
7
7
30
grid cell
7
7

mAP = mean Average Precision