Segmentation

Deep Learning

Aziz Temirkhanov
Lambda, HSE

Fast recap

Statistical learning theory

$\{(x_i, y_i)\}_{i=1}^{\ell}$ - training set

$x \in \mathbb{R}^d$ - data features

$y \in \mathbb{R}$ - target

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} = \mathcal{L}(\theta; x, y)$$

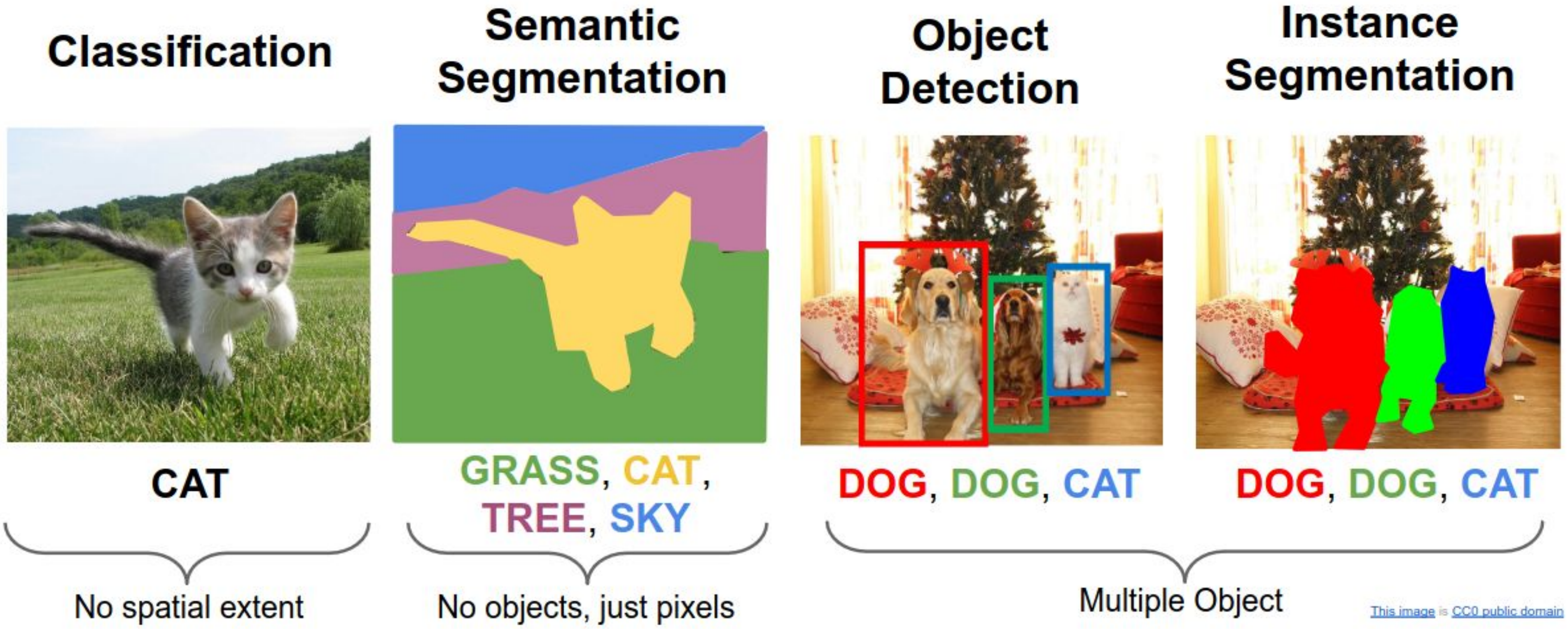
$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(g_\theta(x_i), y_i)$$

STL

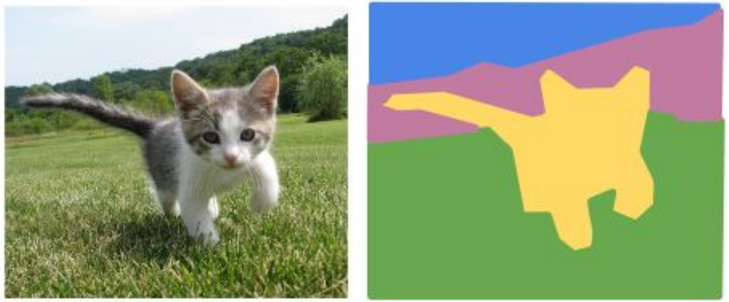
1. Problem is formulated in business language (e.g. “Highlight a tumor within MRI scan”)
2. Reformulate it formally in terms of math and STL (“given an image x, predict segmentation mask y”)
3. Recall a universal approximation theorem
4. Remember that you have a prior knowledge about your data distribution (besides the normality assumption)
5. Fit your function f (your NN) to approximate target function g (your PDF or other law)
6. During fitting, you have to minimize the loss function (ERM). Once it minimized, and other formalities are satisfied, you are done!

Image segmentation

Computer Vision Tasks

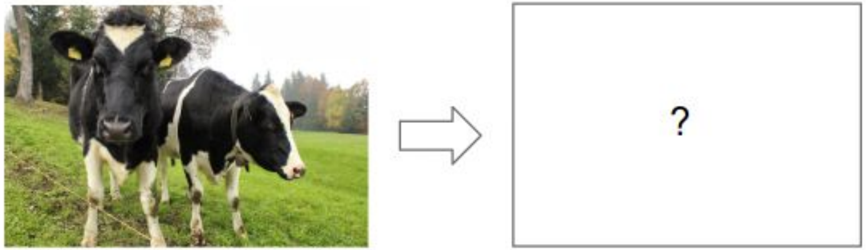


Formulation



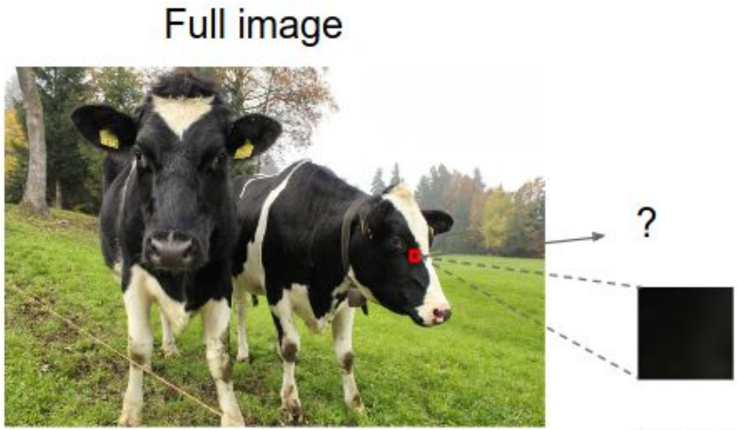
GRASS, CAT,
TREE, SKY, ...

Paired training data: for each training image, each pixel is labeled with a semantic category.



At test time, classify each pixel of a new image.

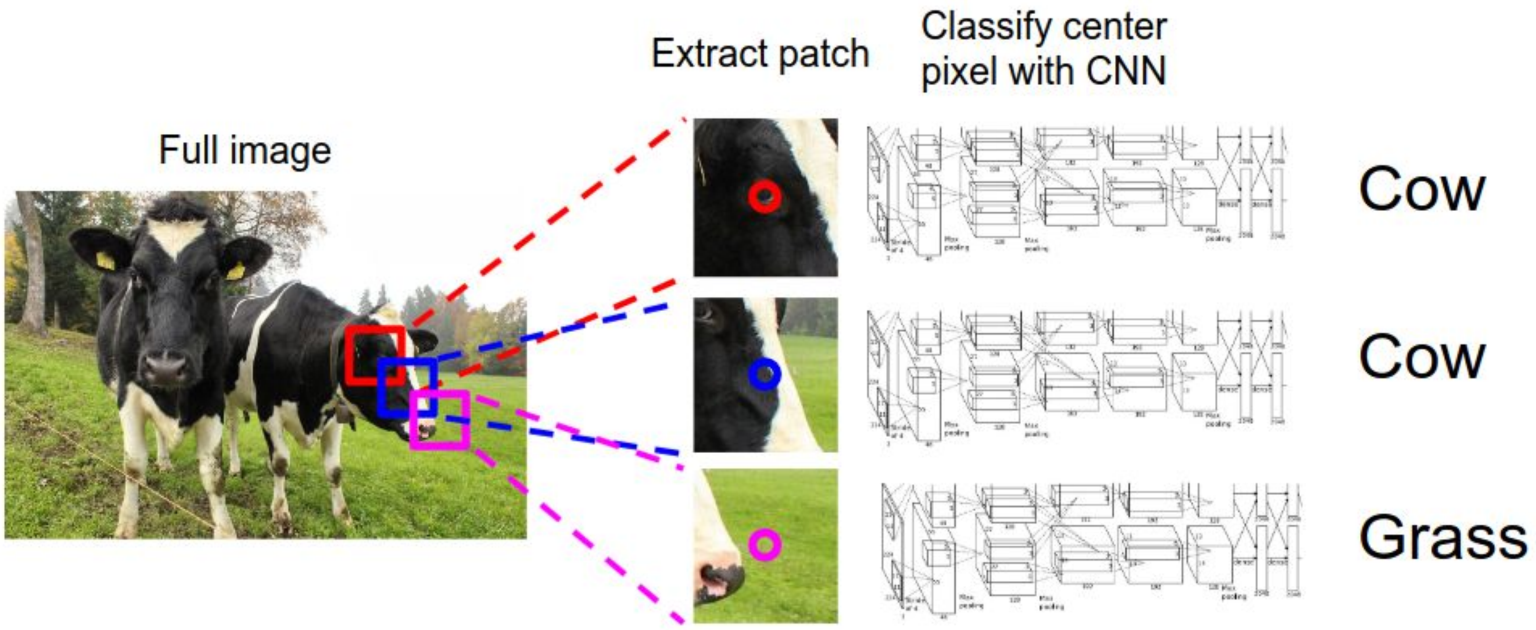
Problem



Impossible to classify without context

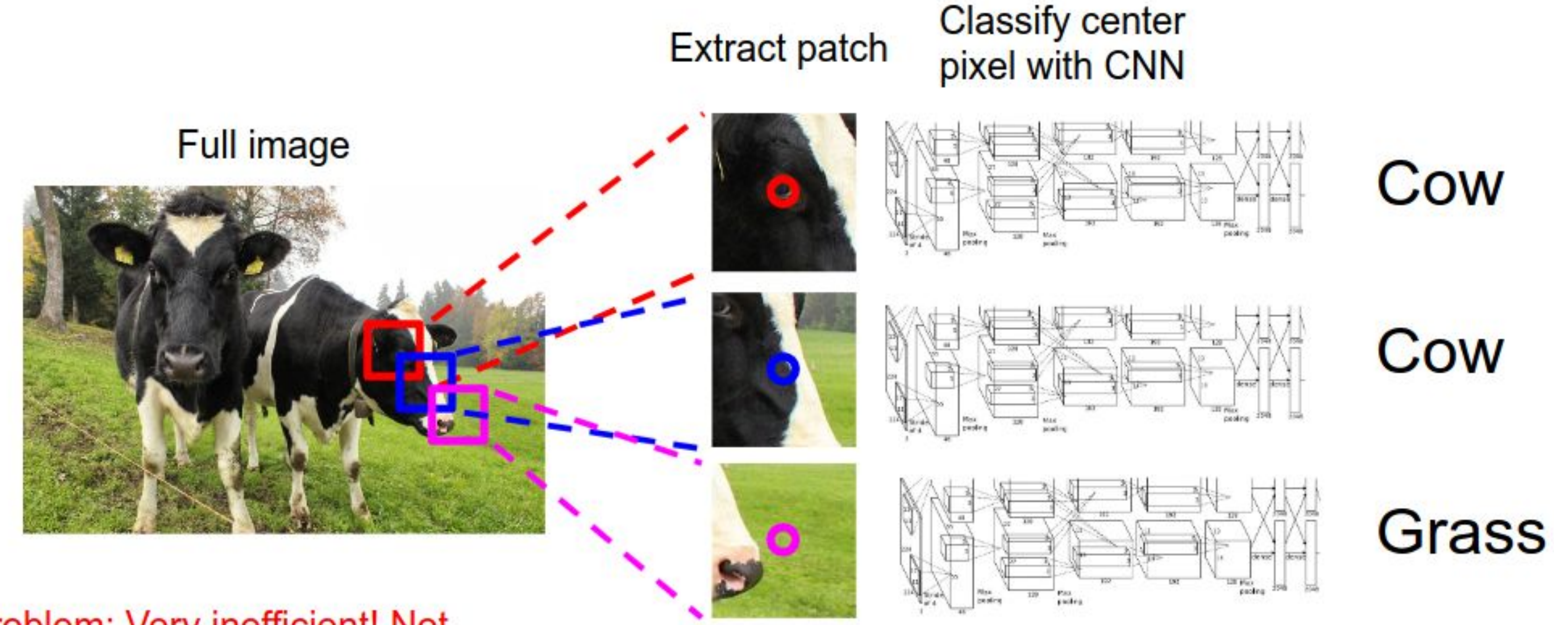
Q: how do we include context?

Context



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

Inefficiency

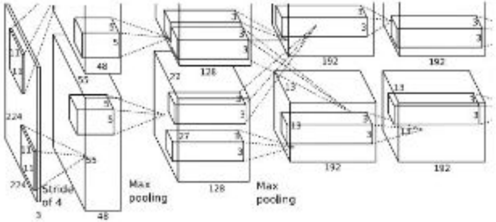


Problem: Very inefficient! Not reusing shared features between overlapping patches

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

Shared Convolutional features

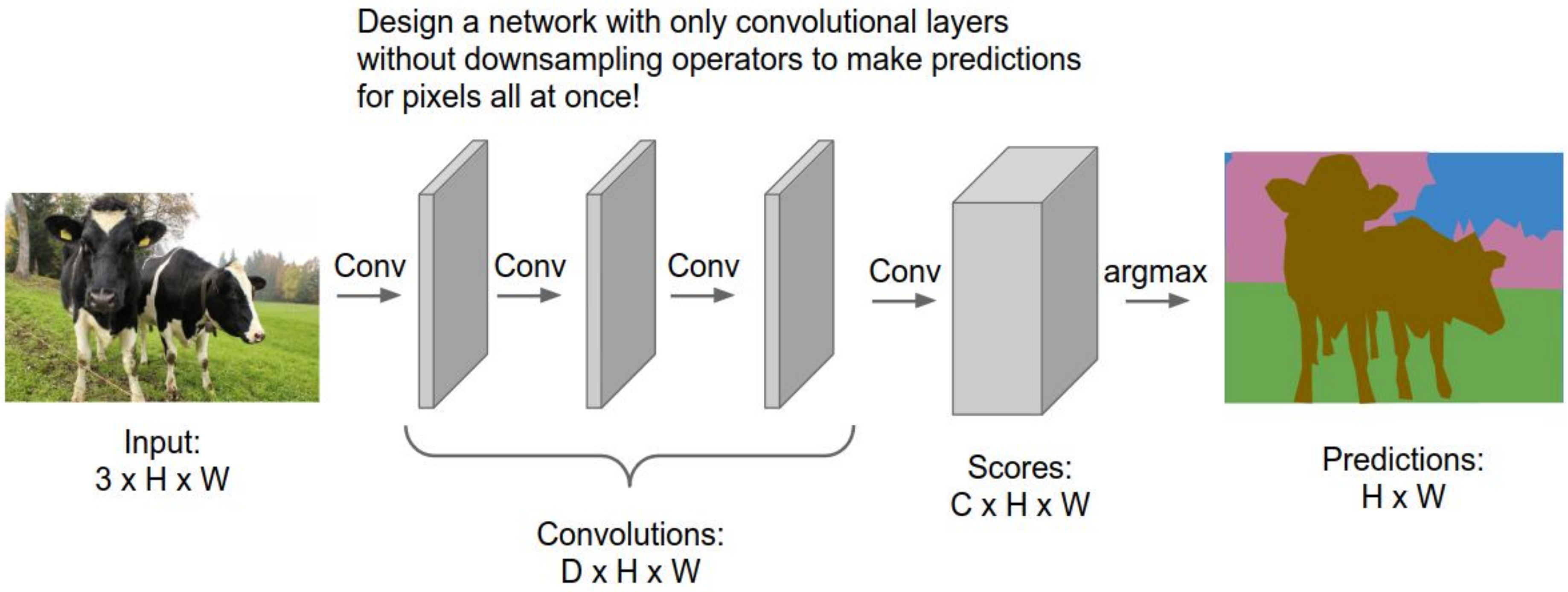
Full image



An intuitive idea: encode the entire image with conv net, and do semantic segmentation on top.

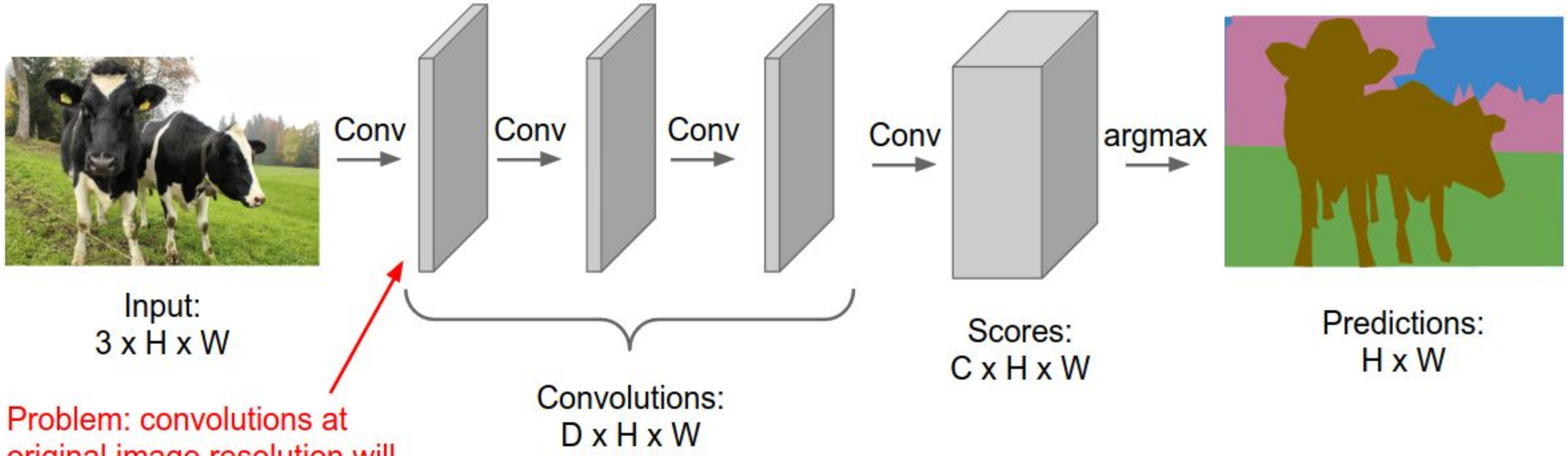
Problem: classification architectures often reduce feature spatial sizes to go deeper, but semantic segmentation requires the output size to be the same as input size.

CNN Segmentation



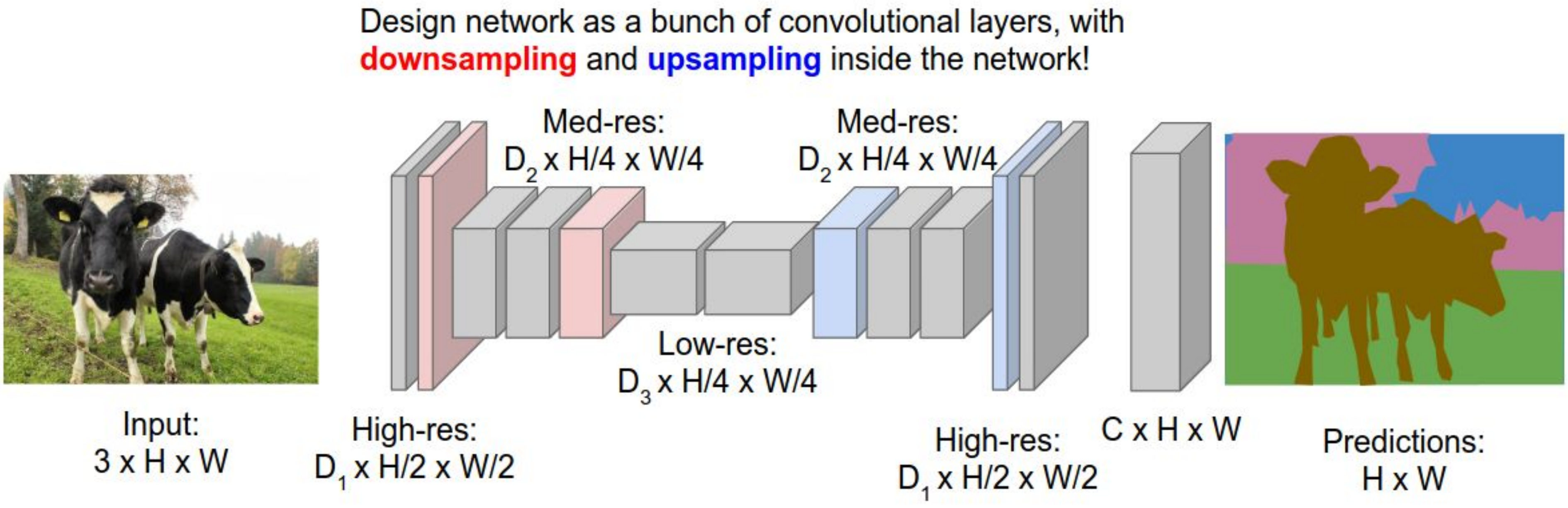
CNN Segmentation

Design a network with only convolutional layers without downsampling operators to make predictions for pixels all at once!



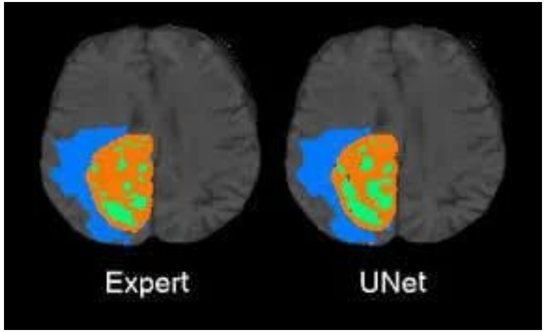
Problem: convolutions at original image resolution will be very expensive ...

CNN Segmentation

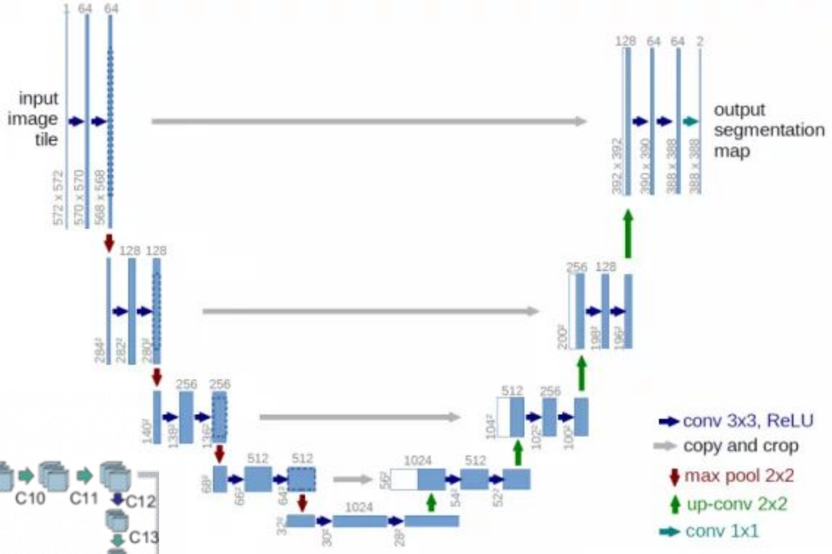


Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

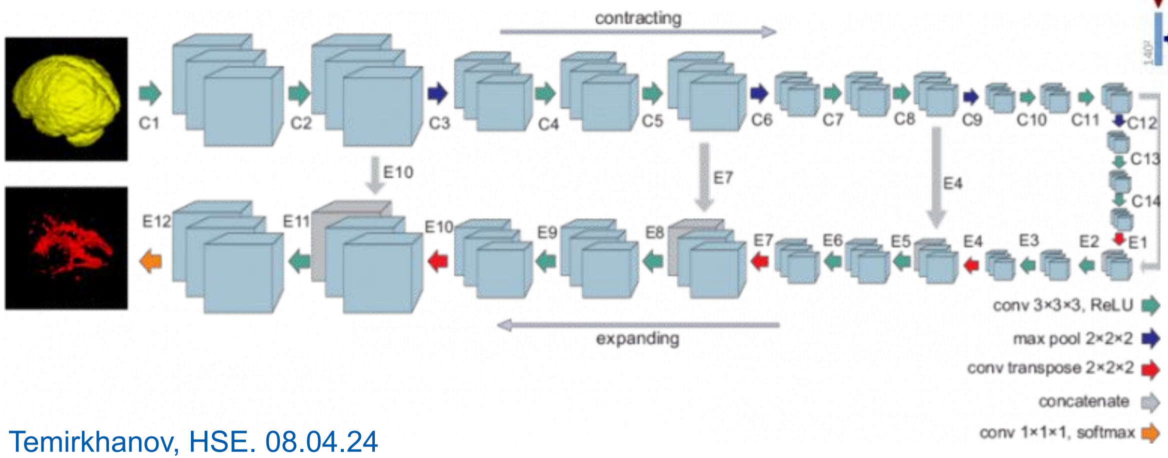
UNet



Expert
UNet

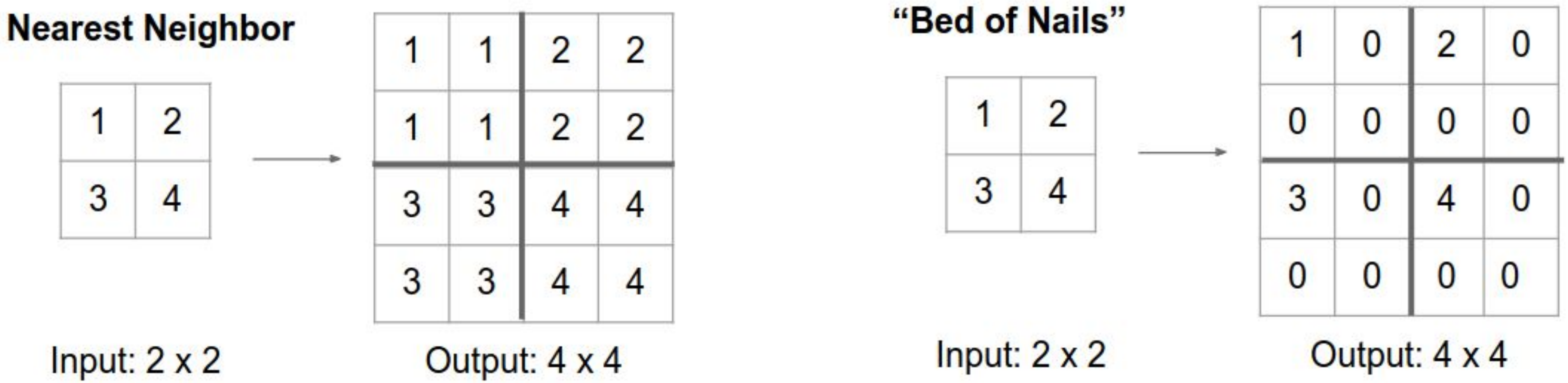


input
image
tile
output
segmentation
map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

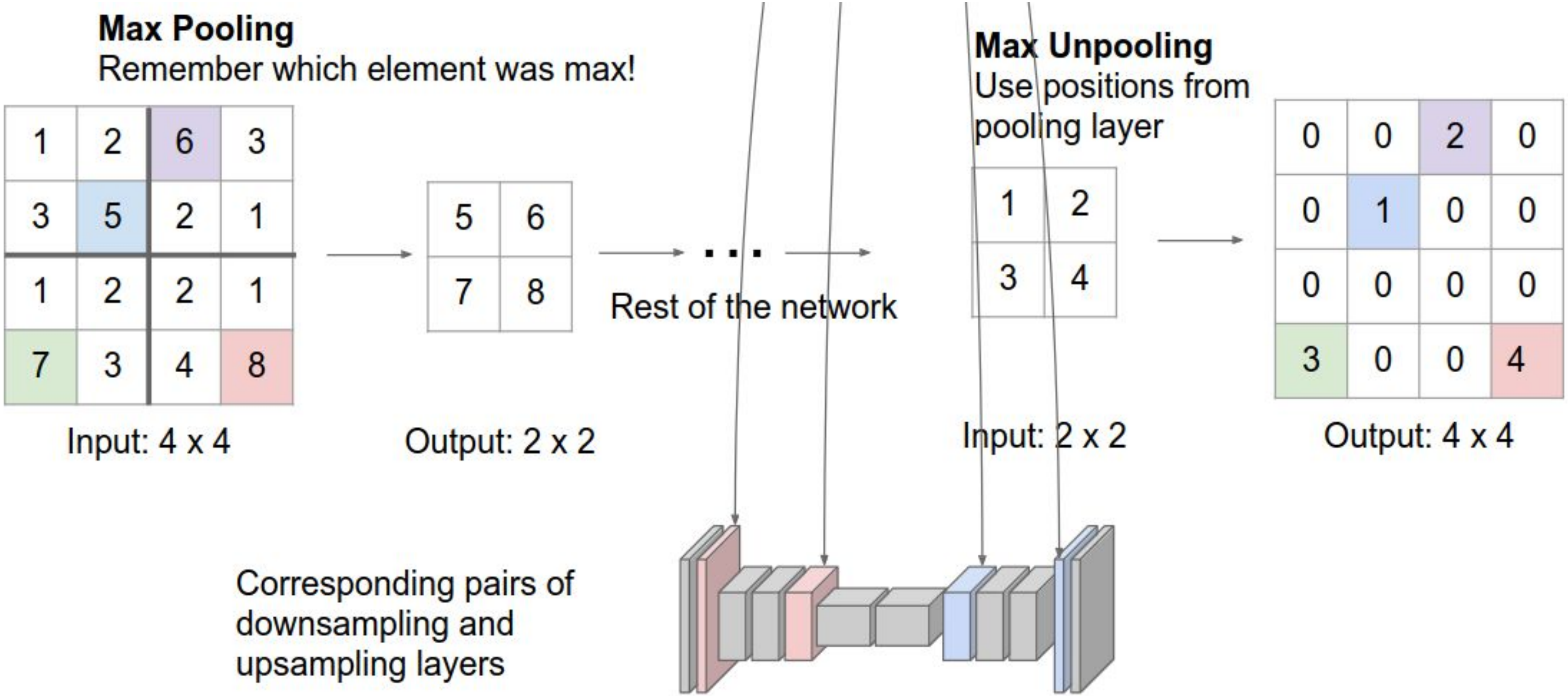


contracting
expanding
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13
C14
E1
E2
E3
E4
E5
E6
E7
E8
E9
E10
E11
E12
conv 3×3×3, ReLU
max pool 2×2×2
conv transpose 2×2×2
concatenate
conv 1×1×1, softmax

Unpooling

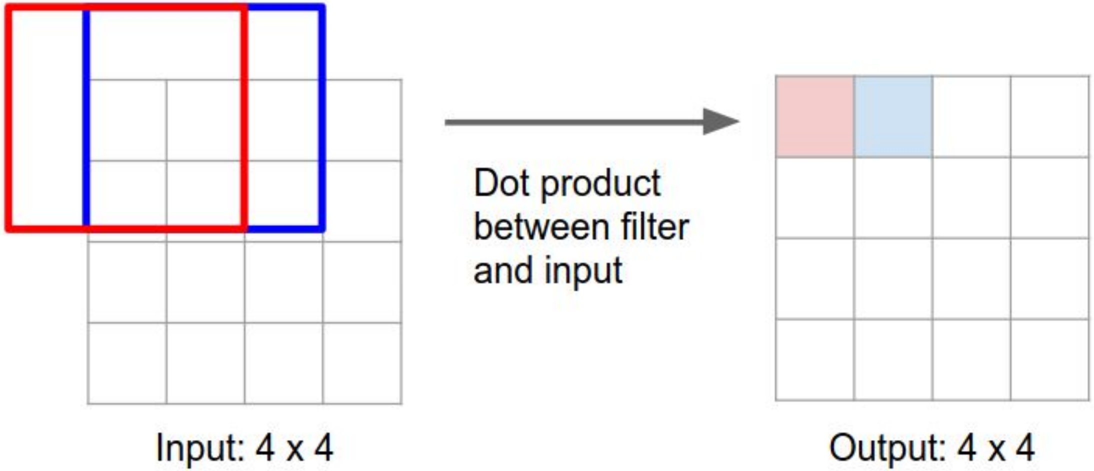

Nearest Neighbor
1 2
3 4
1 1 2 2
1 1 2 2
3 3 4 4
3 3 4 4
Input: 2 x 2
Output: 4 x 4
"Bed of Nails"
1 2
3 4
1 0 2 0
0 0 0 0
3 0 4 0
0 0 0 0
Input: 2 x 2
Output: 4 x 4

Max unpooling



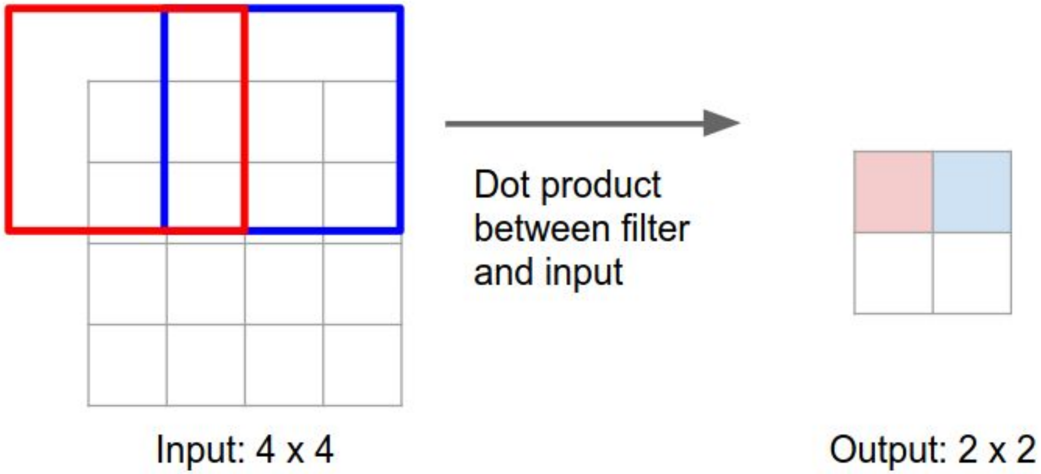
Transpose Convolution

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



Transpose Convolution

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

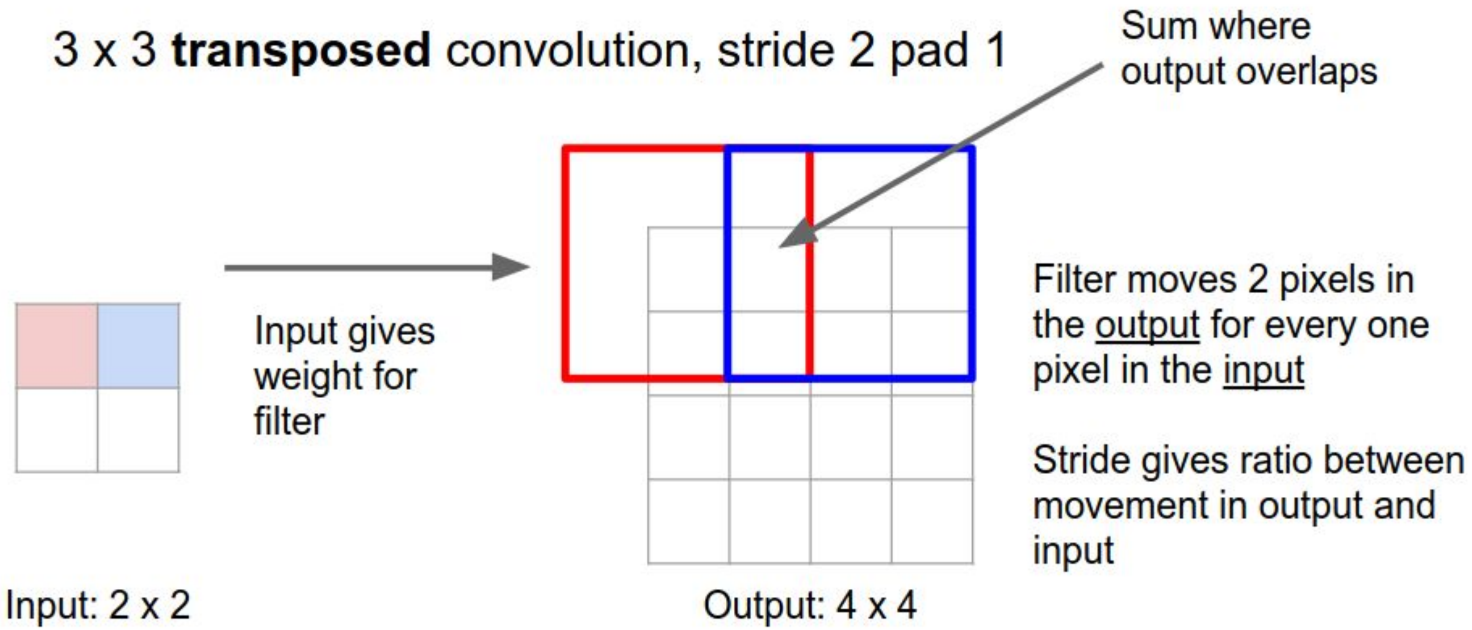


Filter moves 2 pixels in the input for every one pixel in the output

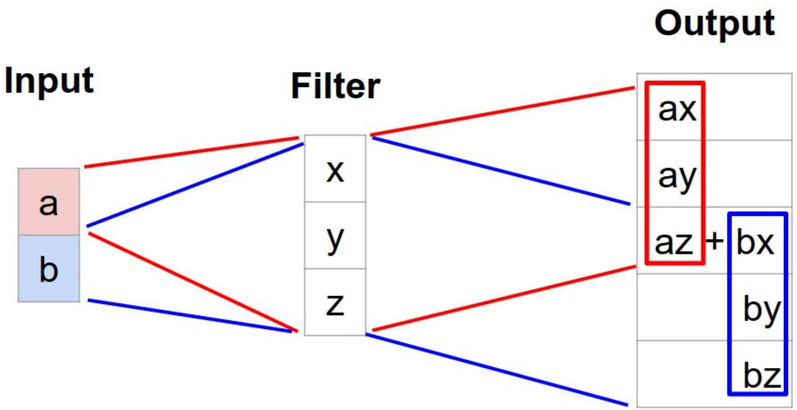
Stride gives ratio between movement in input and output

We can interpret strided convolution as “learnable downsampling”.

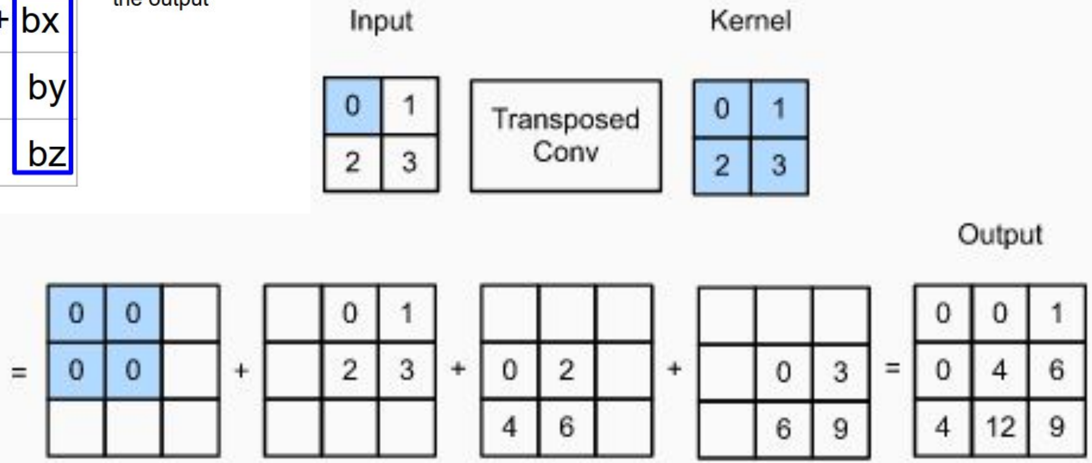
Transposed Convolution



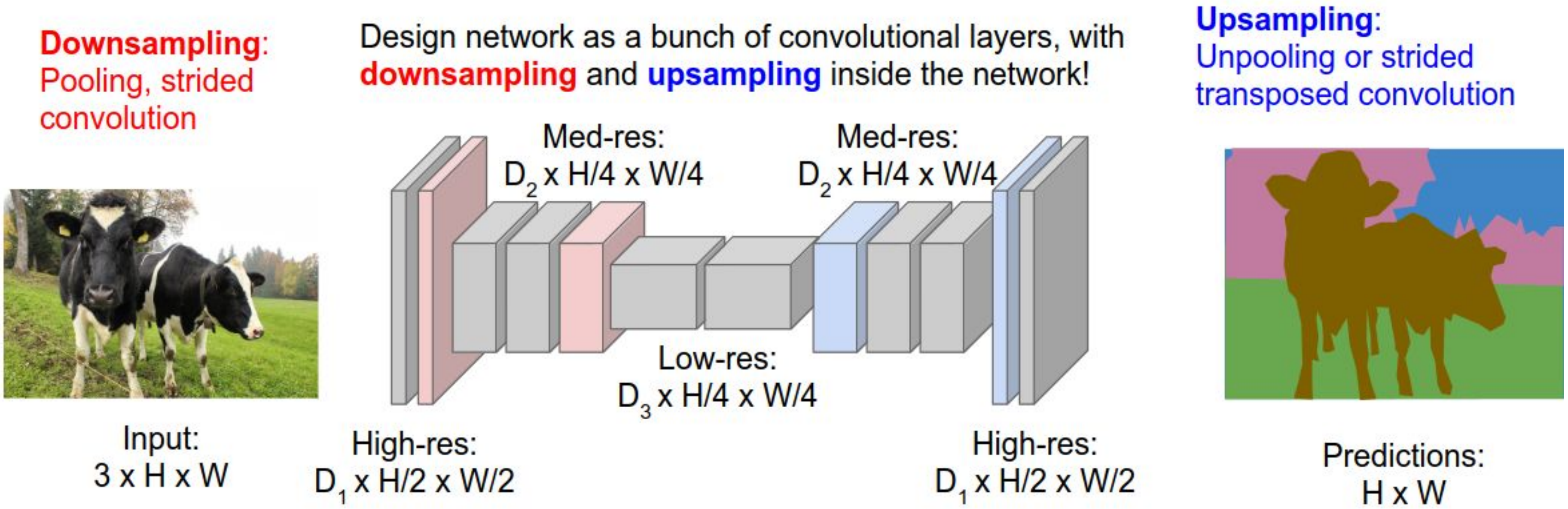
Transposed Convolution



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



Semantic Segmentation

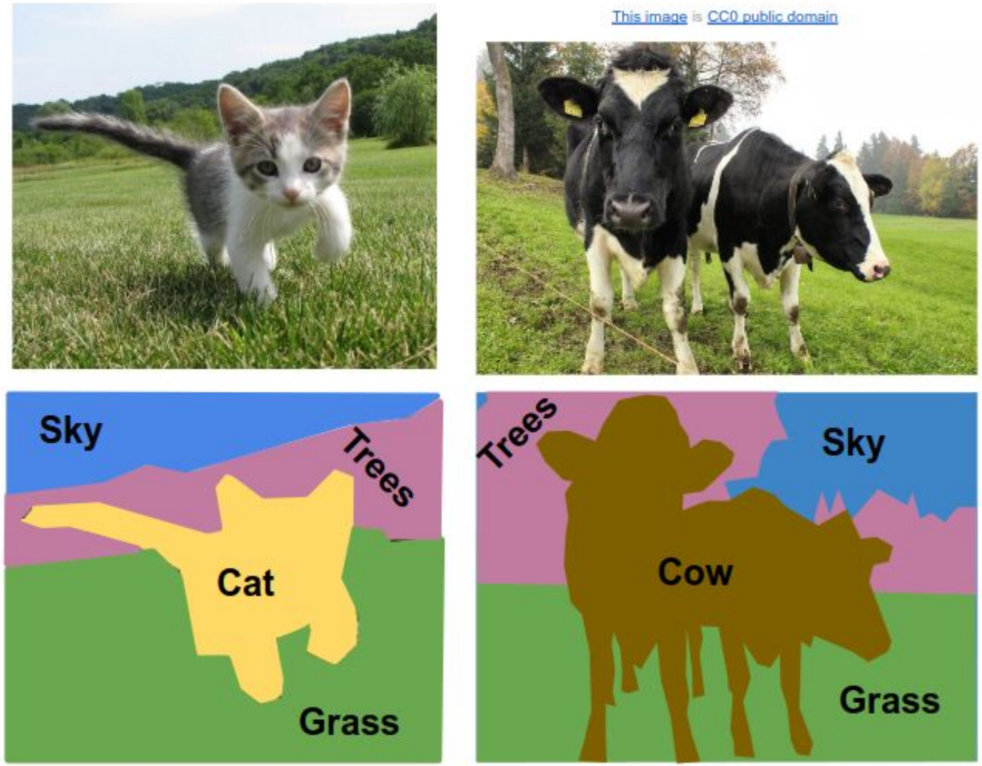


Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

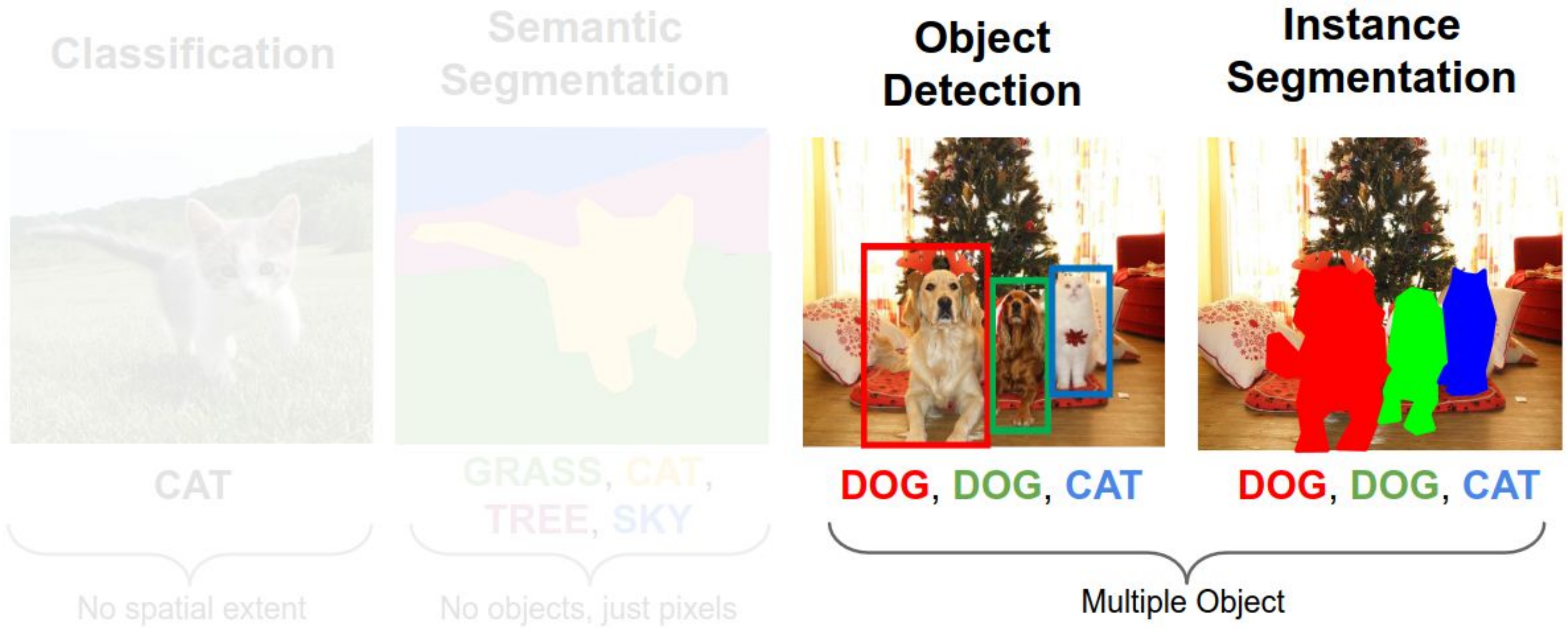
Semantic Segmentation

Label each pixel in the image with a category label

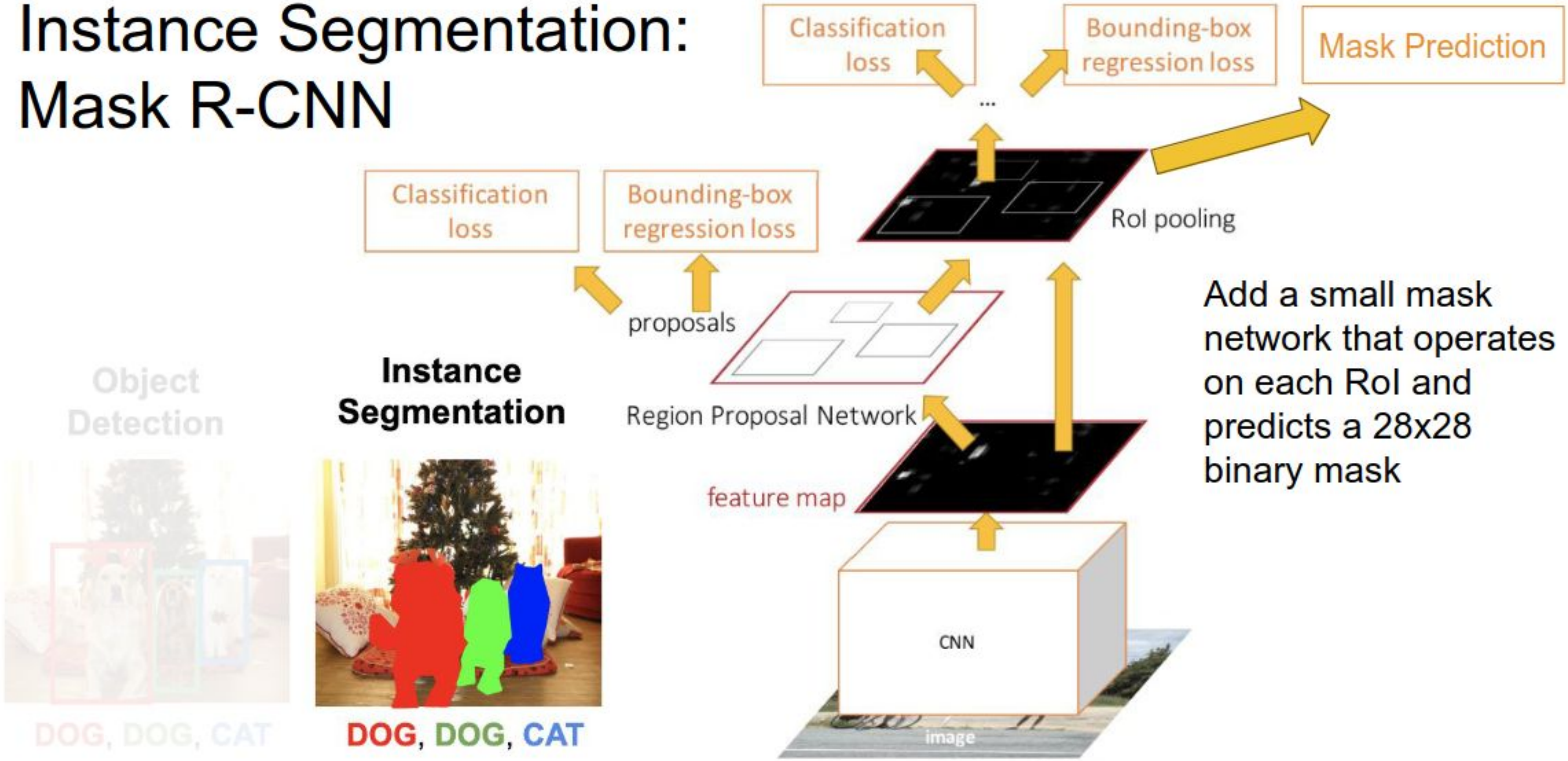
Don’t differentiate instances, only care about pixels



Instance Segmentation



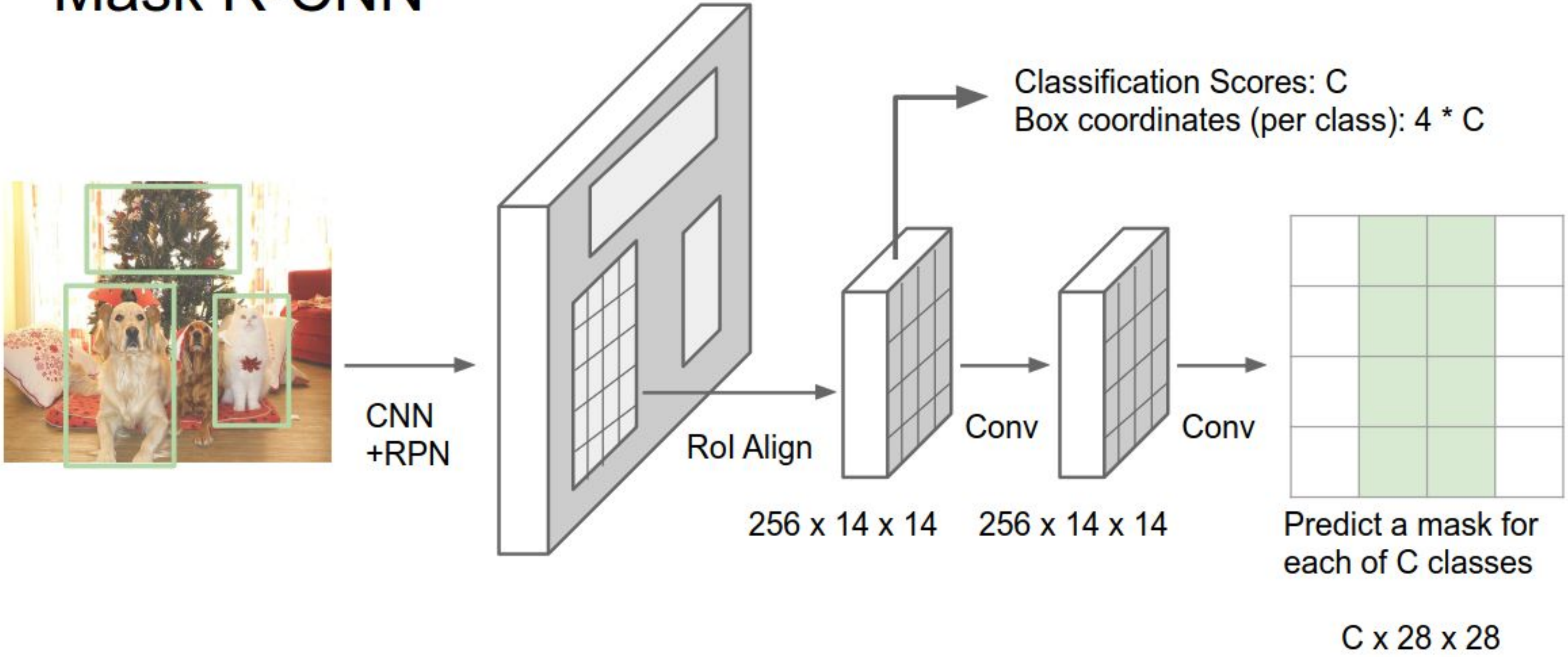
Instance Segmentation: Mask R-CNN



Add a small mask network that operates on each RoI and predicts a 28x28 binary mask

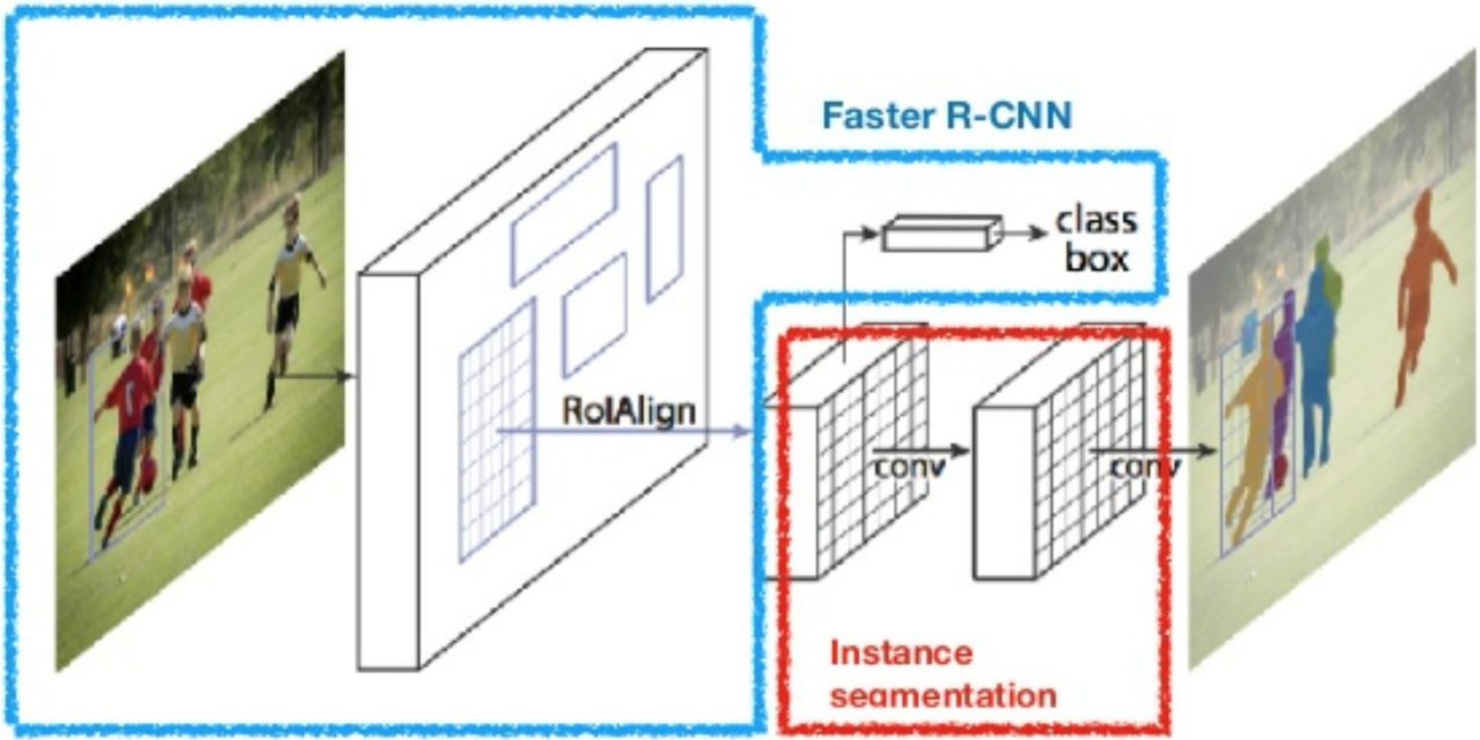
He et al, "Mask R-CNN", ICCV 2017

Mask R-CNN

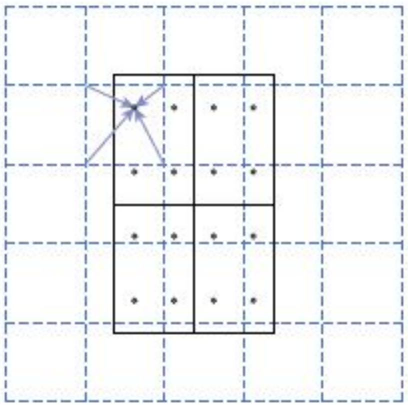


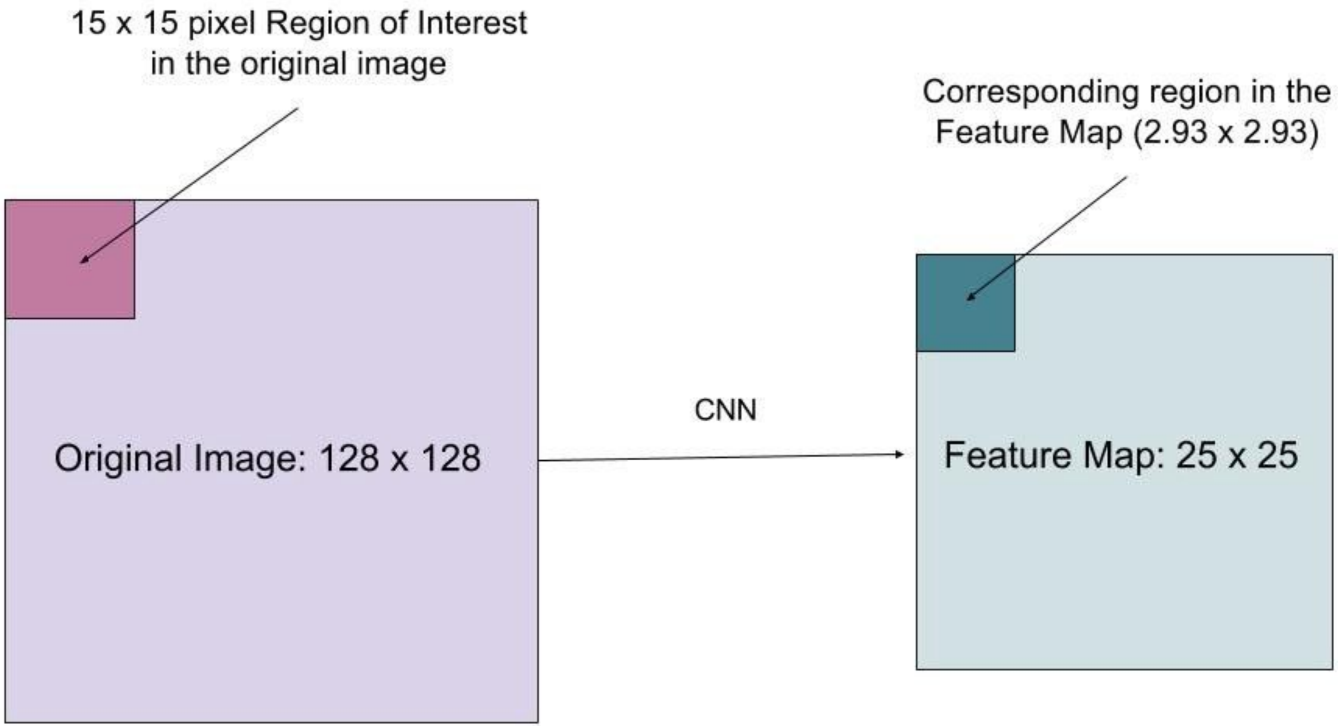
He et al, "Mask R-CNN", arXiv 2017

Mask R-CNN


Faster R-CNN
class
box
RoIAlign
conv
conv
Instance
segmentation

ROI Alignment

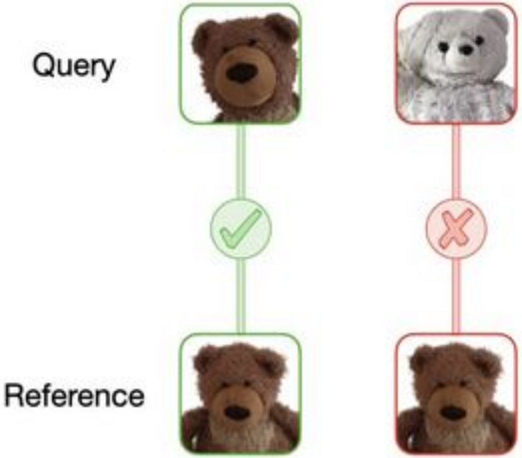
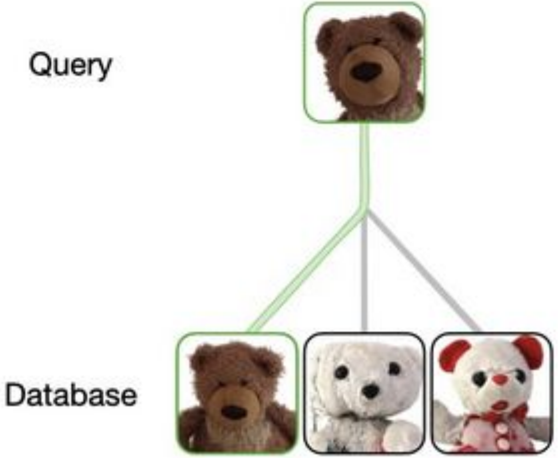




Bonus part

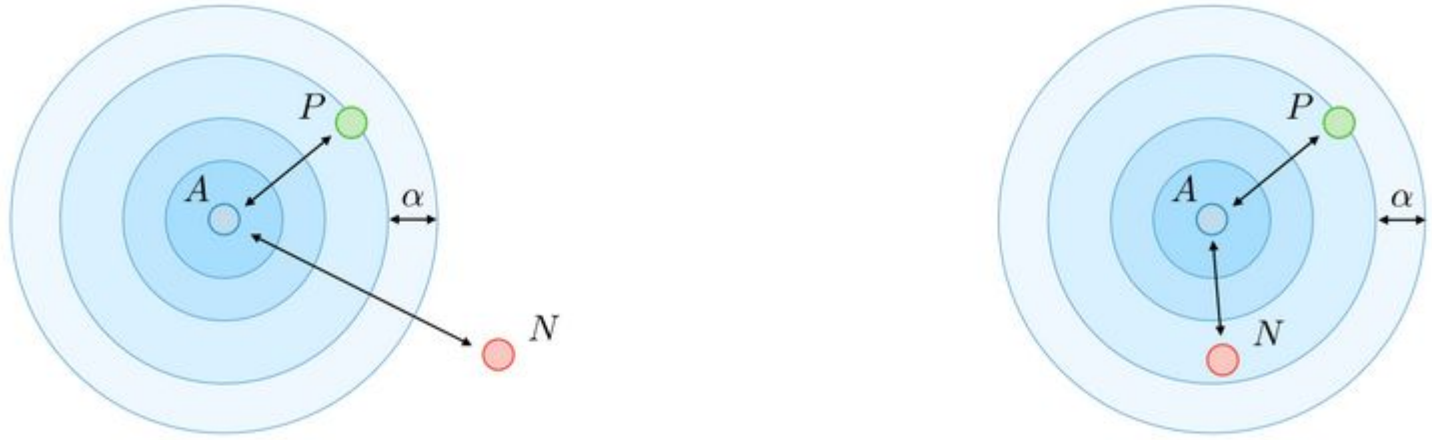
Addition vision tasks

Face verification and recognition

Face verification	Face recognition
• Is this the correct person? • One-to-one lookup	• Is this one of the K persons in the database? • One-to-many lookup
	

Triplet Loss

$$\boxed{\ell(A, P, N) = \max\left(d(A, P) - d(A, N) + \alpha, 0\right)}$$



$\ell(A, P, N) = 0$ $\qquad\qquad$ $\ell(A, P, N) > 0$

Style Transfer and Domain Adaptation