Machine Learning for Image Analysis

Lecture 6: CNNs for Semantic Segmentation

Dagmar Kainmueller, 27.05.2024

HPI Hasso Plattner Institut
Digital Engineering · Universität Potsdam
MAX DELBRÜCK CENTER

Course Outline

- Introduction to Image Analysis
- Basics: Neural Networks
- **Convolutional Neural Networks**
- Transformers
- Model Interpretability
- Self-supervised Learning
- Generative Models (GANs, Diffusion)

Convolutional Neural Networks (CNNs)

- A bit of CNN history
- CNN Layers
    - Convolutions, Pooling, Fully Connected Layers, Normalization
- Training CNNs
    - Advanced Optimization
    - Weight initialization
    - Regularization: Dropout
    - Extended training data: Augmentation, Transfer Learning
    - Sanity-checking the learning process; Hyperparameter tuning strategies
- Famous CNN architectures
- CNNs for image segmentation; CNNs for object detection

Recap: Sanity-checking the Learning Process

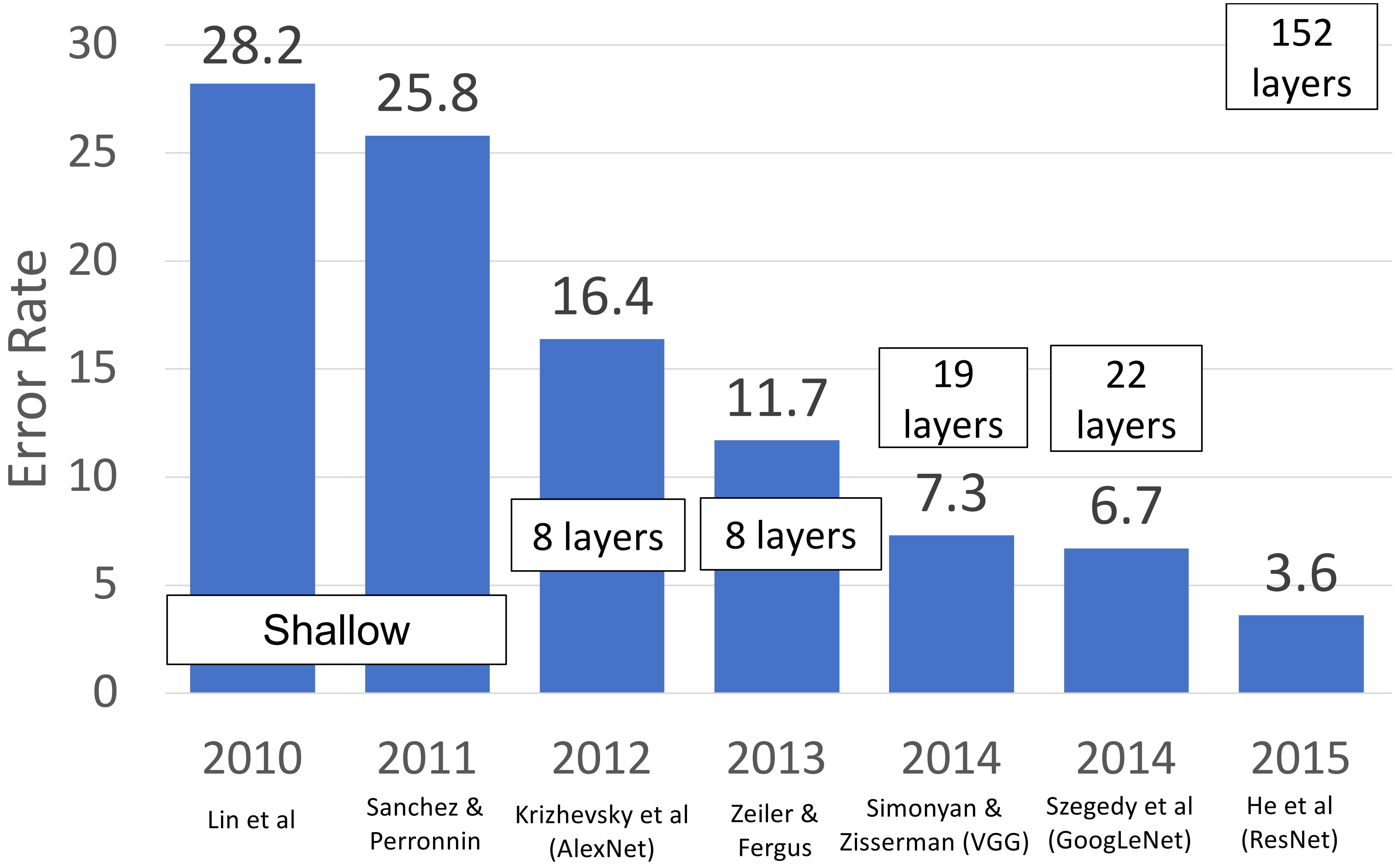
1. Make sure the initial loss makes sense, i.e., close to expected value

2. Make sure you can overfit a small portion of the training data

3. Use random layout for multi-dimensional hyper-parameter search; start coarse / few epochs, move on to finer / longer training as appropriate

4. Visually inspect training- and validation images: Which ones yield highest loss? Which ones yield lowest loss? Does this allow you to hypothesize what might improve your approach?

ImageNet Classification Challenge



Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2019

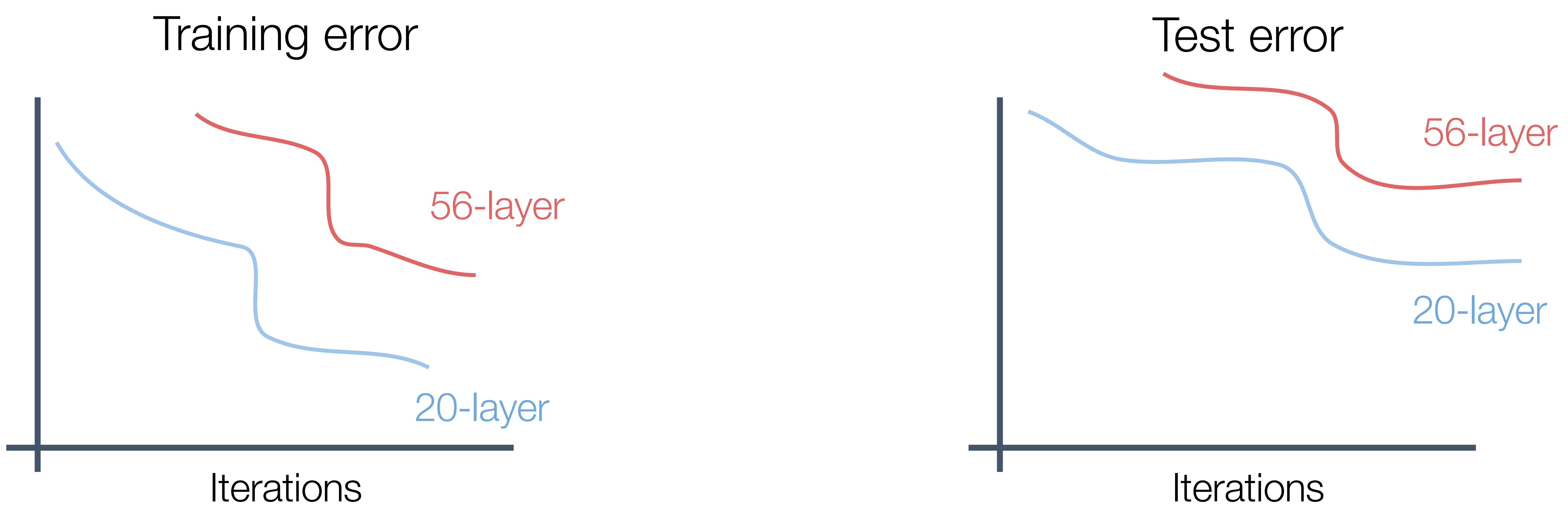
Residual Networks

Once we have Batch Normalization, we can train networks with 10+ layers.
What happens as we go deeper?

He et al, "Deep Residual Learning for Image Recognition", CVPR 2016

Residual Networks

Once we have Batch Normalization, we can train networks with 10+ layers.
What happens as we go deeper?



In fact the deep model seems to be **underfitting** since it also performs worse than the shallow model on the training set! It is actually **underfitting**

He et al, “Deep Residual Learning for Image Recognition”, CVPR 2016

Residual Networks

A deeper model can emulate a shallower model: copy layers from shallower model, set extra layers to identity

Thus deeper models should do at least as good as shallow models

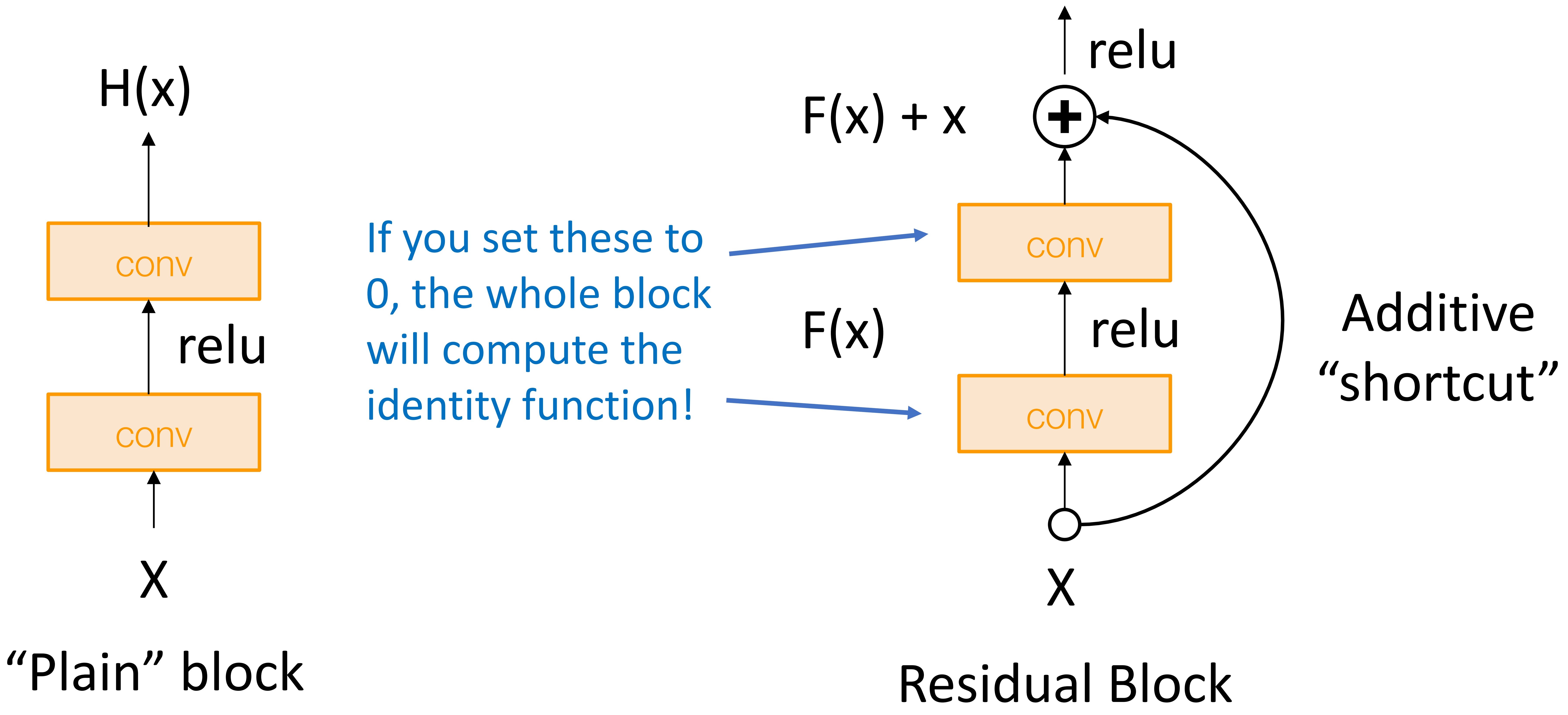
Hypothesis: This is an optimization problem. Deeper models are harder to optimize, and in particular don't learn identity functions to emulate shallow models

Solution: Change the network so learning identity functions with extra layers is easy!

He et al, "Deep Residual Learning for Image Recognition", CVPR 2016

Residual Networks

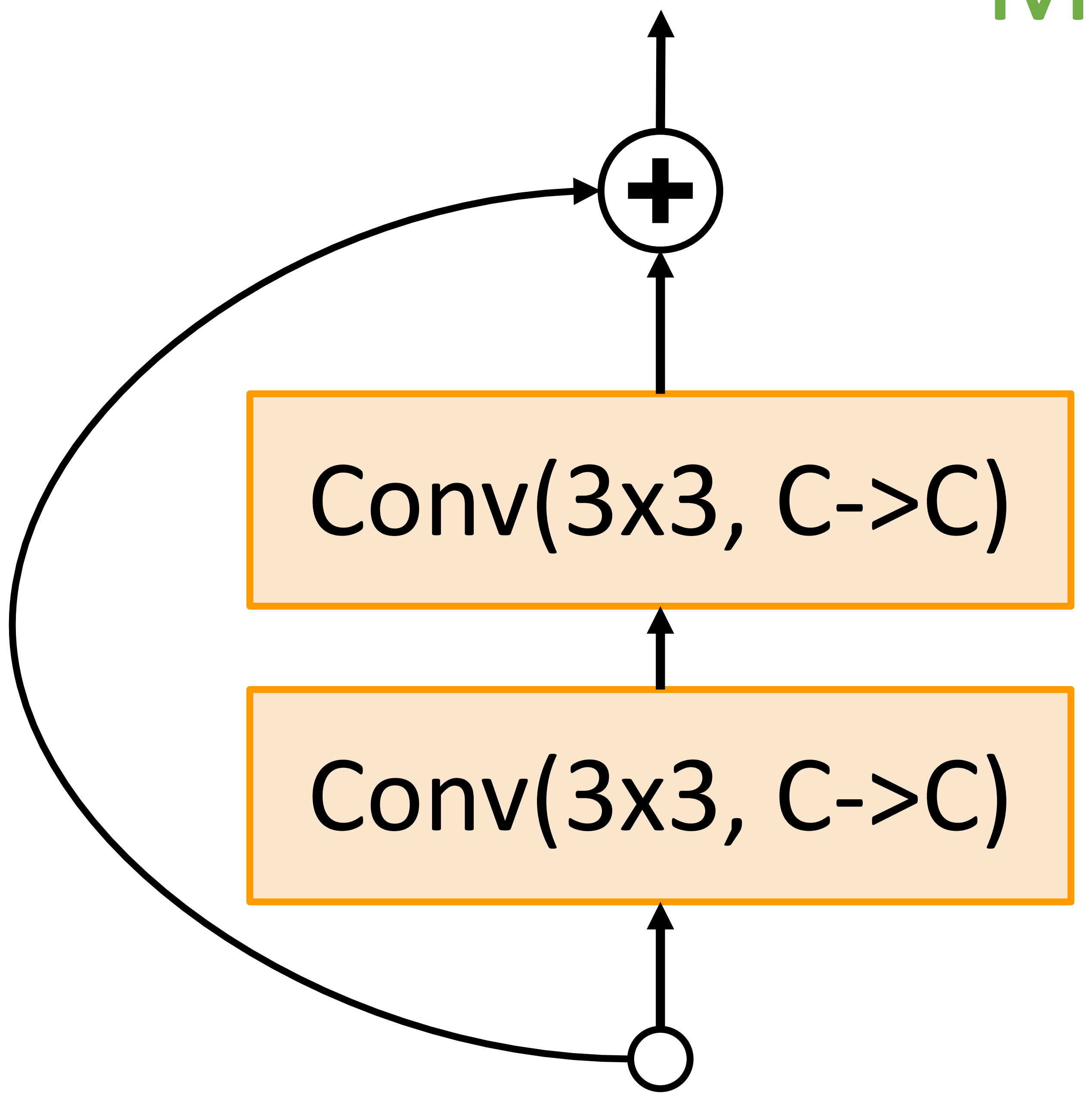
Solution: Change the network so learning identity functions with extra layers is easy!



He et al, "Deep Residual Learning for Image Recognition", CVPR 2016

Residual Networks: Bottleneck Block

More layers, less computational cost!



FLOPs: $9HWC^2$

FLOPs: $9HWC^2$

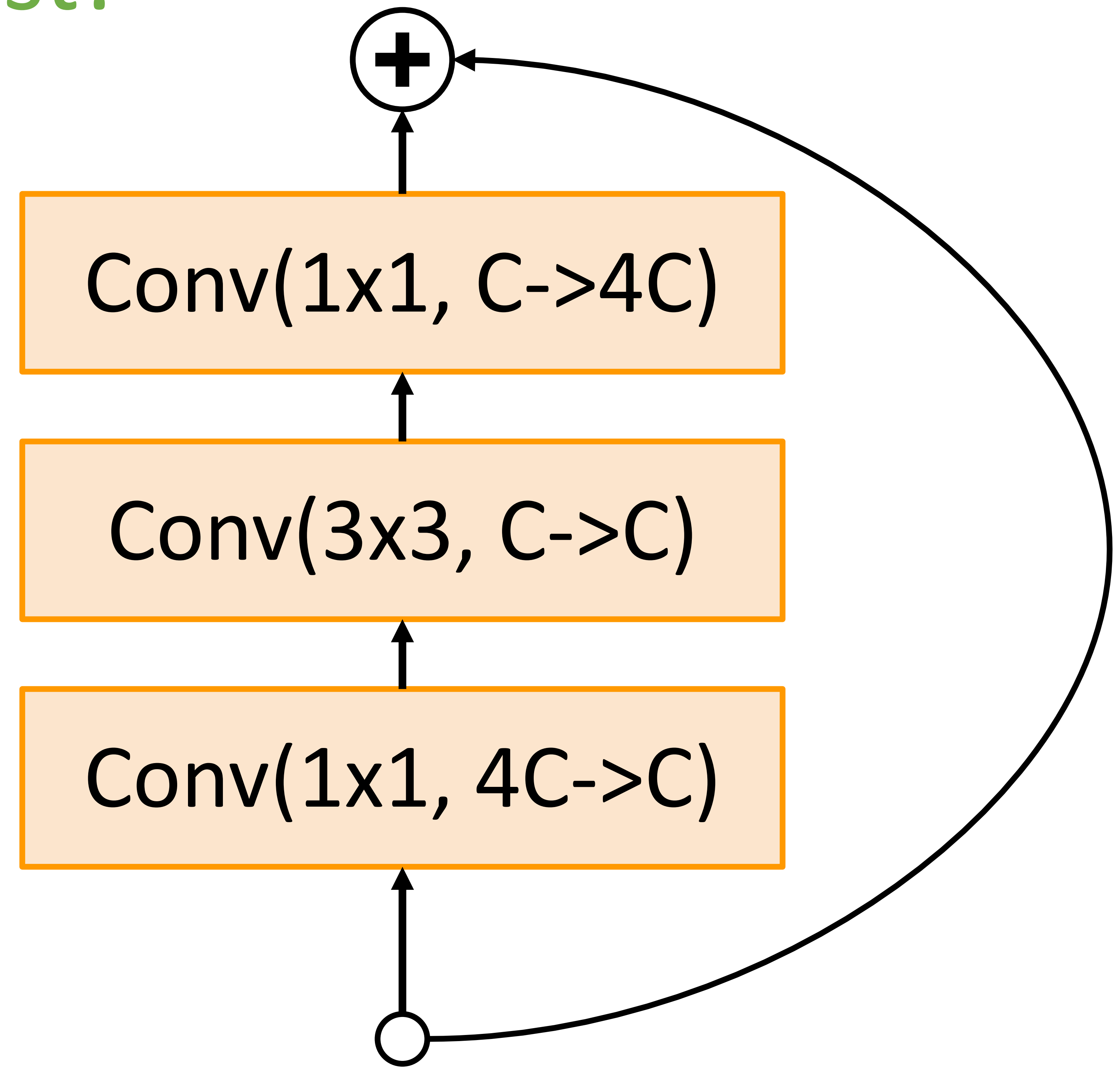
“Basic”
Residual block

Total FLOPs:
$18HWC^2$

FLOPs: $4HWC^2$

FLOPs: $9HWC^2$

FLOPs: $4HWC^2$



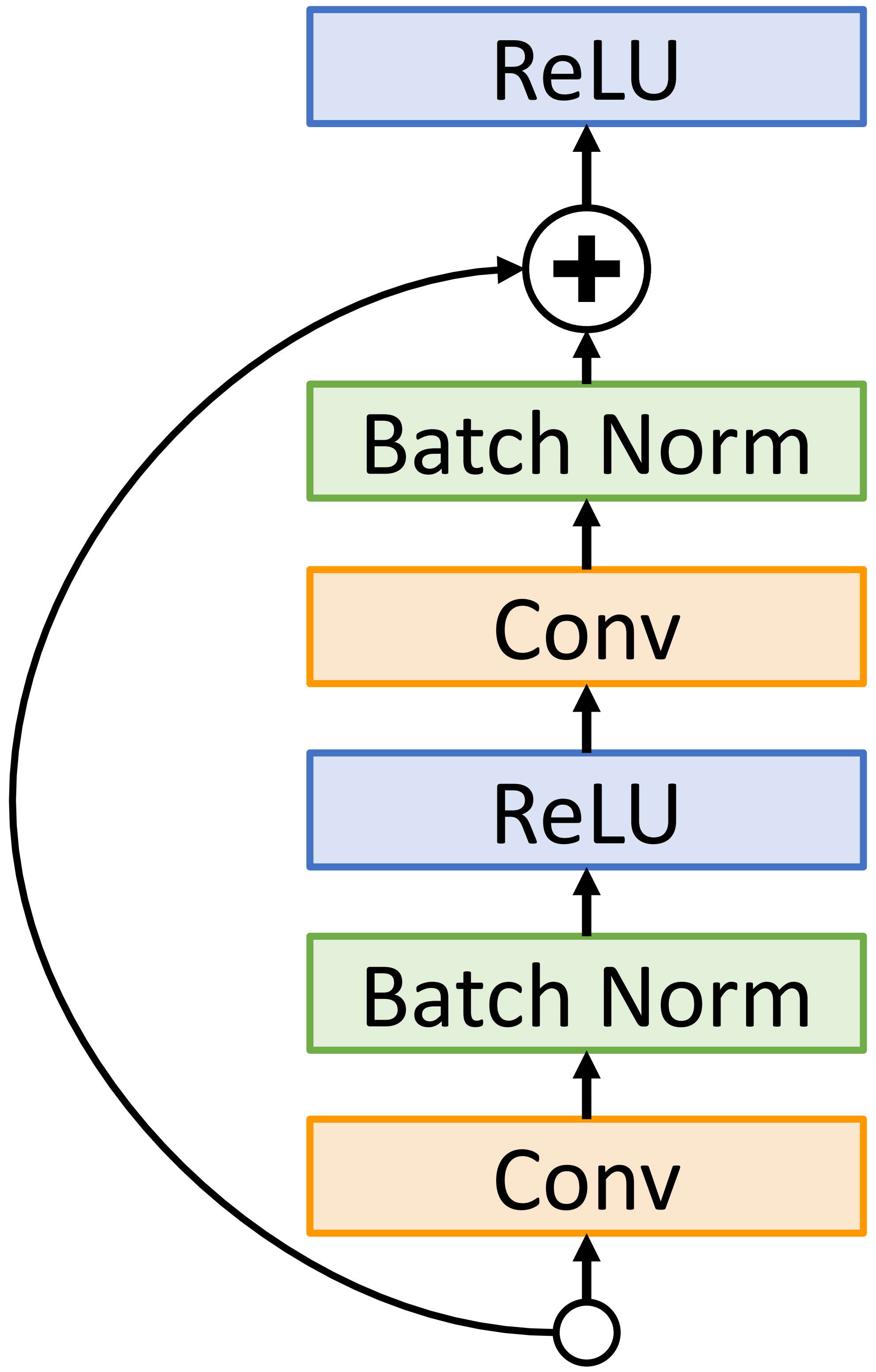
Total FLOPs:
$17HWC^2$

“Bottleneck”
Residual block

He et al, “Deep Residual Learning for Image Recognition”, CVPR 2016

Improving Residual Networks: Block Design

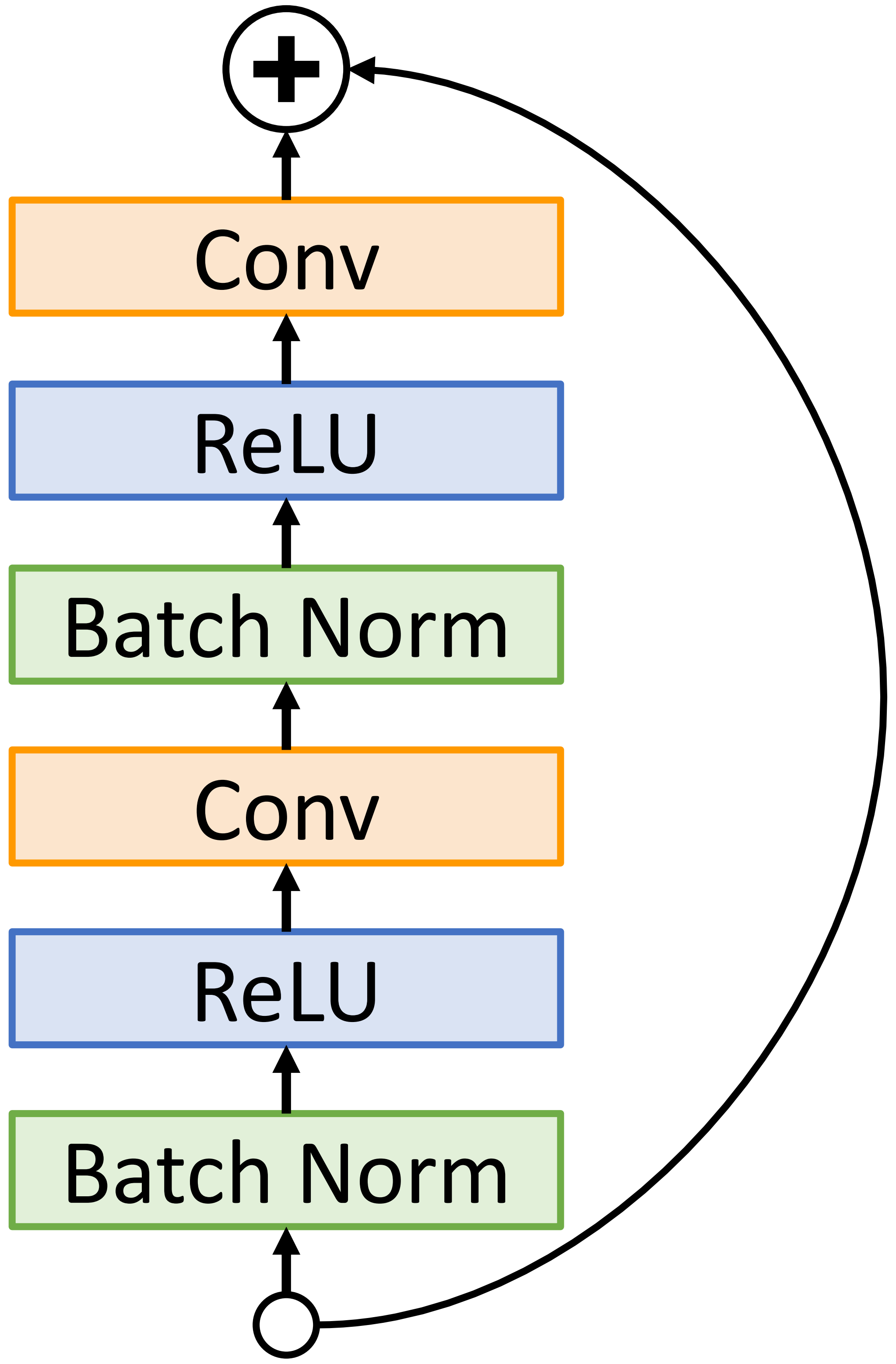
Original ResNet block



Note ReLU **after** residual:

Cannot actually learn identity function since outputs are nonnegative!

"Pre-Activation" ResNet Block

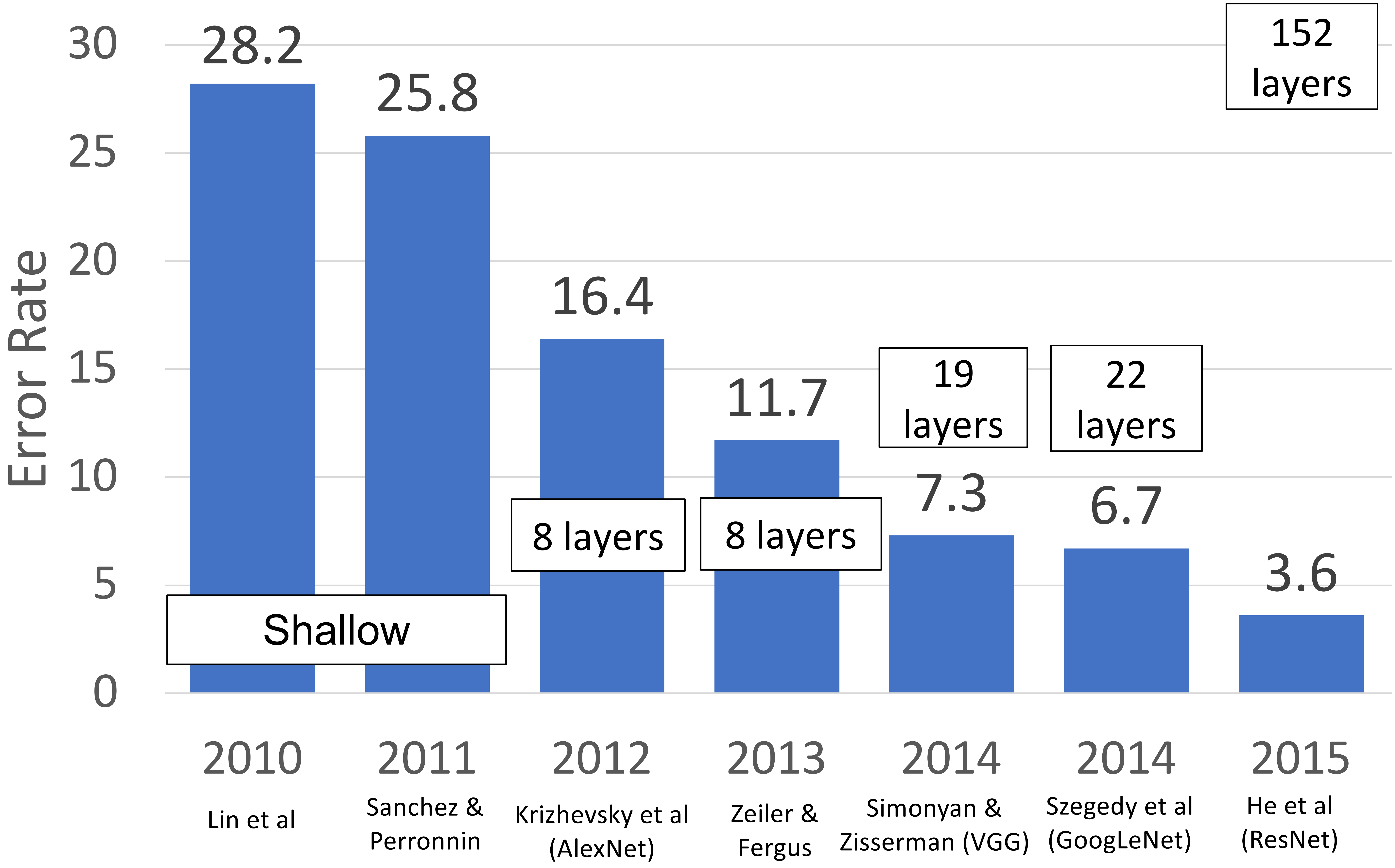


Note ReLU **inside** residual:

Can learn true identity function by setting Conv weights to zero!

He et al, "Identity mappings in deep residual networks", ECCV 2016

ImageNet Classification Challenge

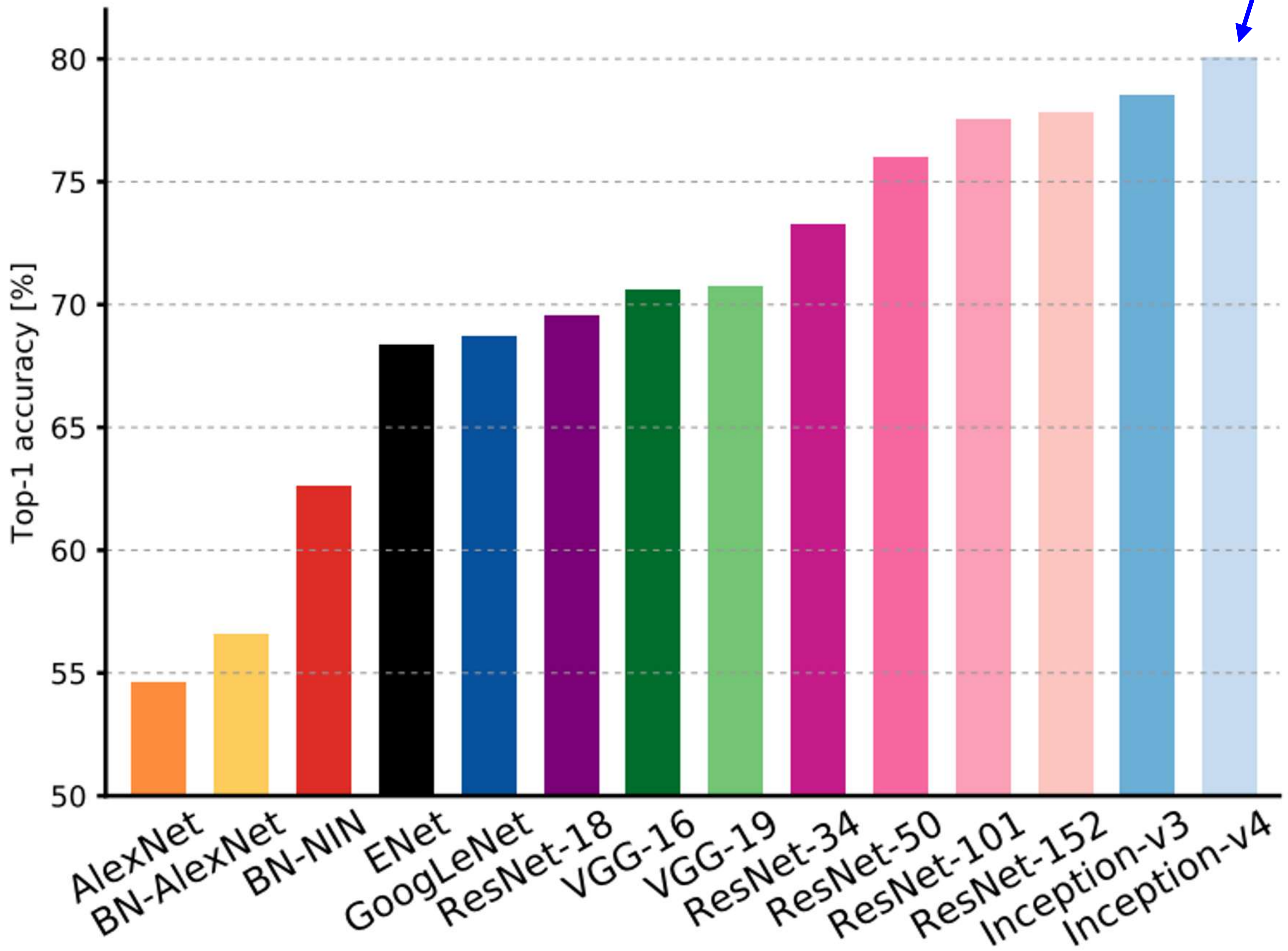


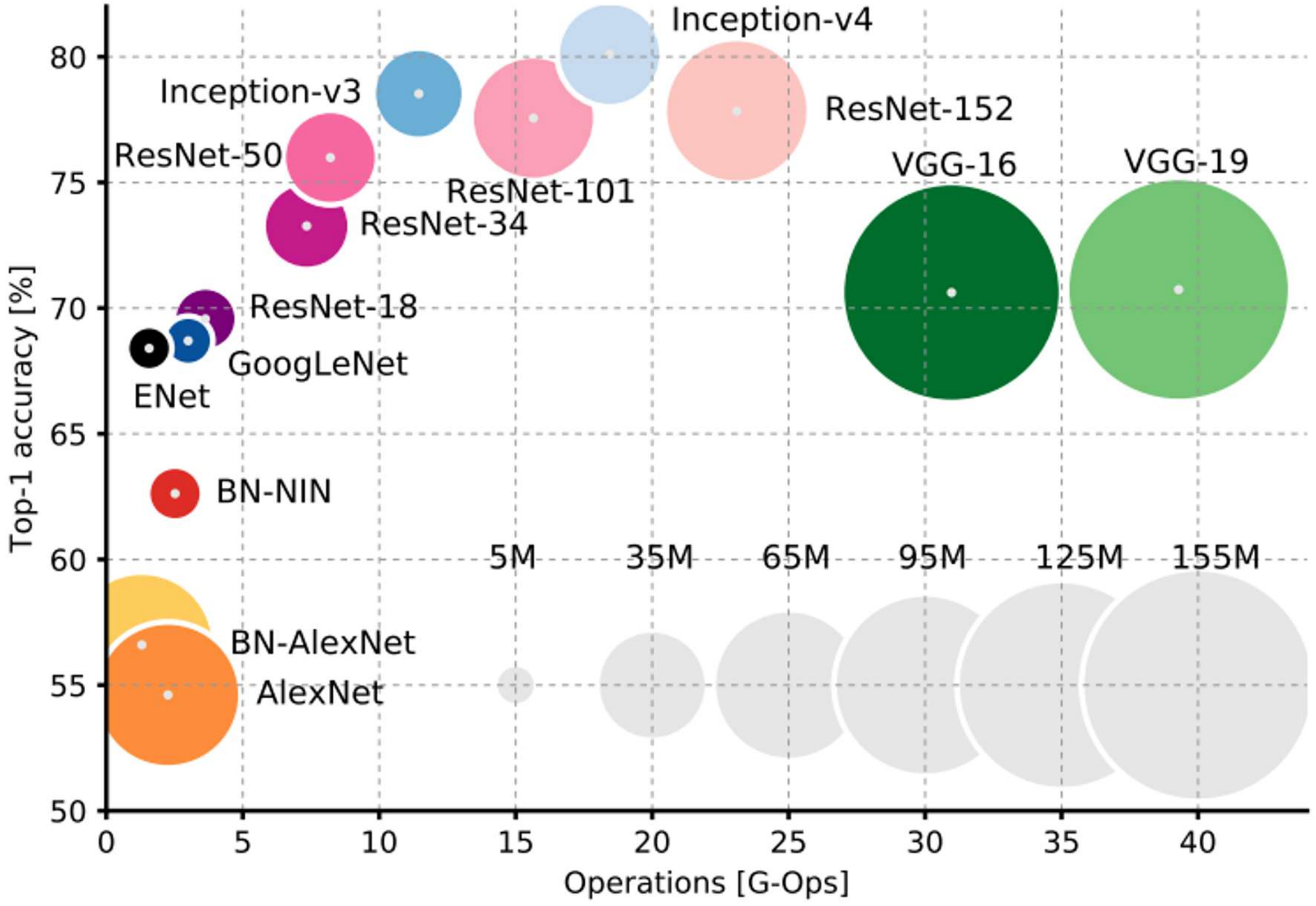
CNN architectures have continued to evolve!

Credit: Justin Johnson, U Michigan, Deep Learning for Computer Vision 2019

Comparing Complexity

Inception-v4: Resnet + Inception!





Canziani et al, "An analysis of deep neural network models for practical applications", 2017

Convolutional Neural Networks (CNNs)

- A bit of CNN history
- CNN Layers
  - Convolutions, Pooling, Fully Connected Layers, Normalization
- Training CNNs
  - Advanced Optimization
  - Weight initialization
  - Regularization: Dropout
  - Extended training data: Augmentation, Transfer Learning
  - Sanity-checking the learning process; Hyperparameter tuning strategies
- Famous CNN architectures
- CNNs for image segmentation; CNNs for object detection

Recap: Image Analysis Tasks

Image Classification

Assign one class label to an image

{dog, cat, truck, plane, …}

→ CAT

Image Classification

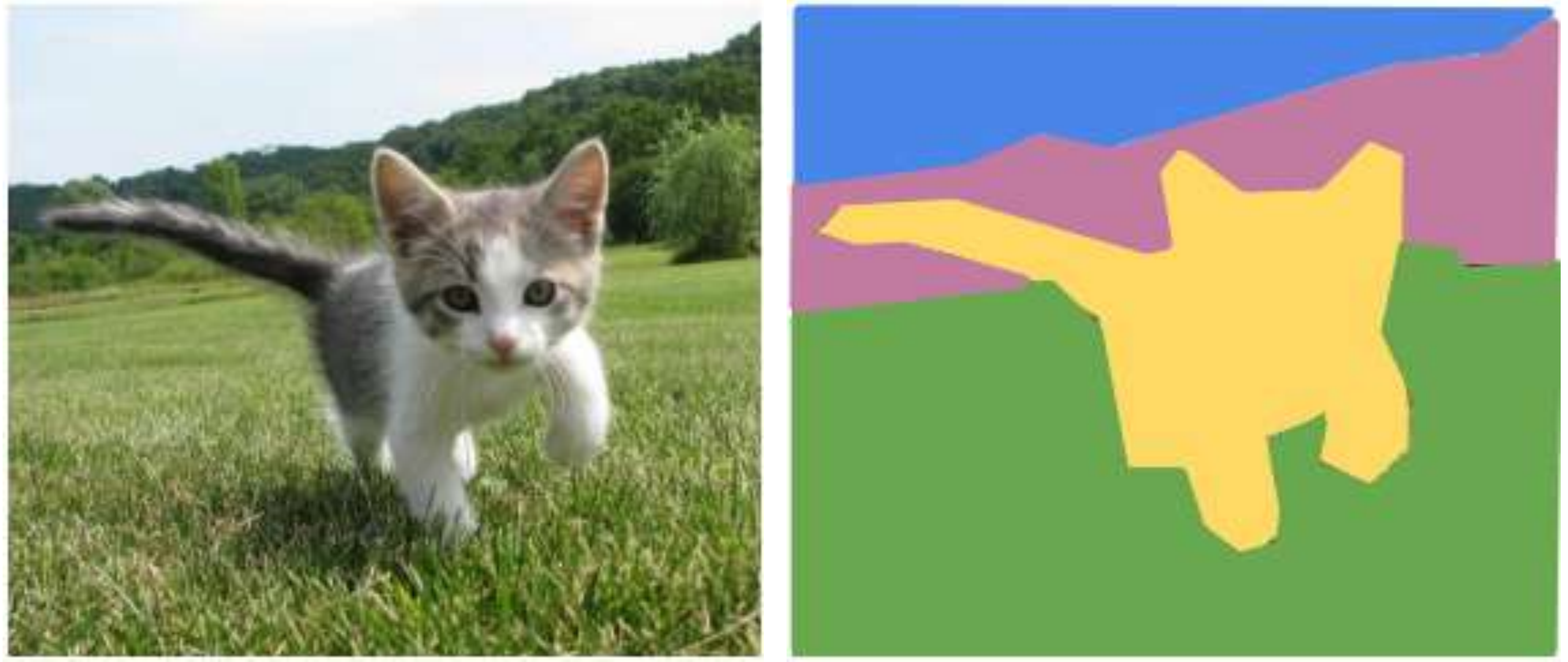
Assign one class label to an image
{dog, cat, truck, plane, …}

⟶ CAT

- No spatial extent
- What if localized information is sought in an image? → Pixel-wise tasks

Semantic Segmentation

Assign one class label to each pixel in an image
{dog, cat, truck, plane, ...}

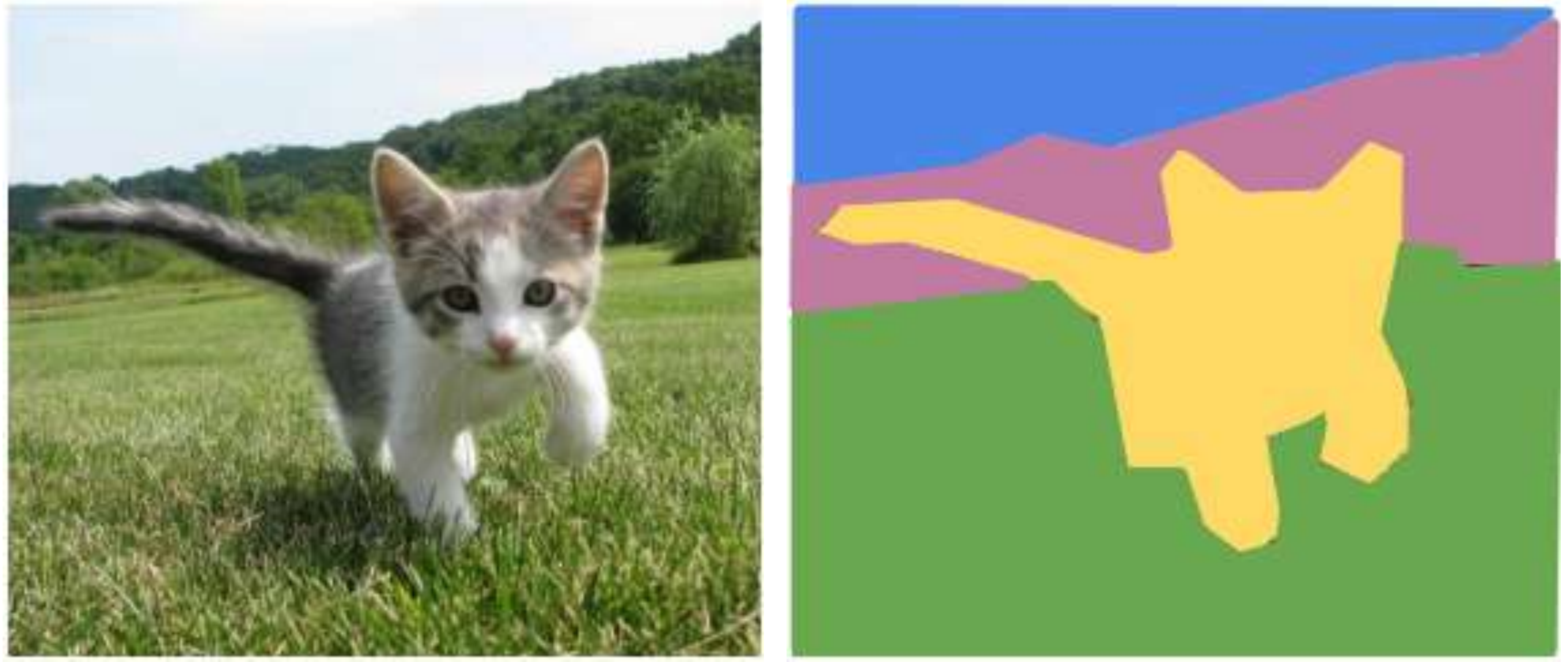


{cat, grass, tree, sky}

Semantic Segmentation

Assign one class label to each pixel in an image

{dog, cat, truck, plane, …}

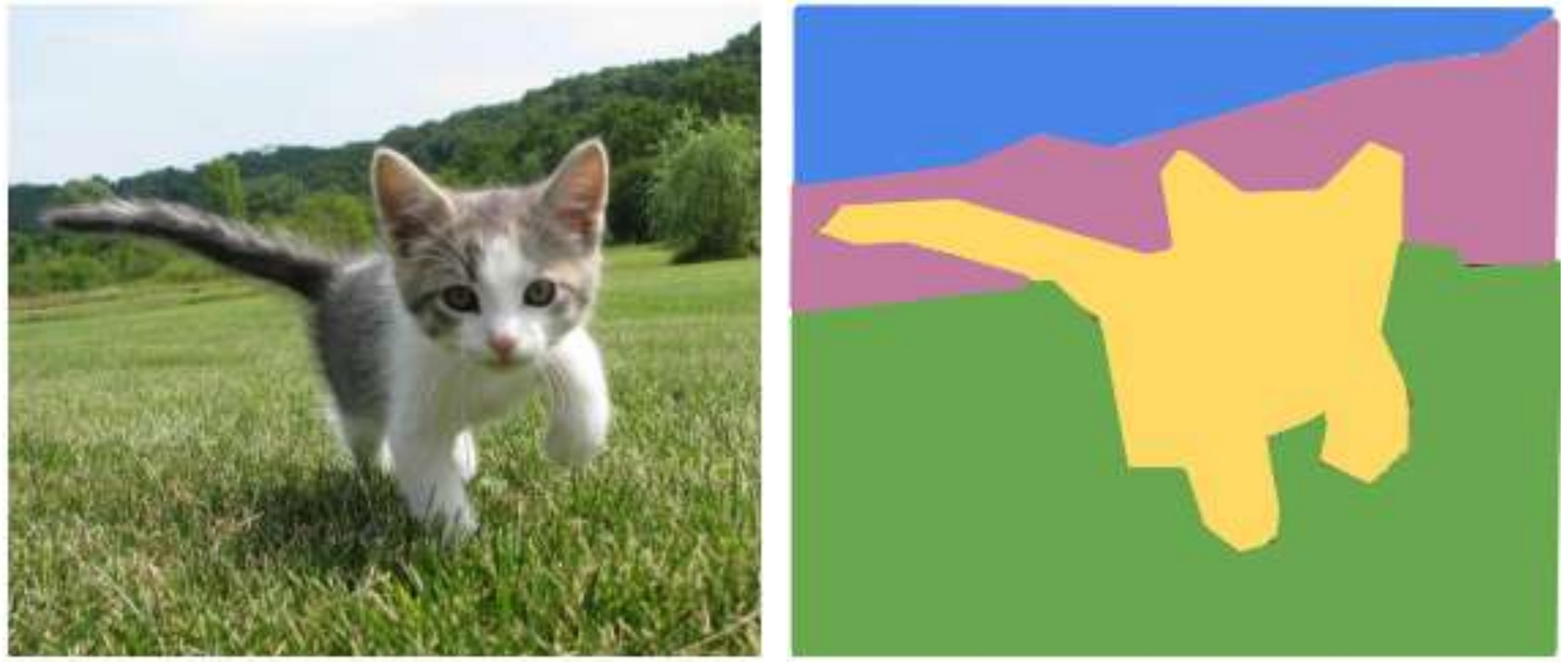


{cat, grass, tree, sky}

{cow, grass, tree, sky}

Semantic Segmentation

Assign one class label to each pixel in an image
{dog, cat, truck, plane, …}



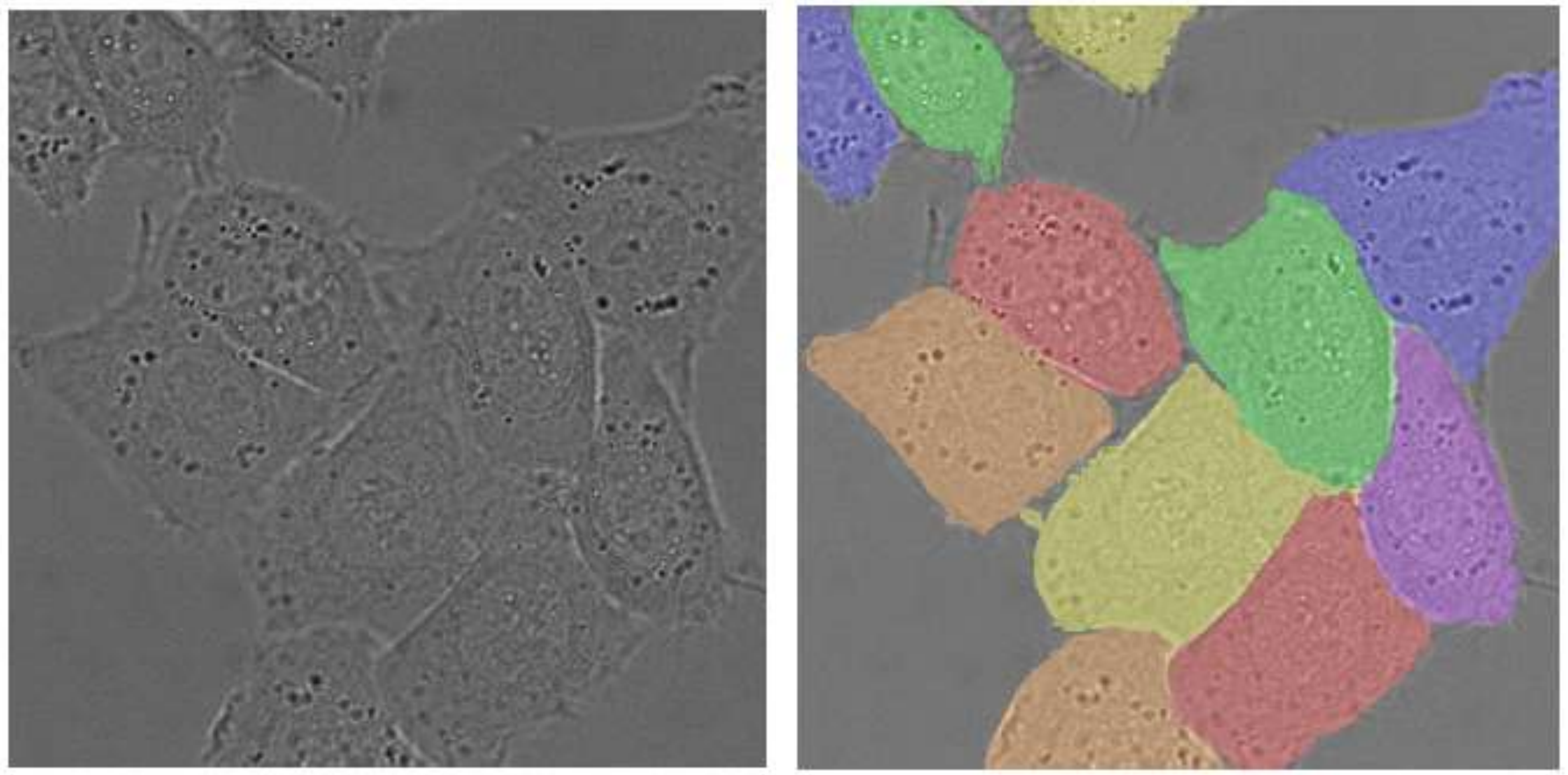
{cat, grass, tree, sky}

{cow, grass, tree, sky}

- Does not differentiate instances

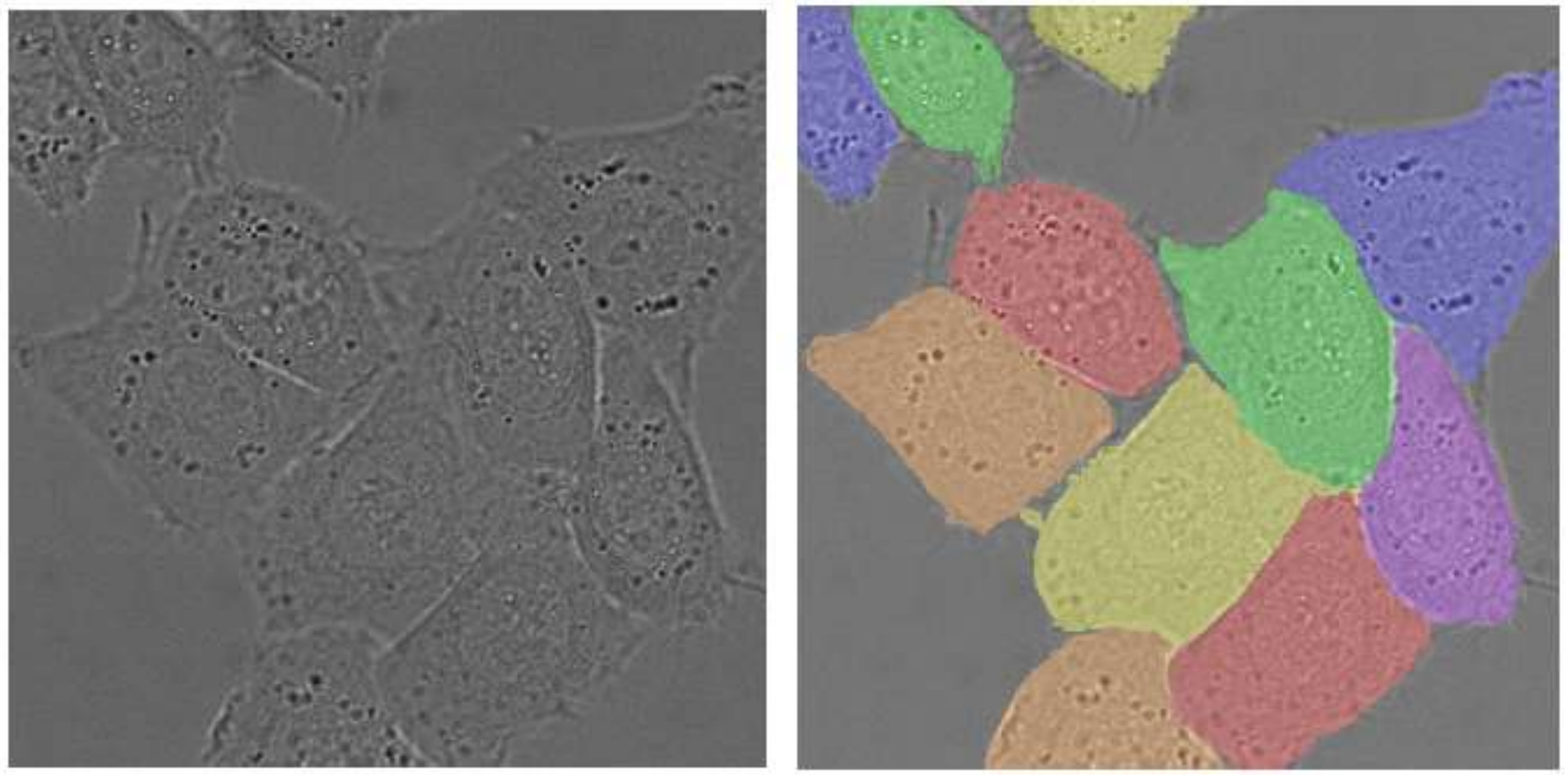
Instance Segmentation

Assign one instance label to each pixel in an image

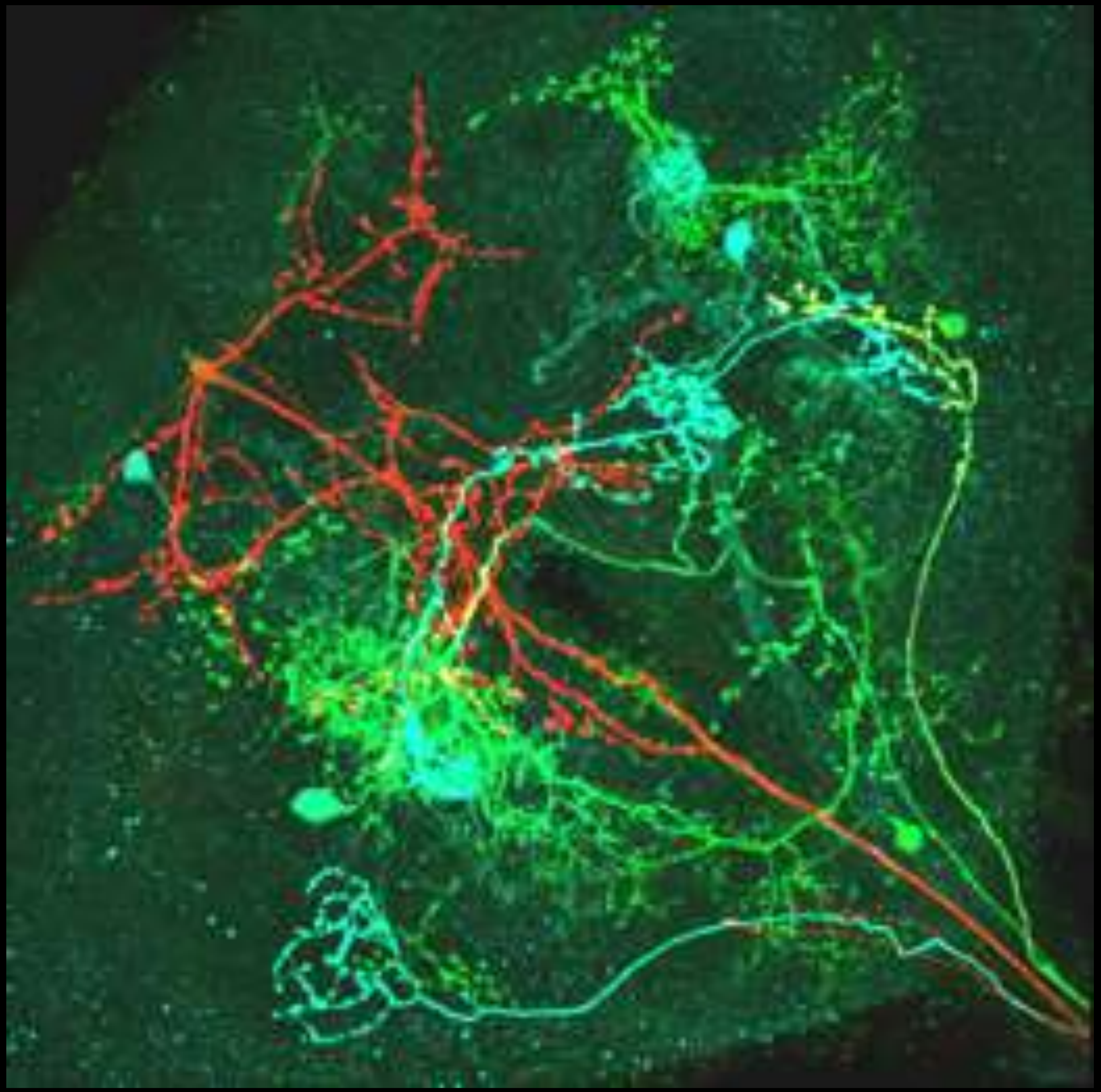


Instance Segmentation

Assign one instance label to each pixel in an image



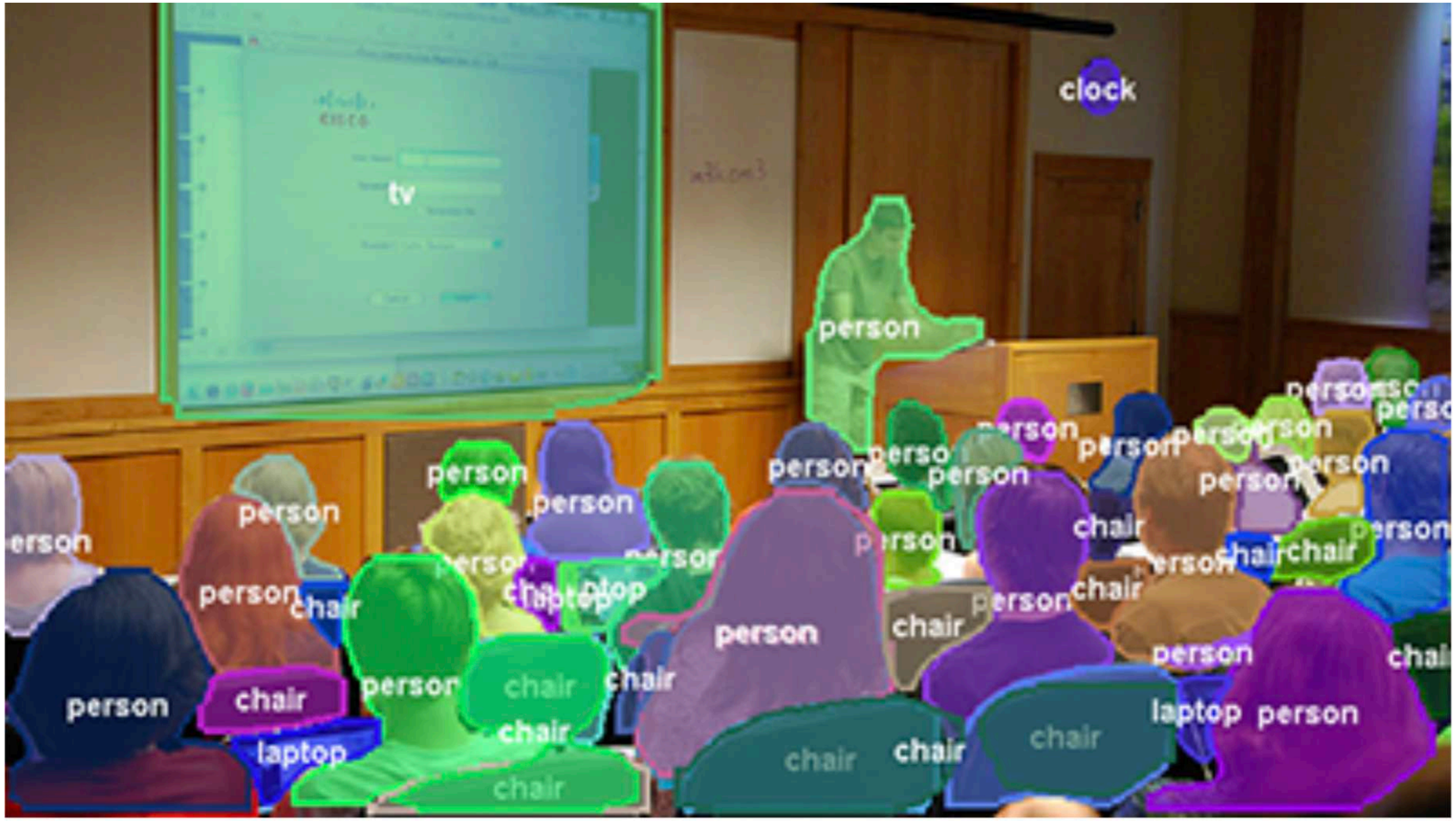
Instance Segmentation



FlyLight Project, HHMI Janelia

Semantic Instance Segmentation (aka Panoptic Segmentation)

Assign one class and instance label to each pixel in an image



CNNs for Semantic Segmentation

The Problem

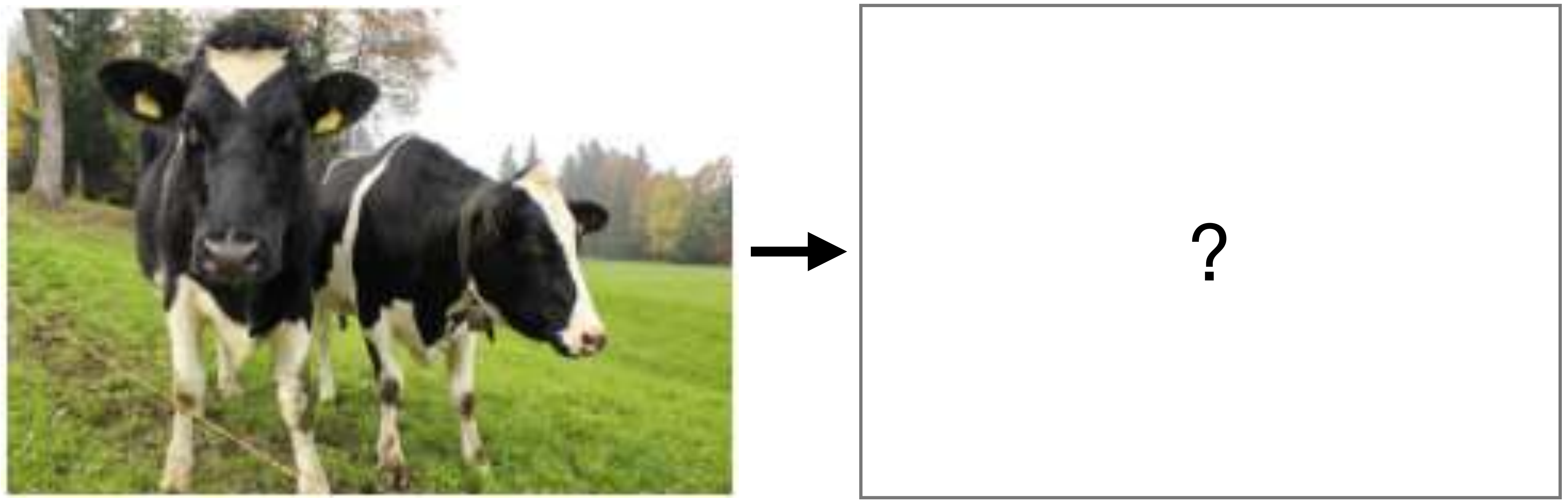
{grass, cat, tree, sky, ...}

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

The Problem

{grass, cat, tree, sky, ...}

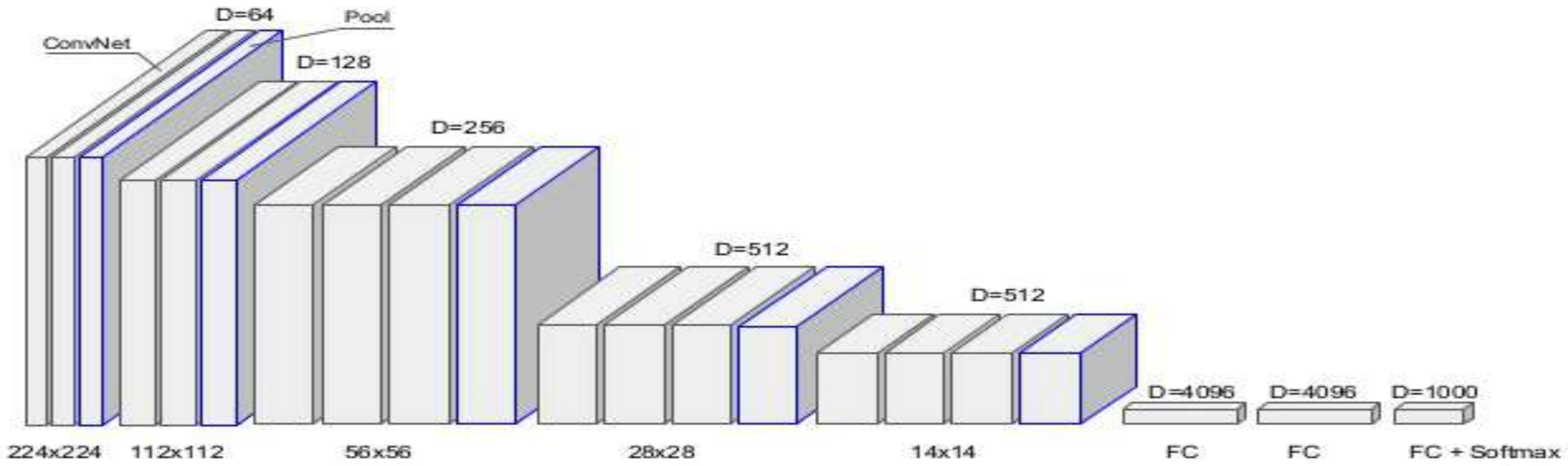
Paired training data:
for each training image, each pixel is labeled with a semantic category



At test time, classify each pixel of a new image

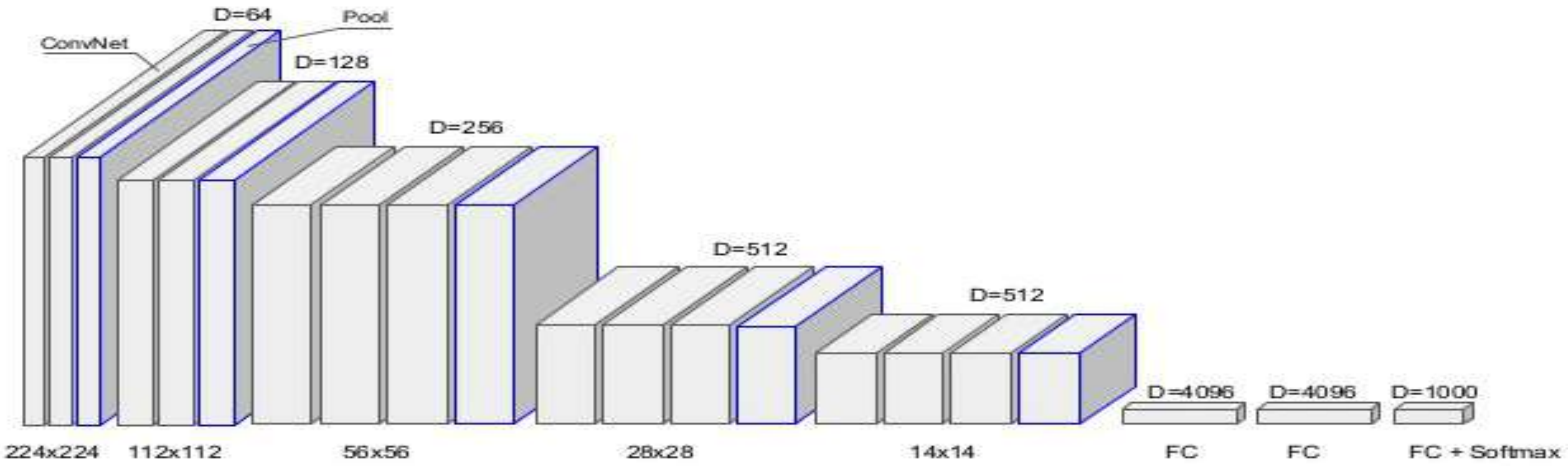
Recap ConvNets for Image Classification

Classical ConvNet topology - VGG19 (2013)



Recap ConvNets for Image Classification

Classical ConvNet topology - VGG19 (2013)



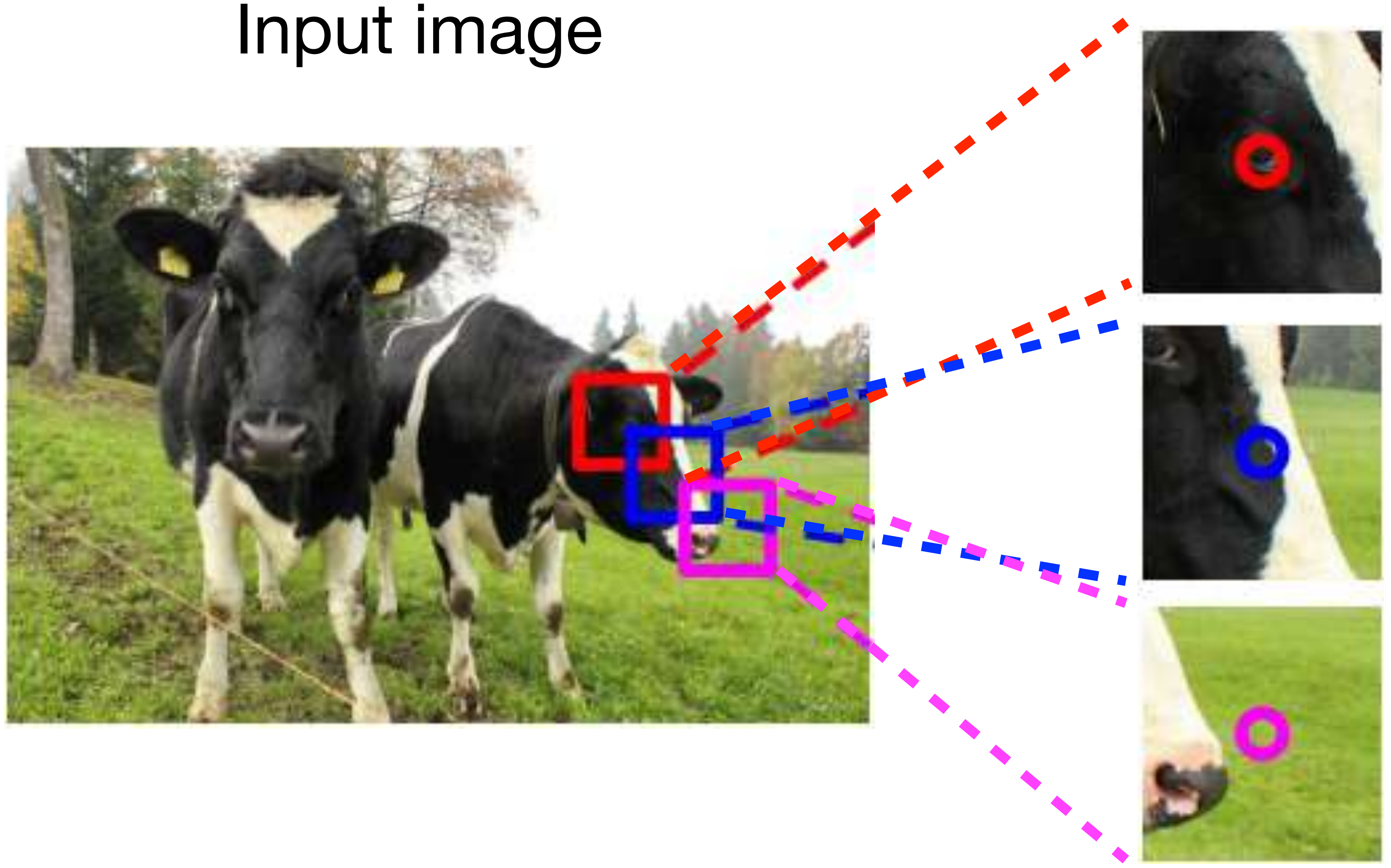
- Convolutions are translation equivariant
- Convolutional layers are relatively cheap (vs fully connected) in terms of memory and compute

Idea: Sliding Window

Idea: Sliding Window

Extract patch

Input image



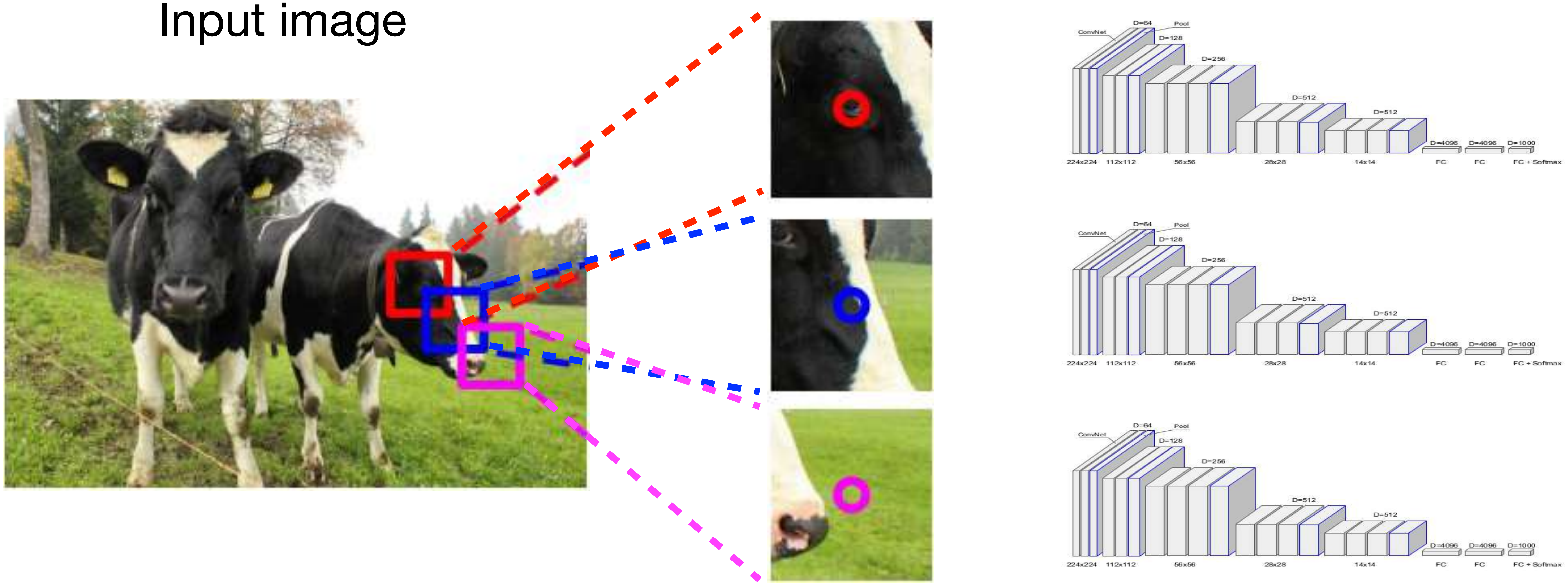
Credits to Fei-Fei Li, Jiajun Wu and Ruohan Gao (Stanford, Deep Learning for Computer Vision, 2022, cs231n)

Idea: Sliding Window

Input image

Extract patch

Classify center pixel with ConvNet

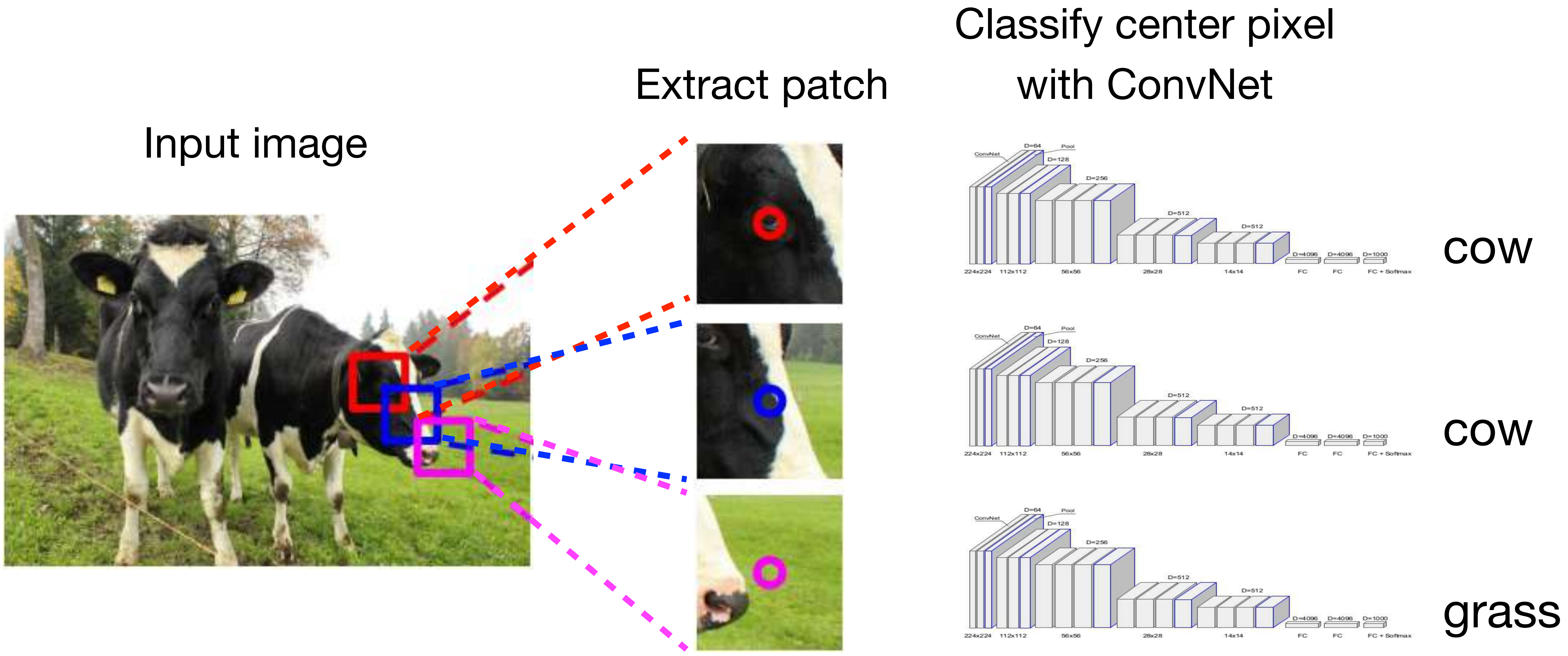


cow

cow

grass

Idea: Sliding Window

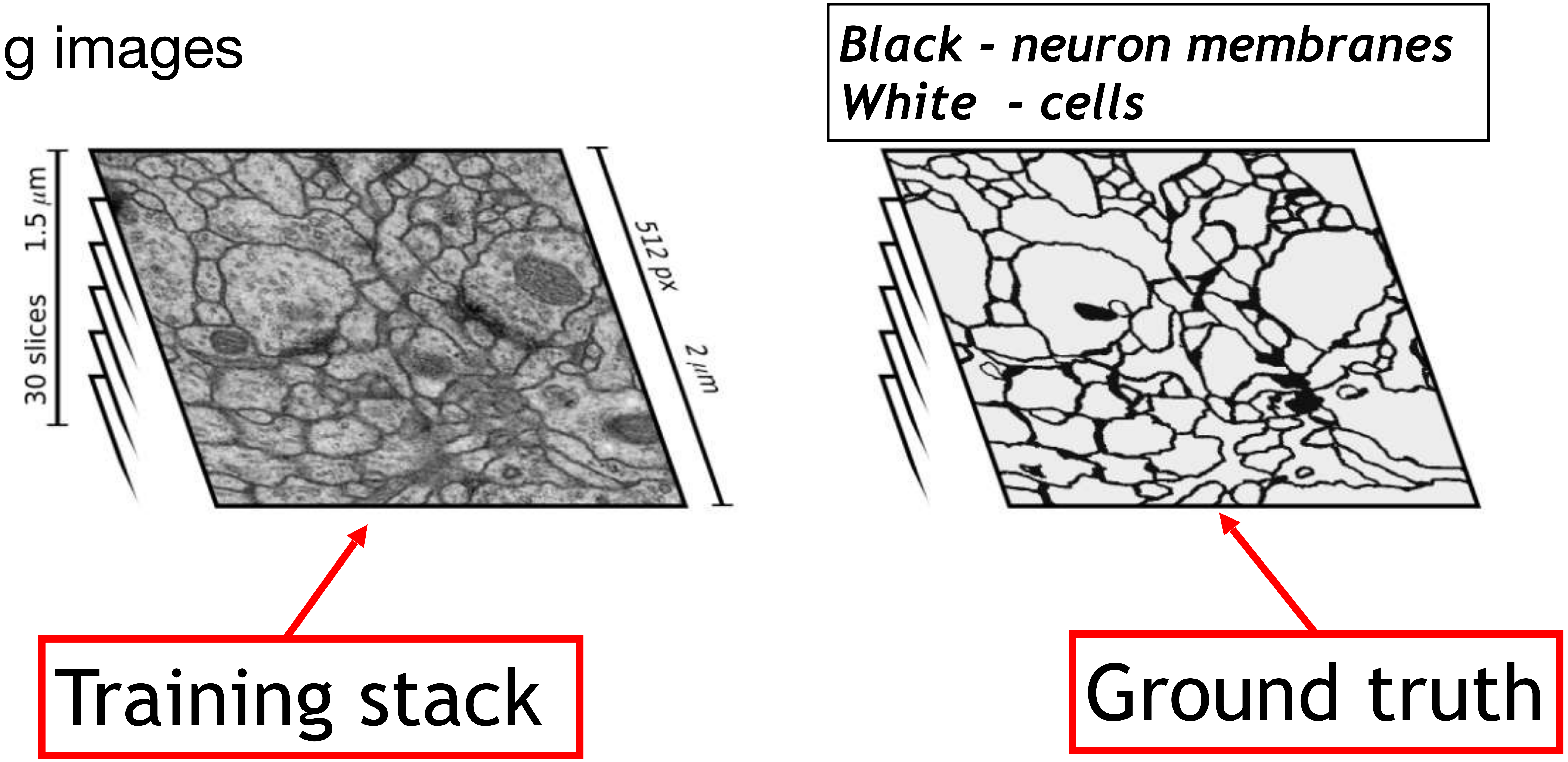


Ciresan et al, Deep neural networks segment neuronal membranes in electron microscopy, NIPS 2012
Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013
Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014

Idea: Sliding Window

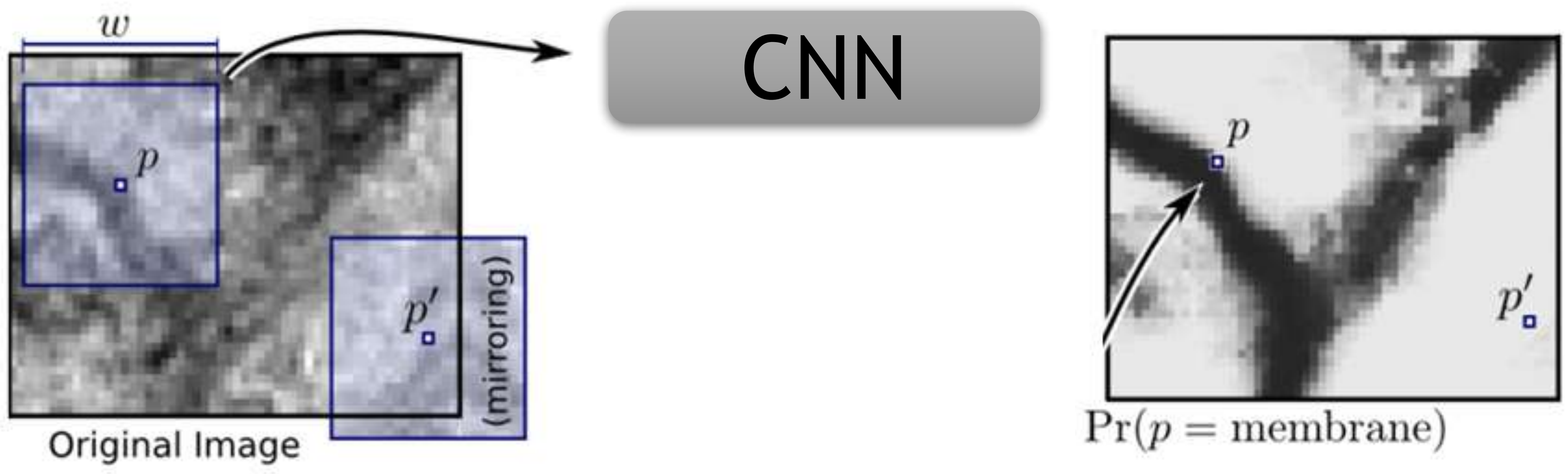
ISBI 2012 EM Segmentation Challenge

- Stacks of Electron microscopy (EM) images
- 30 training images



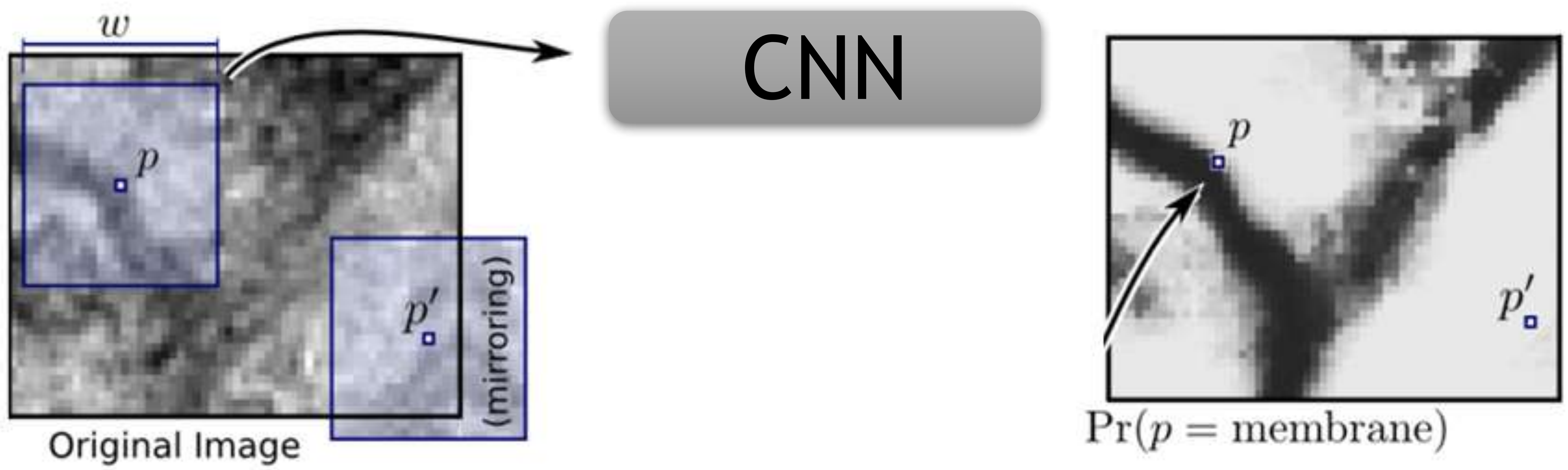
Idea: Sliding Window

ISBI 2012 Winner: Ciresan et al.



Idea: Sliding Window

ISBI 2012 Winner: Ciresan et al.

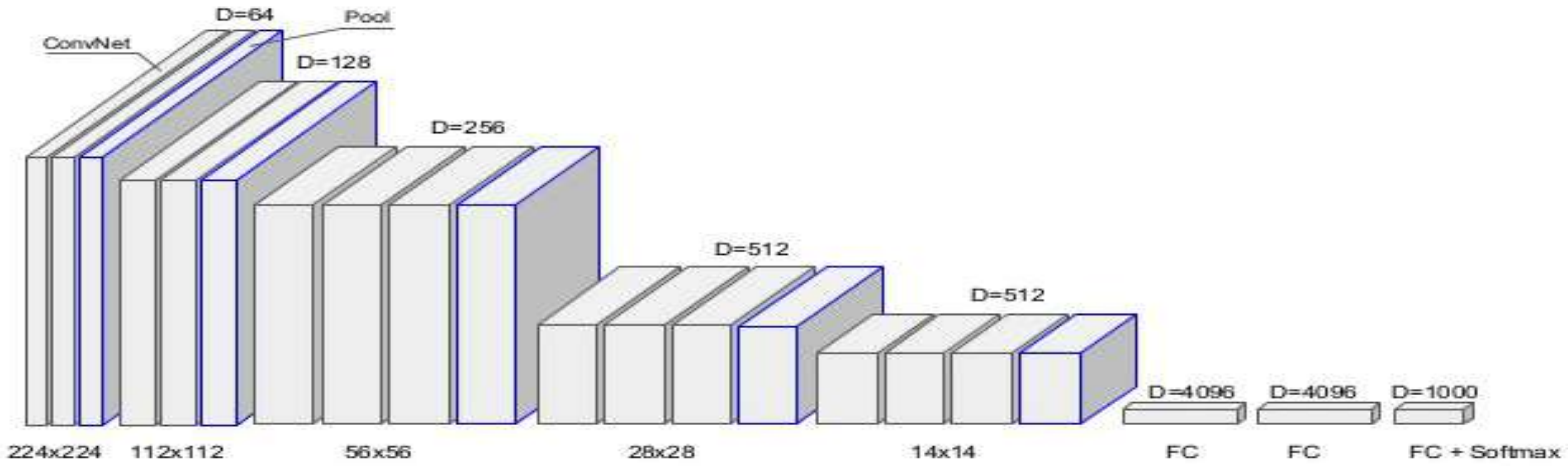


Problem: Very inefficient!

- Network must be run separately for each patch
- Not reusing shared operations between overlapping patches

Idea: Fully Convolutional Network

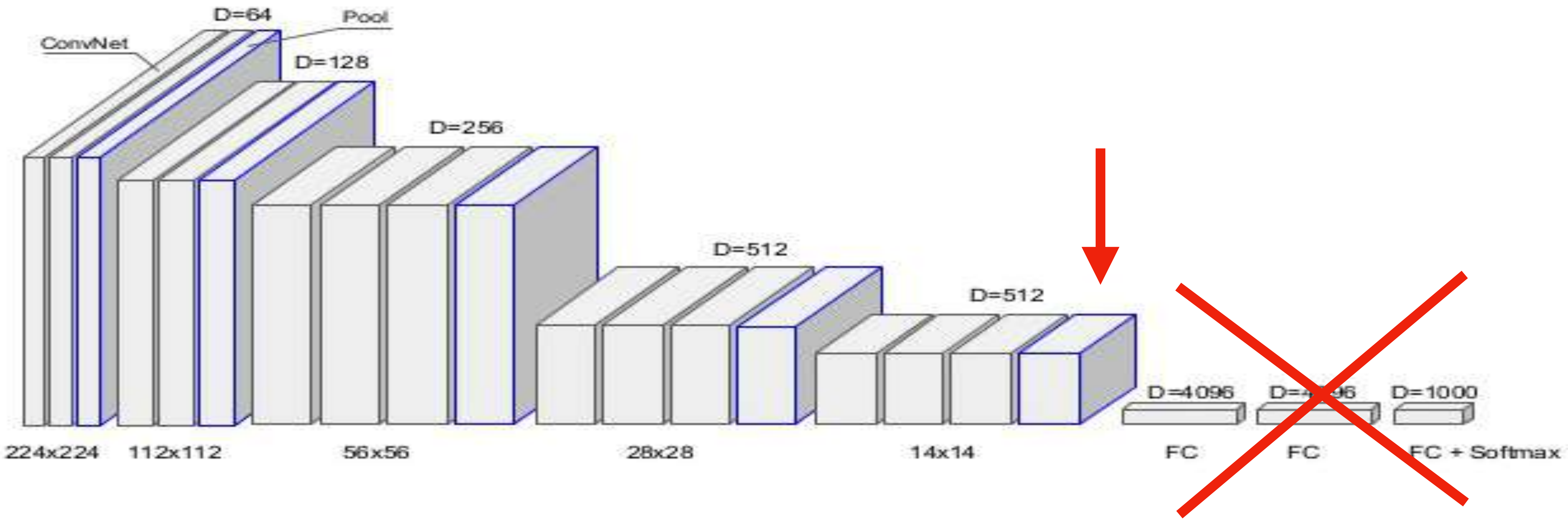
Getting rid of redundant computations:



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

Idea: Fully Convolutional Network

Getting rid of redundant computations:

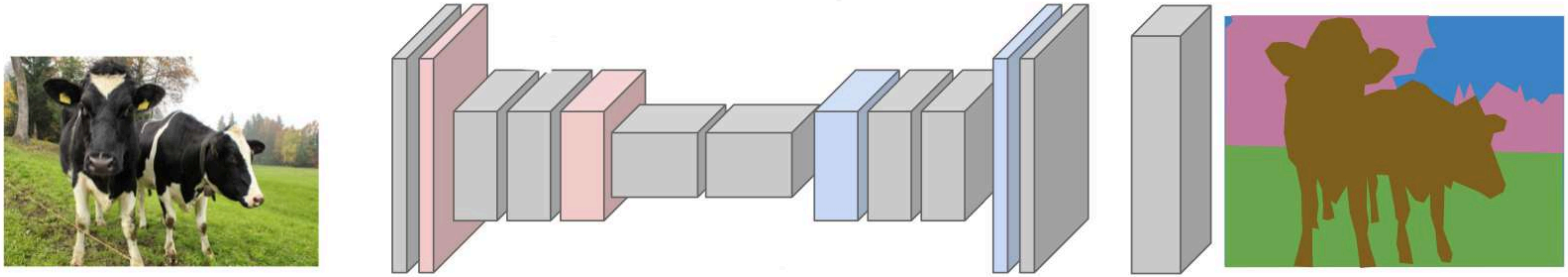


Long, Shelhamer and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015
Noh et al., “Learning Deconvolution Network for Semantic Segmentation”, ICCV 2015

Idea: Fully Convolutional Network

Getting rid of redundant computations:

Design network with downsampling and upsampling inside the network



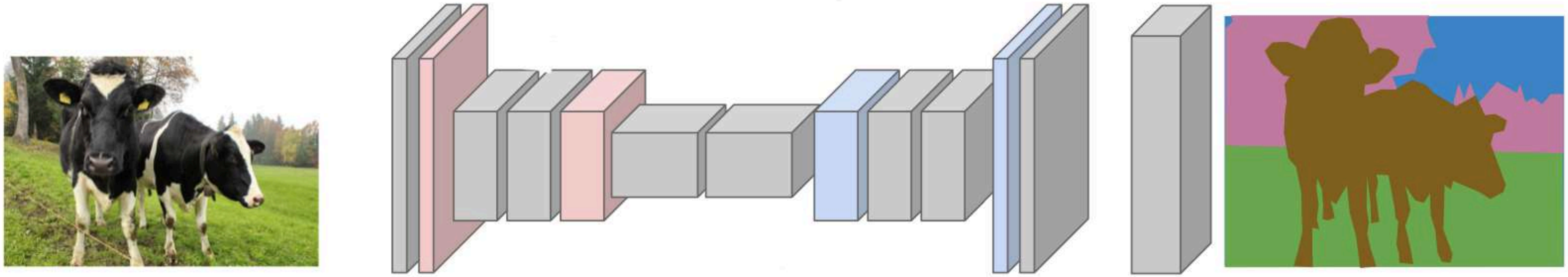
Long, Shelhamer and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015

Noh et al., “Learning Deconvolution Network for Semantic Segmentation”, ICCV 2015

Idea: Fully Convolutional Network

Getting rid of redundant computations:

Design network with downsampling and upsampling inside the network



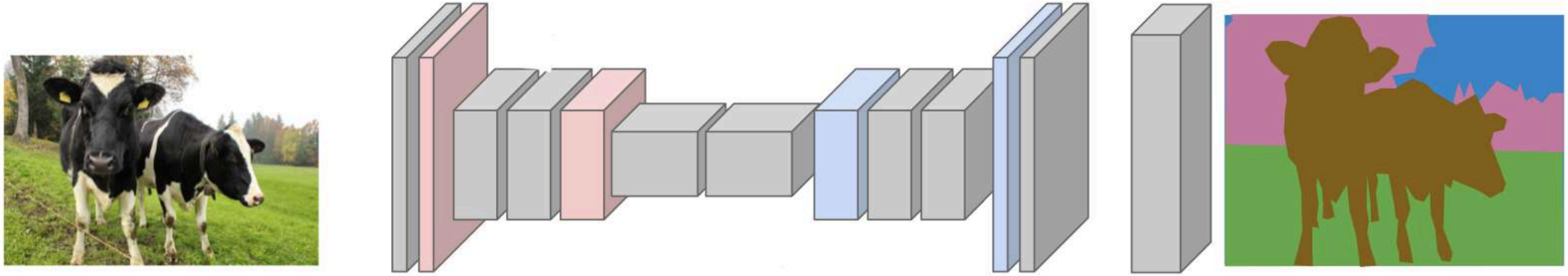
Downsampling:
Pooling, strided
convolution

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

Idea: Fully Convolutional Network

Getting rid of redundant computations:

Design network with downsampling and upsampling inside the network



Downsampling:
Pooling, strided convolution

Upsampling:
???

Long, Shelhamer and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015

Noh et al., "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Upsampling

Nearest Neighbor

1	2
3	4

Input: 2 x 2

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

Output: 4 x 4

Upsampling

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

Bilinear Interpolation

1	2
3	4

1	1.25	1.75	2
1.5	1.75	2.25	2.5
2.5	2.75	3.25	3.5
3	3.25	3.75	4

Input: 2 x 2

Output: 4 x 4

Credits to Fei-Fei Li, Jiajun Wu and Ruohan Gao (Stanford, Deep Learning for Computer Vision, 2022, cs231n)

Upsampling

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

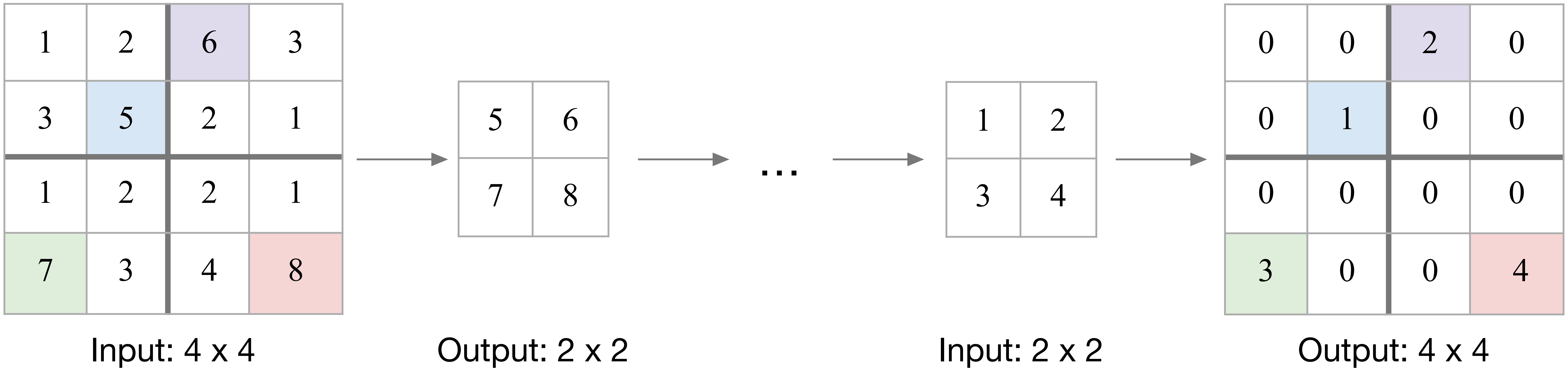
Upsampling

Max Pooling

Remember which element was max

Max Unpooling

Use position from pooling layer



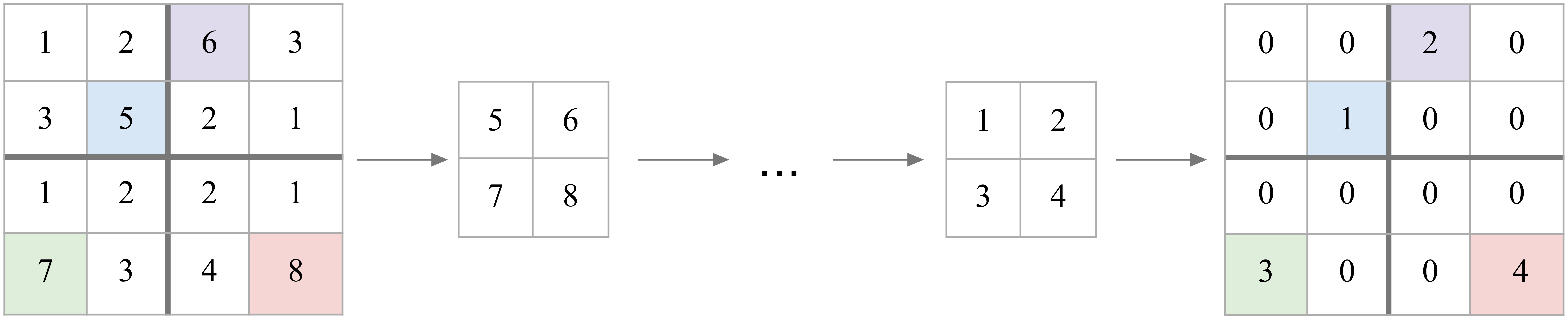
Upsampling

Max Pooling

Remember which element was max

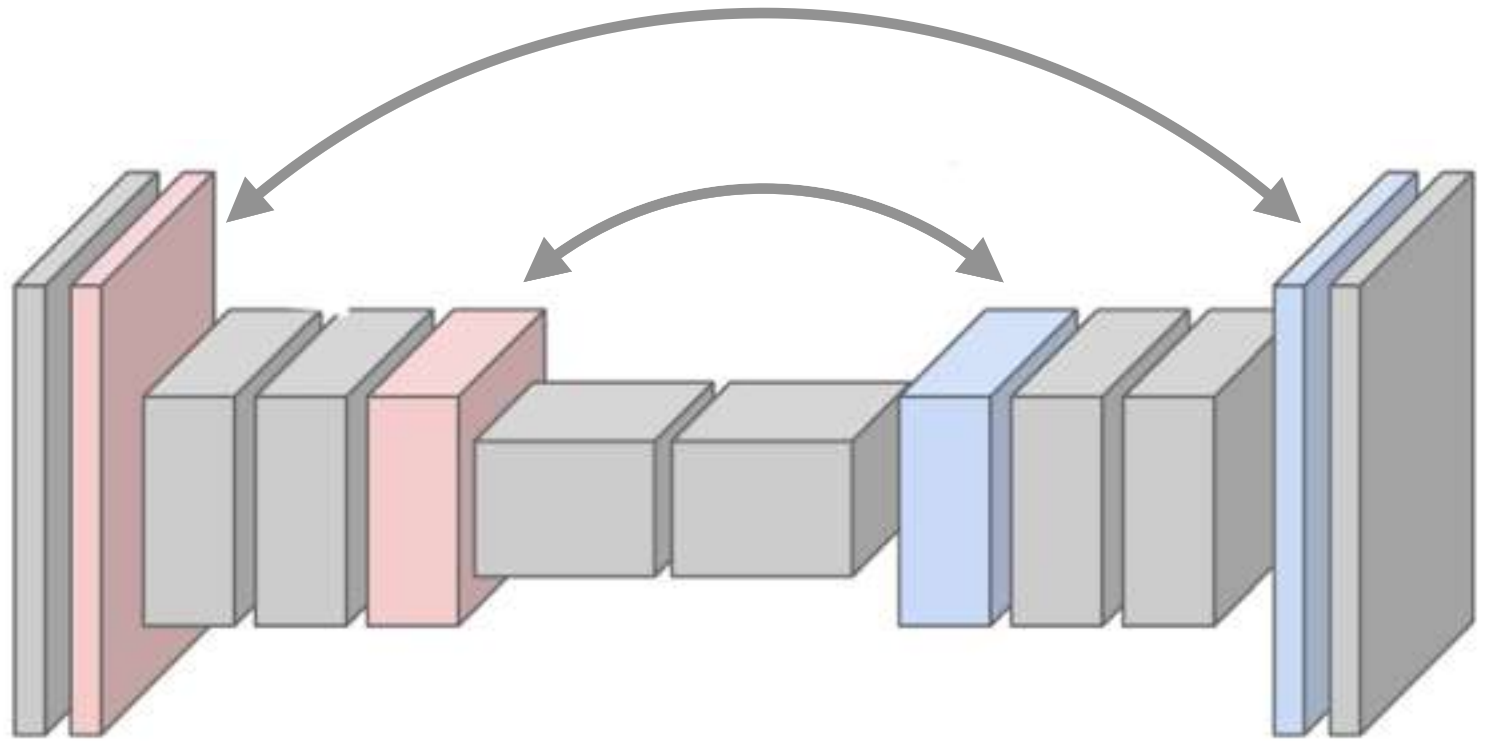
Max Unpooling

Use position from pooling layer



Input: 4 x 4 | Output: 2 x 2 | Input: 2 x 2 | Output: 4 x 4

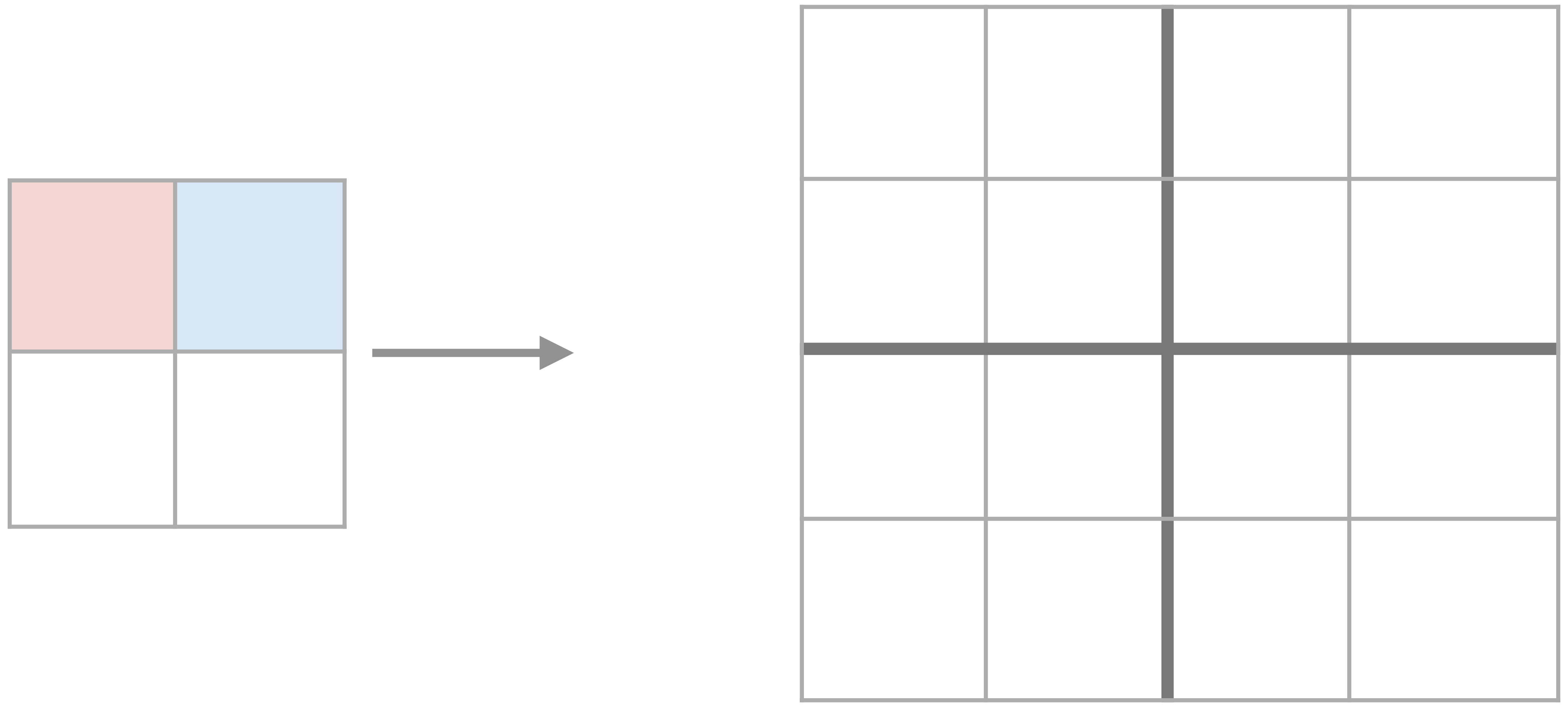
Corresponding pairs of
downsampling and upsampling layers



Credits to Fei-Fei Li, Jiajun Wu and Ruohan Gao (Stanford, Deep Learning for Computer Vision, 2022, cs231n)

Learnable Upsampling: Transposed Convolution

3 x 3 transposed convolution, stride 2 pad 1

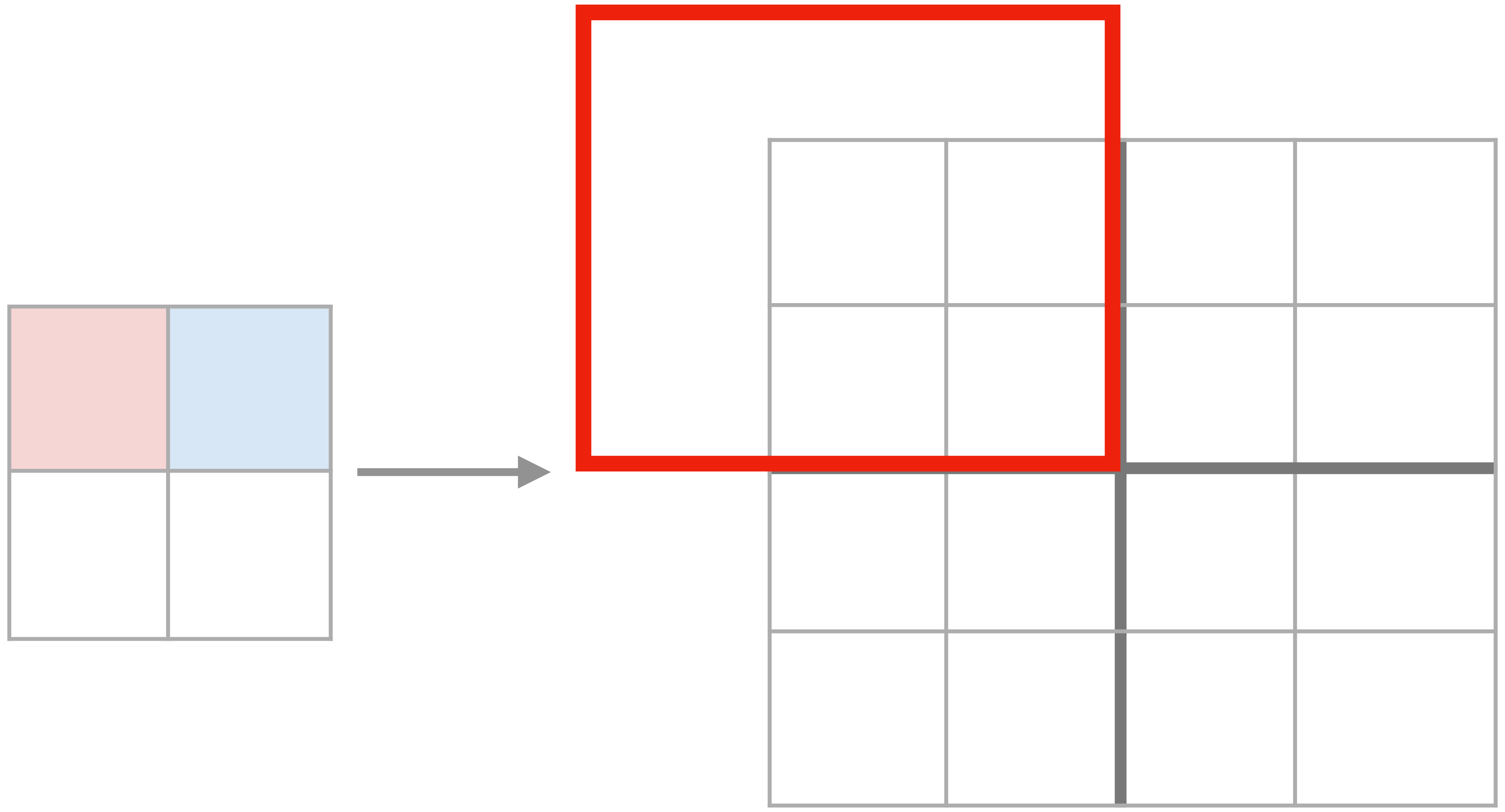


Input: 2 x 2

Output: 4 x 4

Learnable Upsampling: Transposed Convolution

3 x 3 transposed convolution, stride 2 pad 1

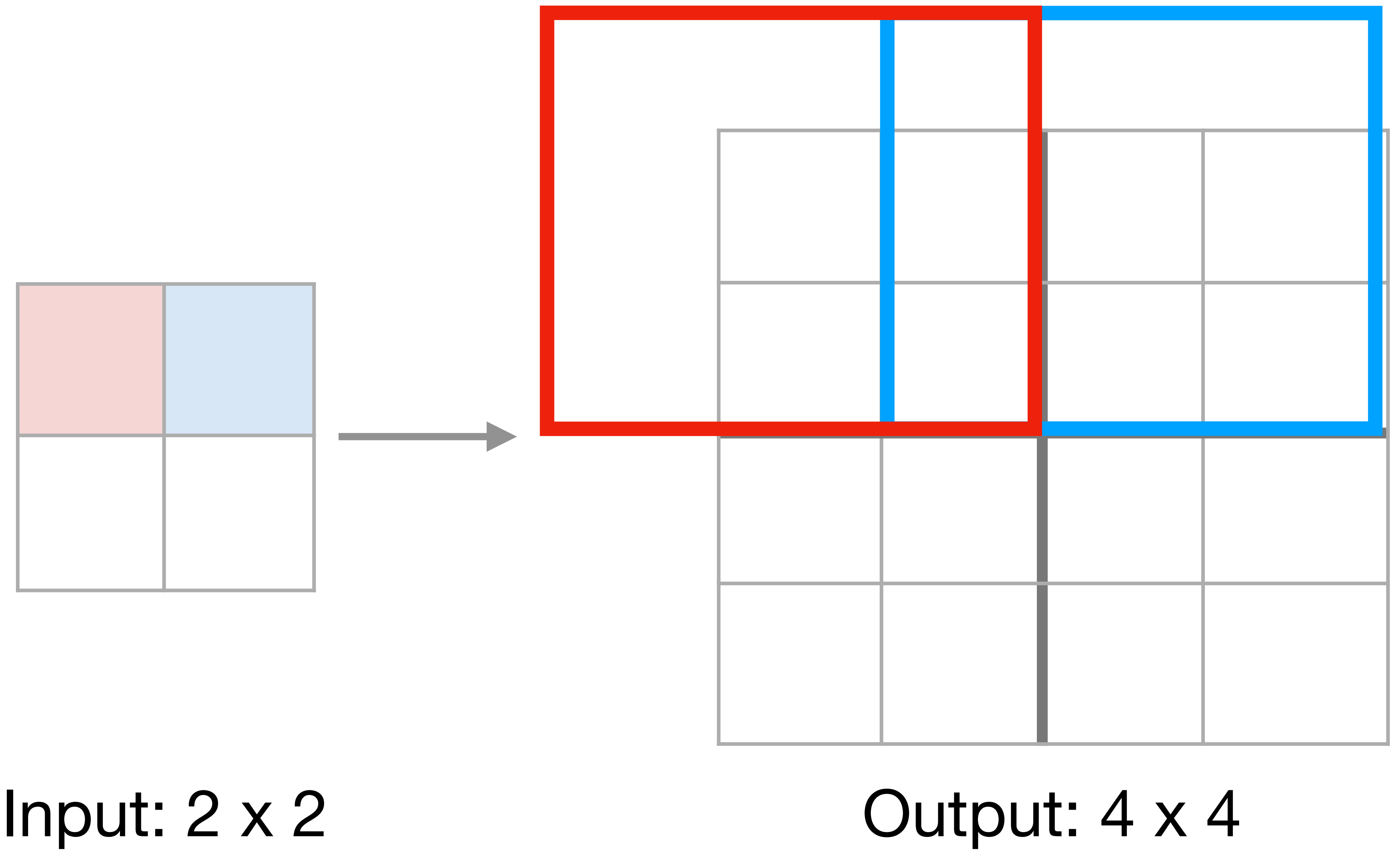


Input: 2 x 2

Output: 4 x 4

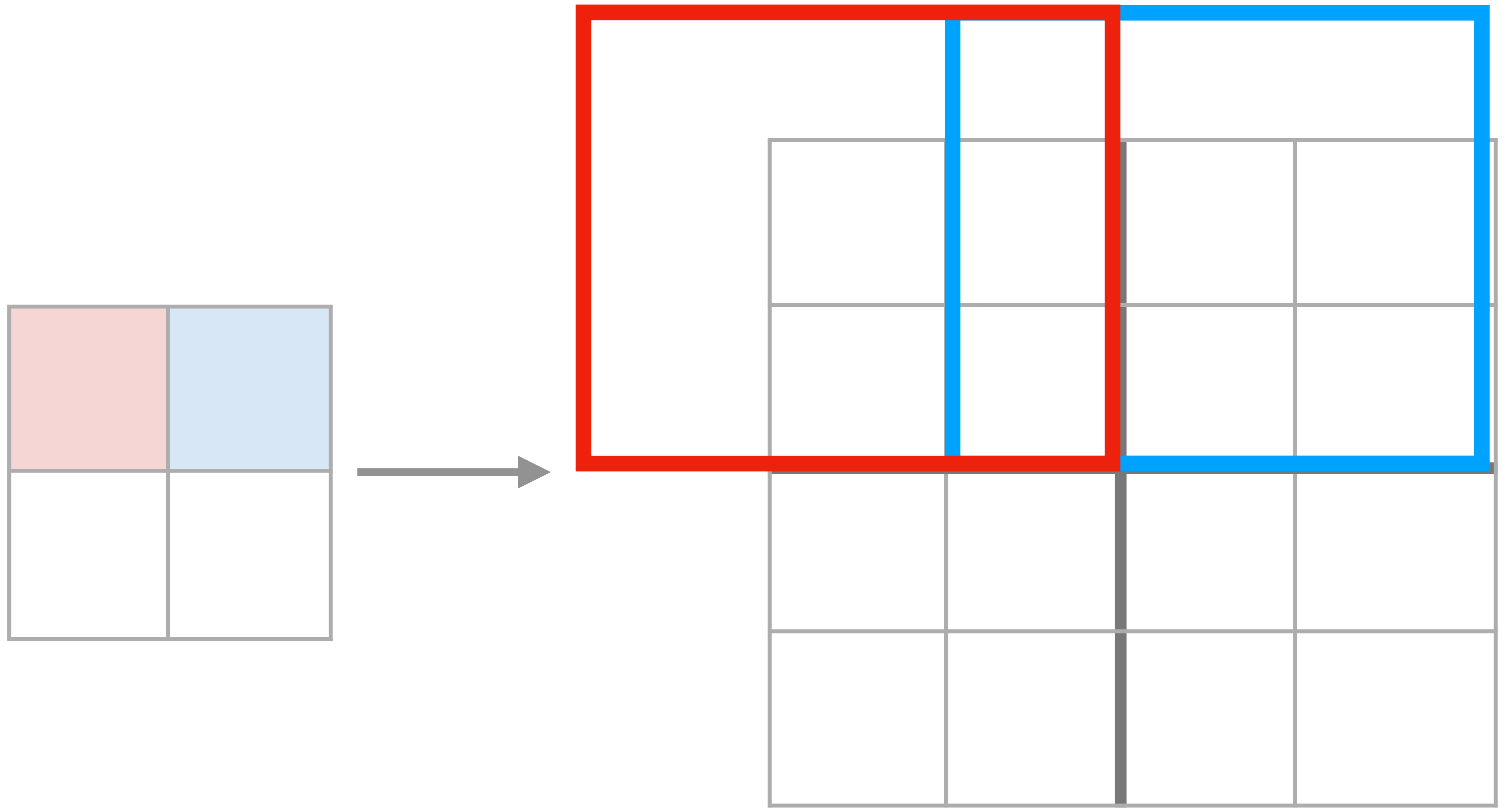
Learnable Upsampling: Transposed Convolution

3 x 3 transposed convolution, stride 2 pad 1



Learnable Upsampling: Transposed Convolution

3 x 3 transposed convolution, stride 2 pad 1



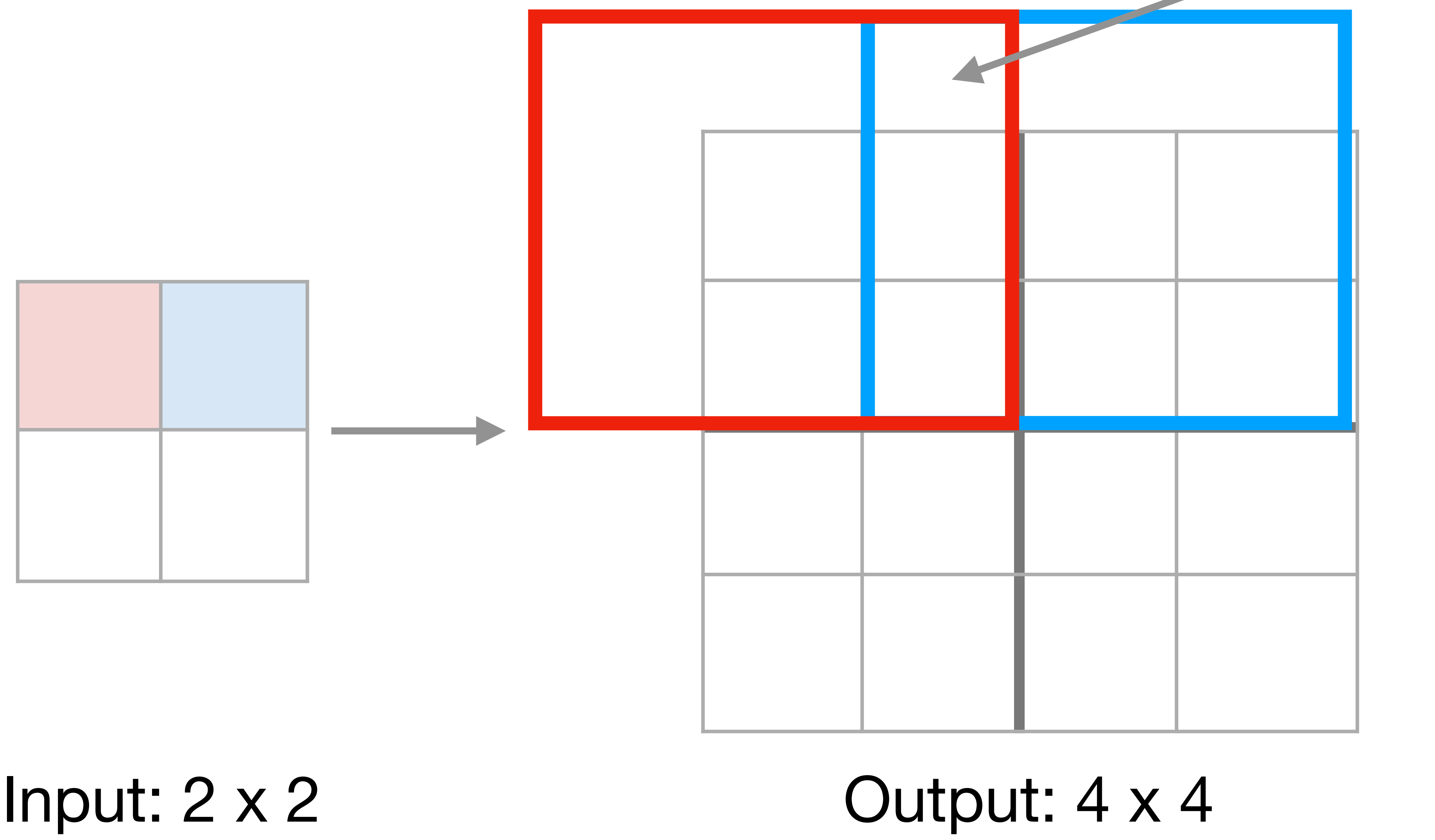
Input: 2 x 2

Output: 4 x 4

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

Learnable Upsampling: Transposed Convolution

3 x 3 transposed convolution, stride 2 pad 1



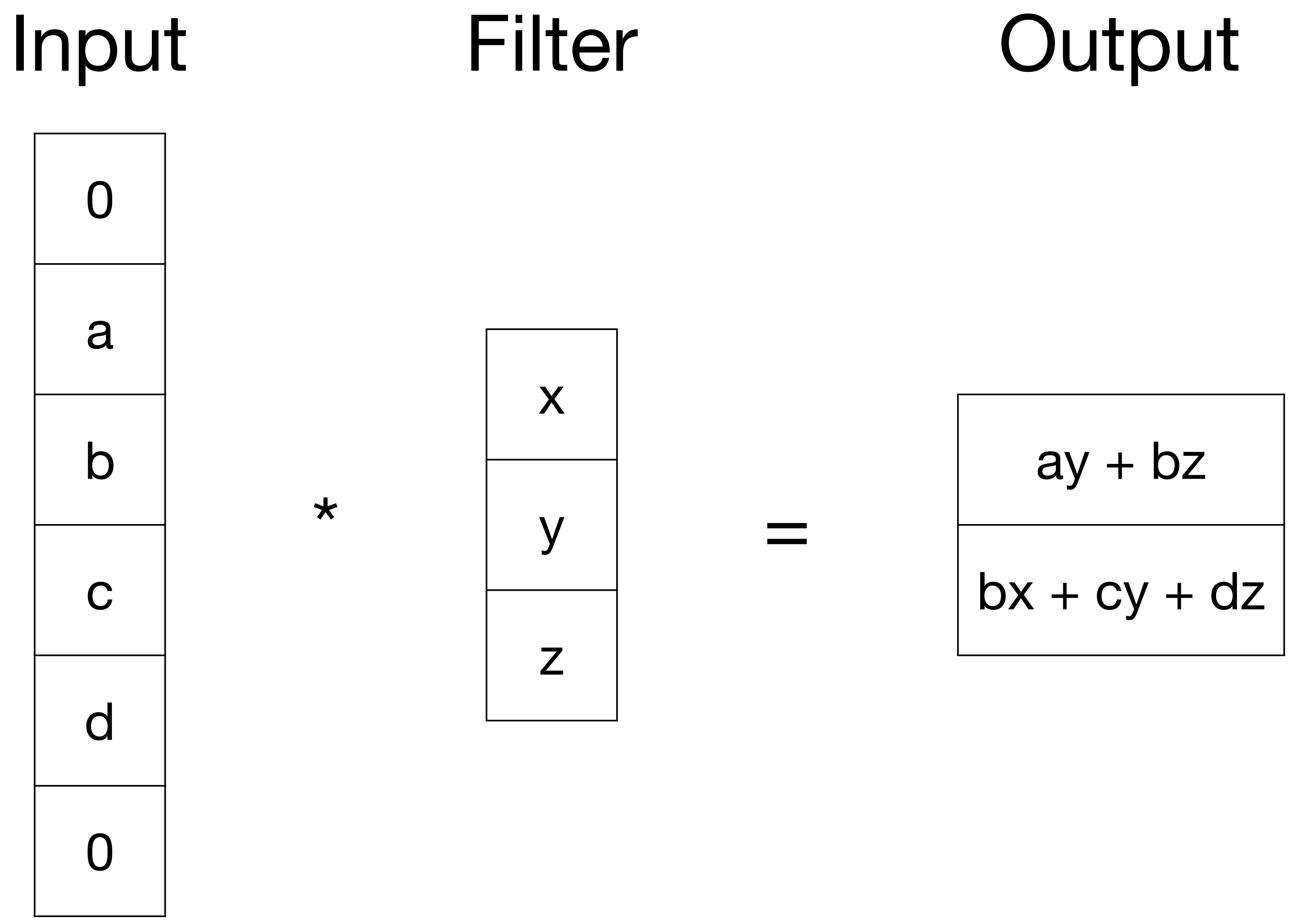
Sum where output overlaps

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

Learnable Upsampling: 1D Example

Convolution

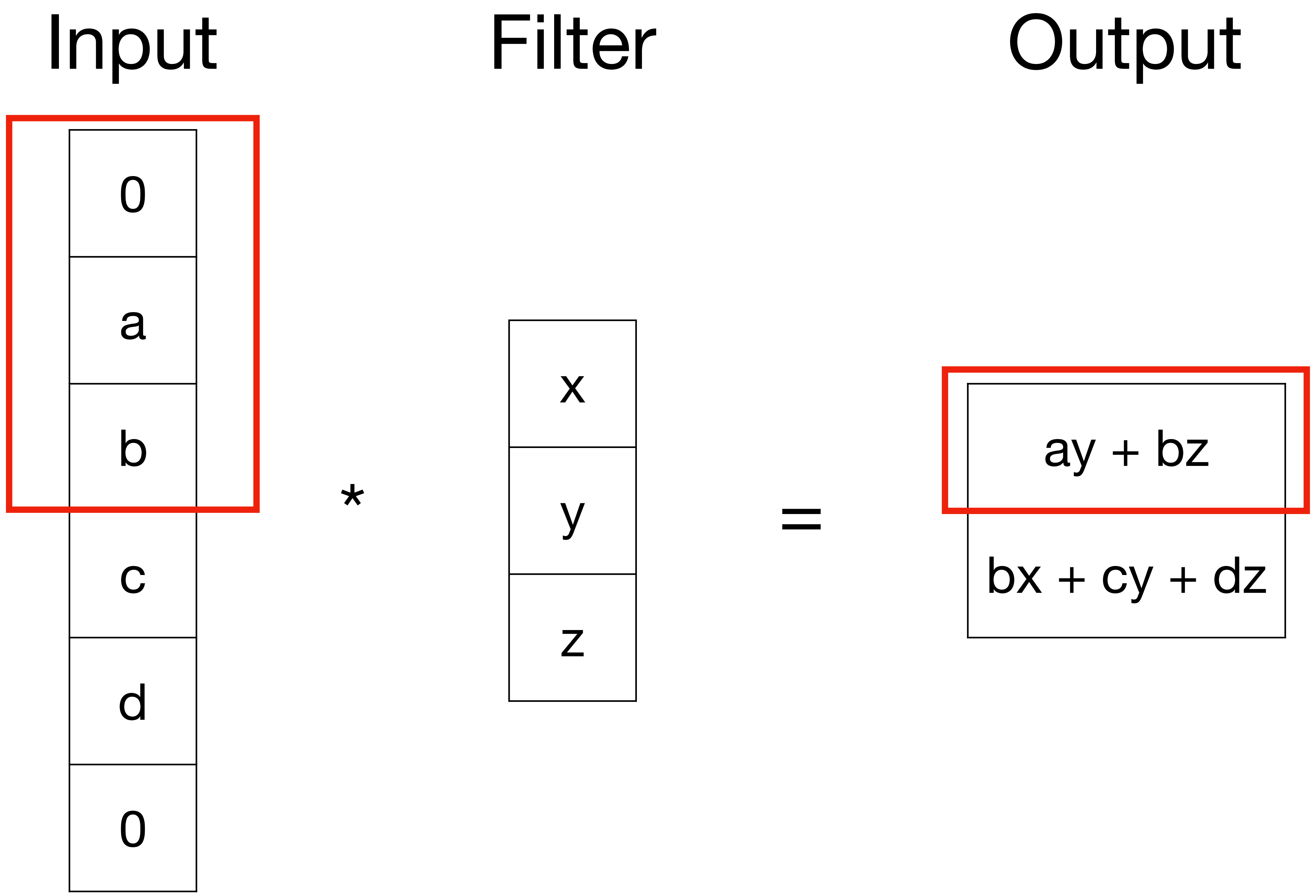
1D conv: kernel size = 3, stride = 2, padding=1



Learnable Upsampling: 1D Example

Convolution

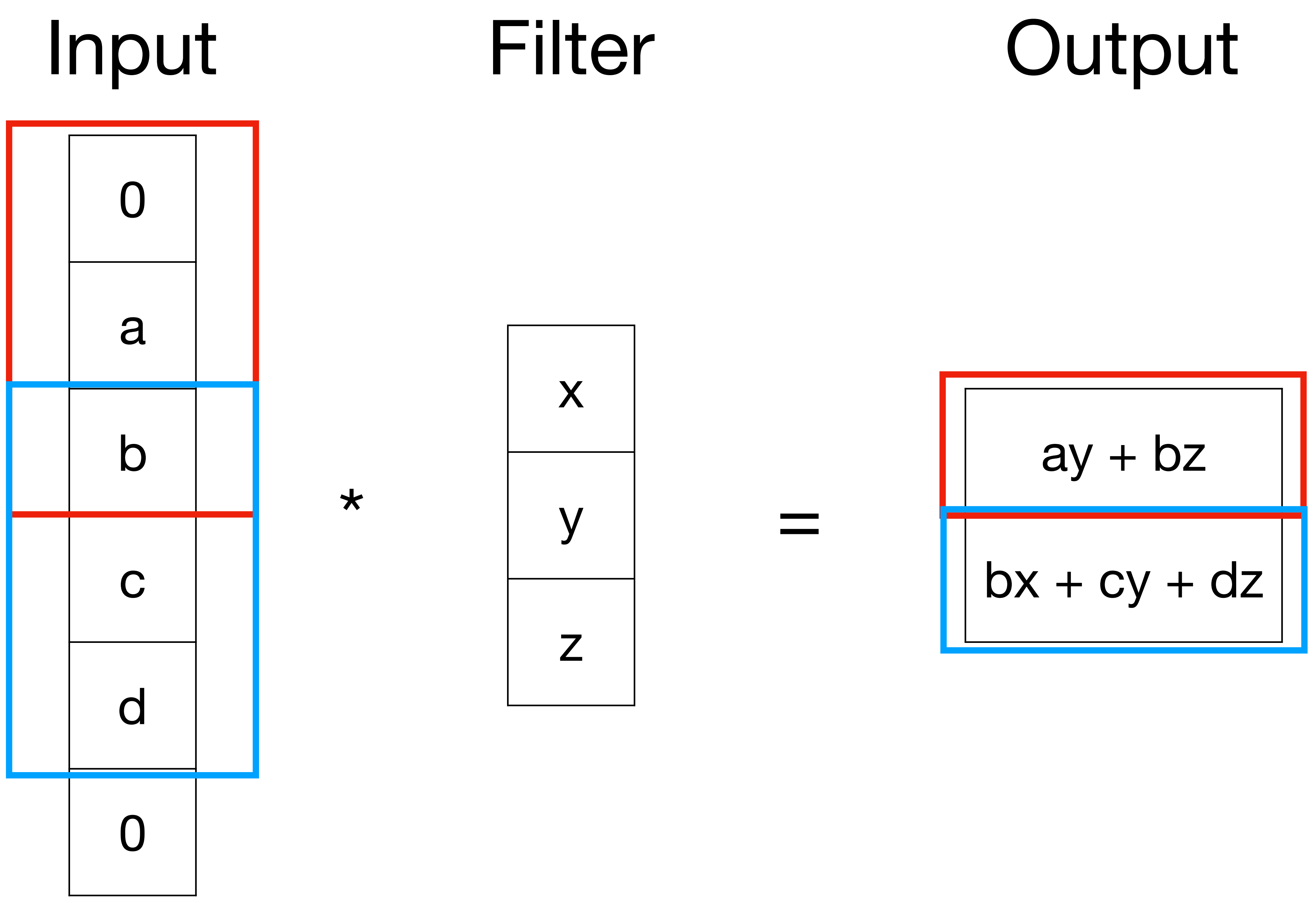
1D conv: kernel size = 3, stride = 2, padding=1



Learnable Upsampling: 1D Example

Convolution

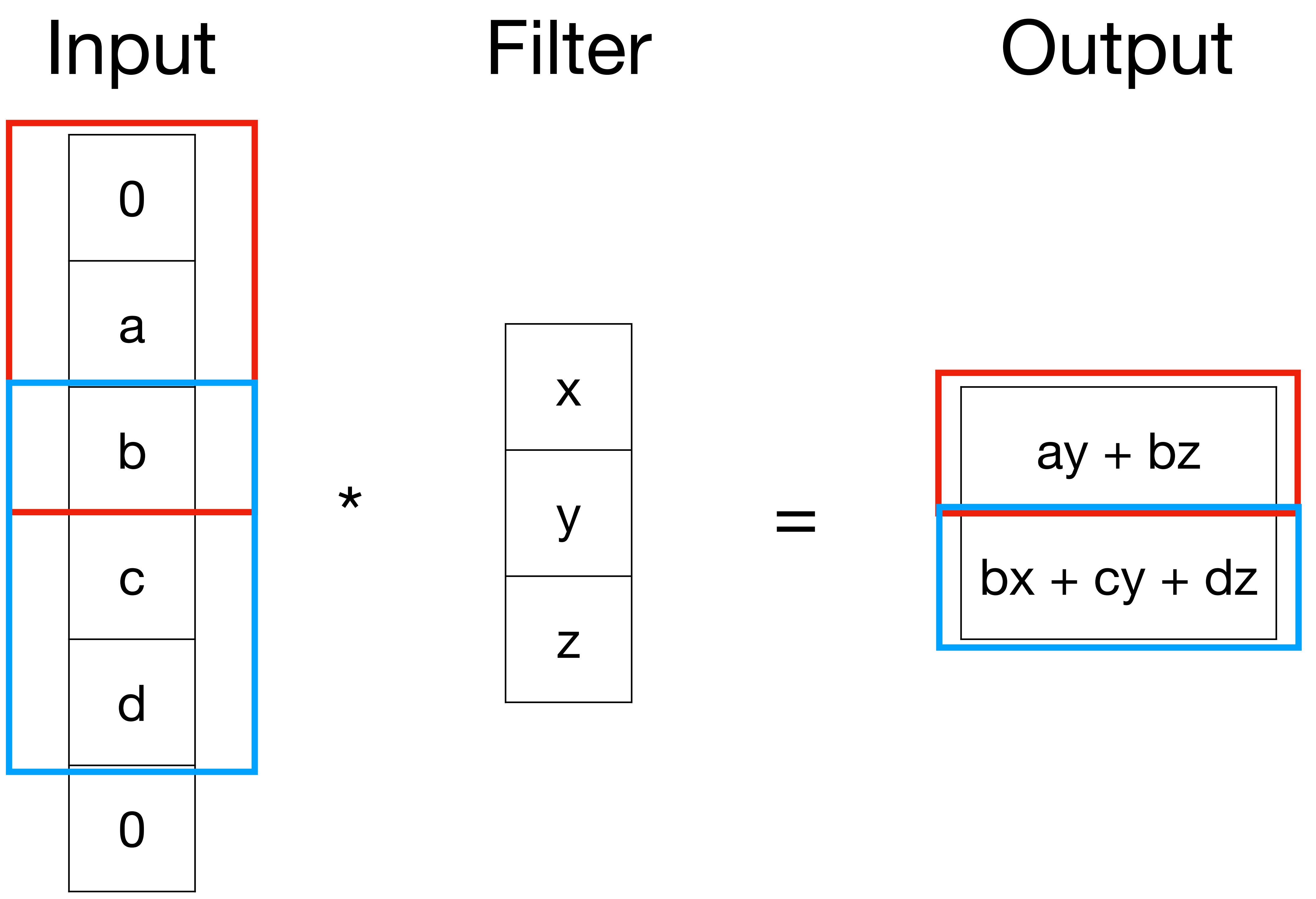
1D conv: kernel size = 3, stride = 2, padding=1



Learnable Upsampling: 1D Example

Convolution

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



as Matrix Multiplication:

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

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

Learnable Upsampling: 1D Example

Transposed Convolution

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

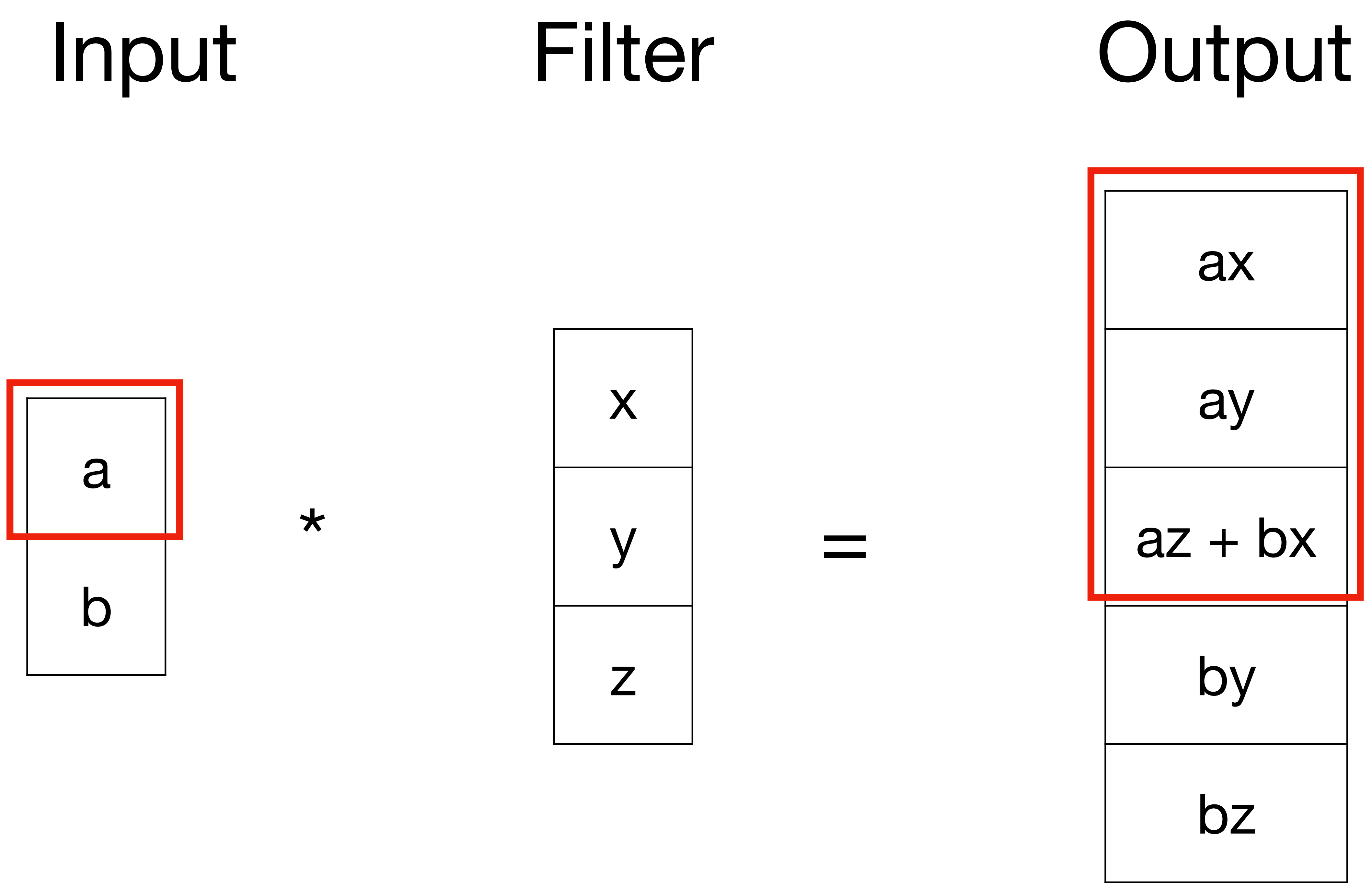
Input Filter Output

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

Learnable Upsampling: 1D Example

Transposed Convolution

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

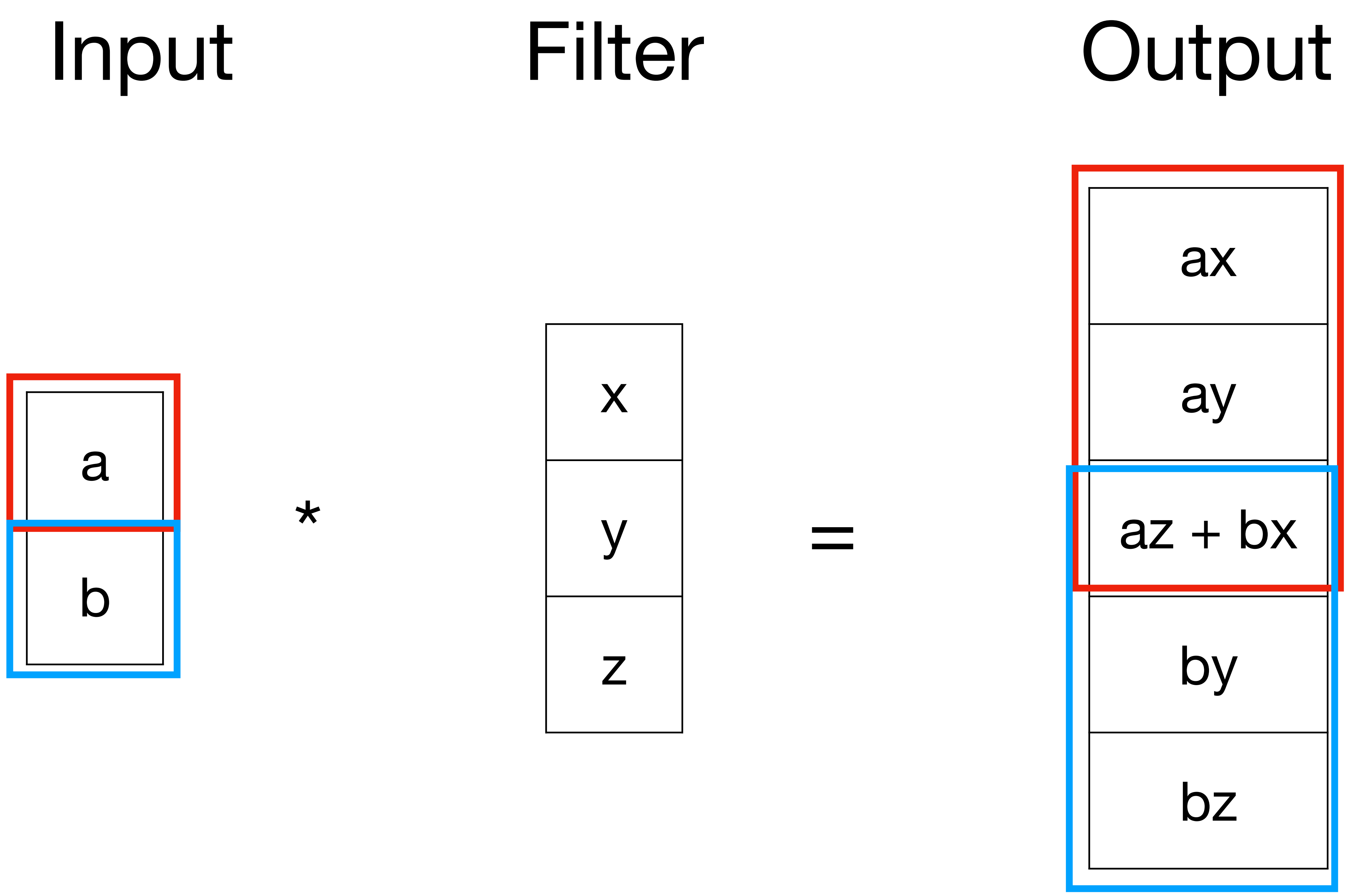


Credits to Fei-Fei Li, Jiajun Wu and Ruohan Gao (Stanford, Deep Learning for Computer Vision, 2022, cs231n)

Learnable Upsampling: 1D Example

Transposed Convolution

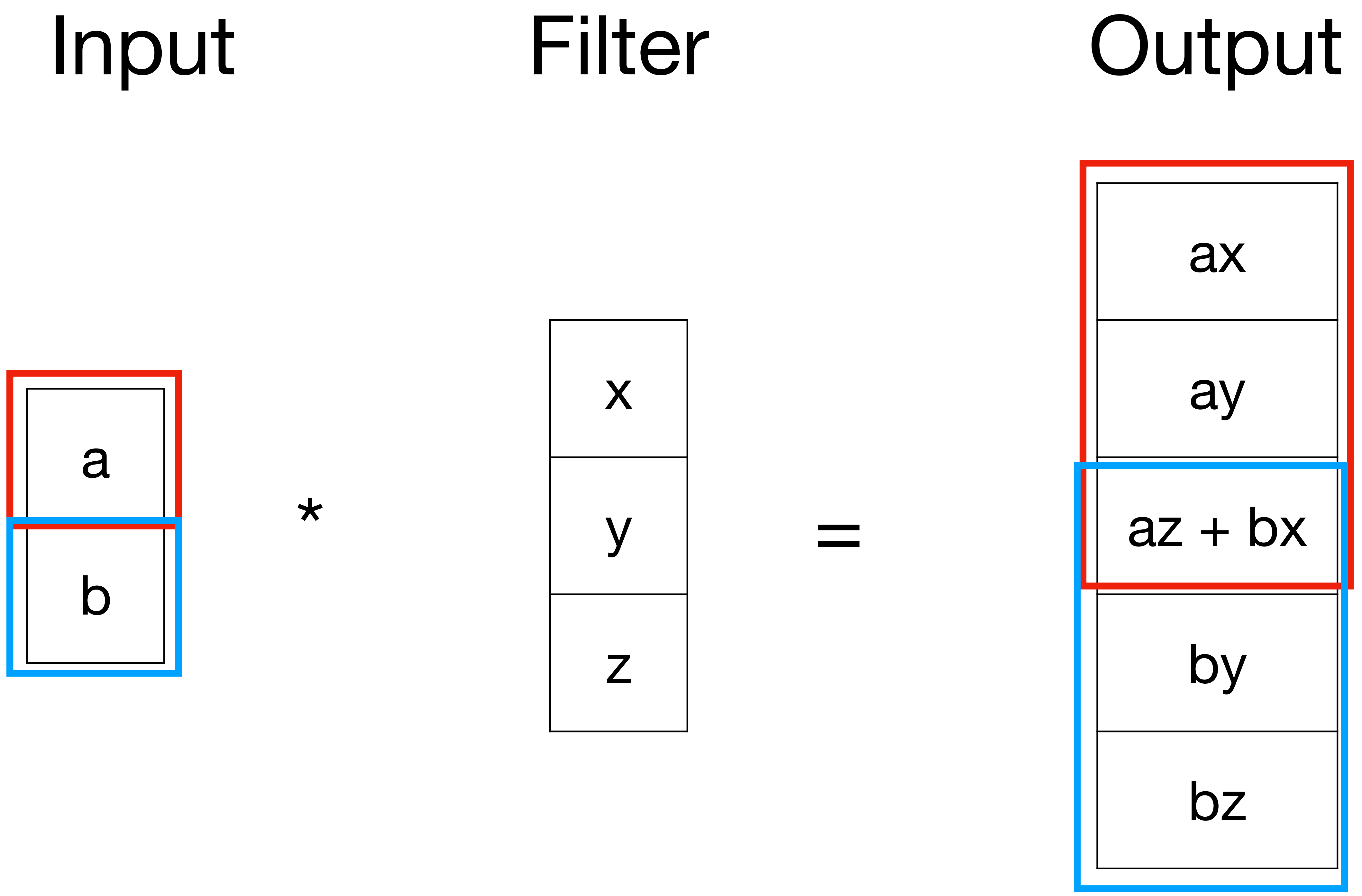
1D transposed conv: kernel size = 3, stride = 2, padding=1



Learnable Upsampling: 1D Example

Transposed Convolution

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



as Matrix Multiplication:

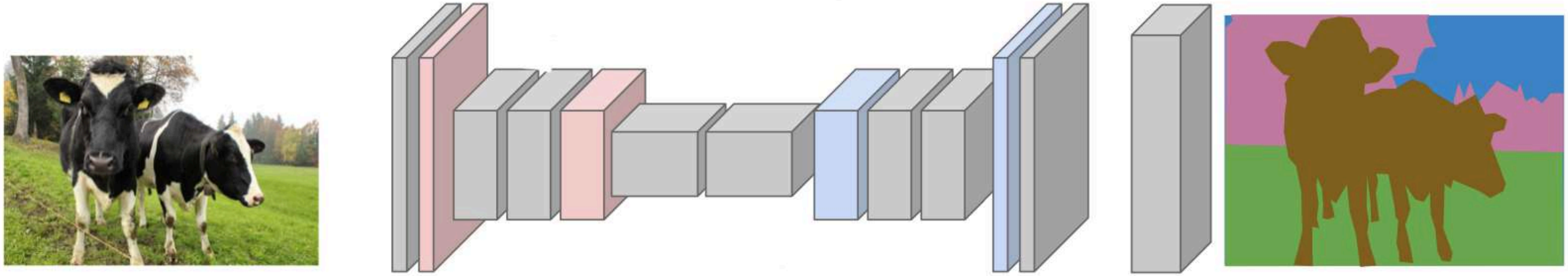
$$\vec{x}^T * \vec{a} = X^T \vec{a}$$

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

Idea: Fully Convolutional Network

Getting rid of redundant computations:

Design network with downsampling and upsampling inside the network



Downsampling:
Pooling, strided
convolution

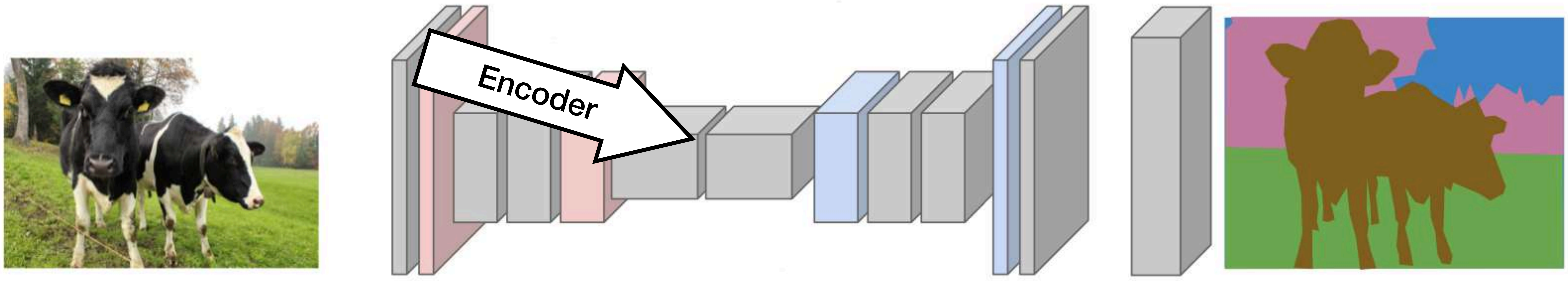
Upsampling:
Nearest neighbor, interpolation,
max unpooling, transposed convolution

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

Idea: Fully Convolutional Network

Getting rid of redundant computations:

Design network with downsampling and upsampling inside the network



Downsampling:
Pooling, strided convolution

Upsampling:
Nearest neighbor, interpolation, max unpooling, transposed convolution

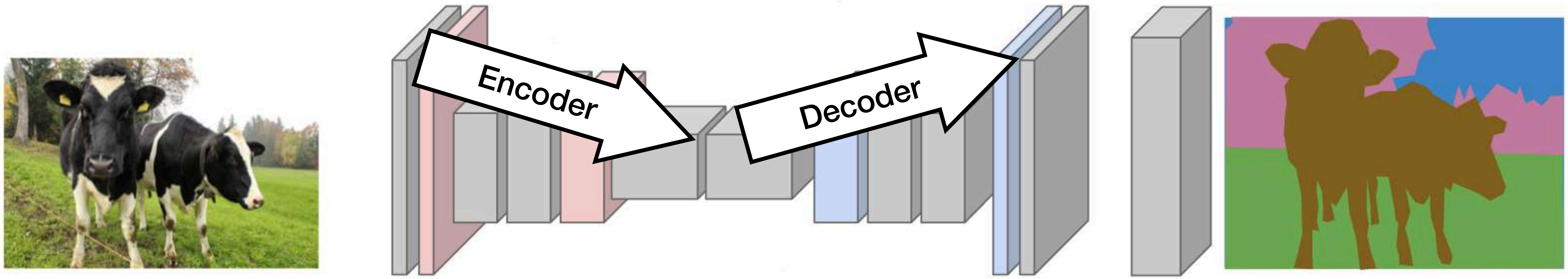
Long, Shelhamer and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015

Noh et al., “Learning Deconvolution Network for Semantic Segmentation”, ICCV 2015

Idea: Fully Convolutional Network

Getting rid of redundant computations:

Design network with downsampling and upsampling inside the network



Downsampling:
Pooling, strided convolution

Upsampling:
Nearest neighbor, interpolation, max unpooling, transposed convolution

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

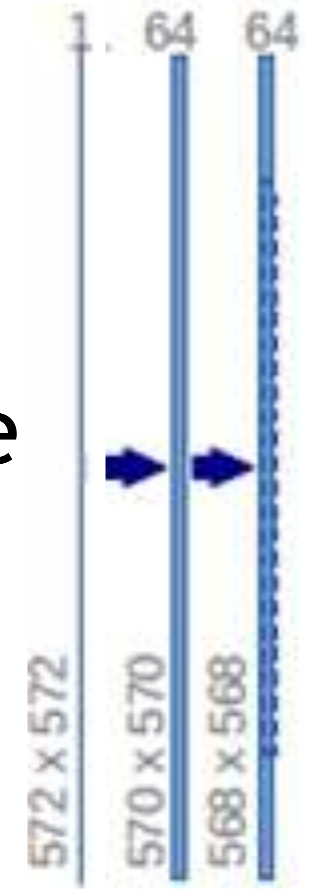
The U-Net

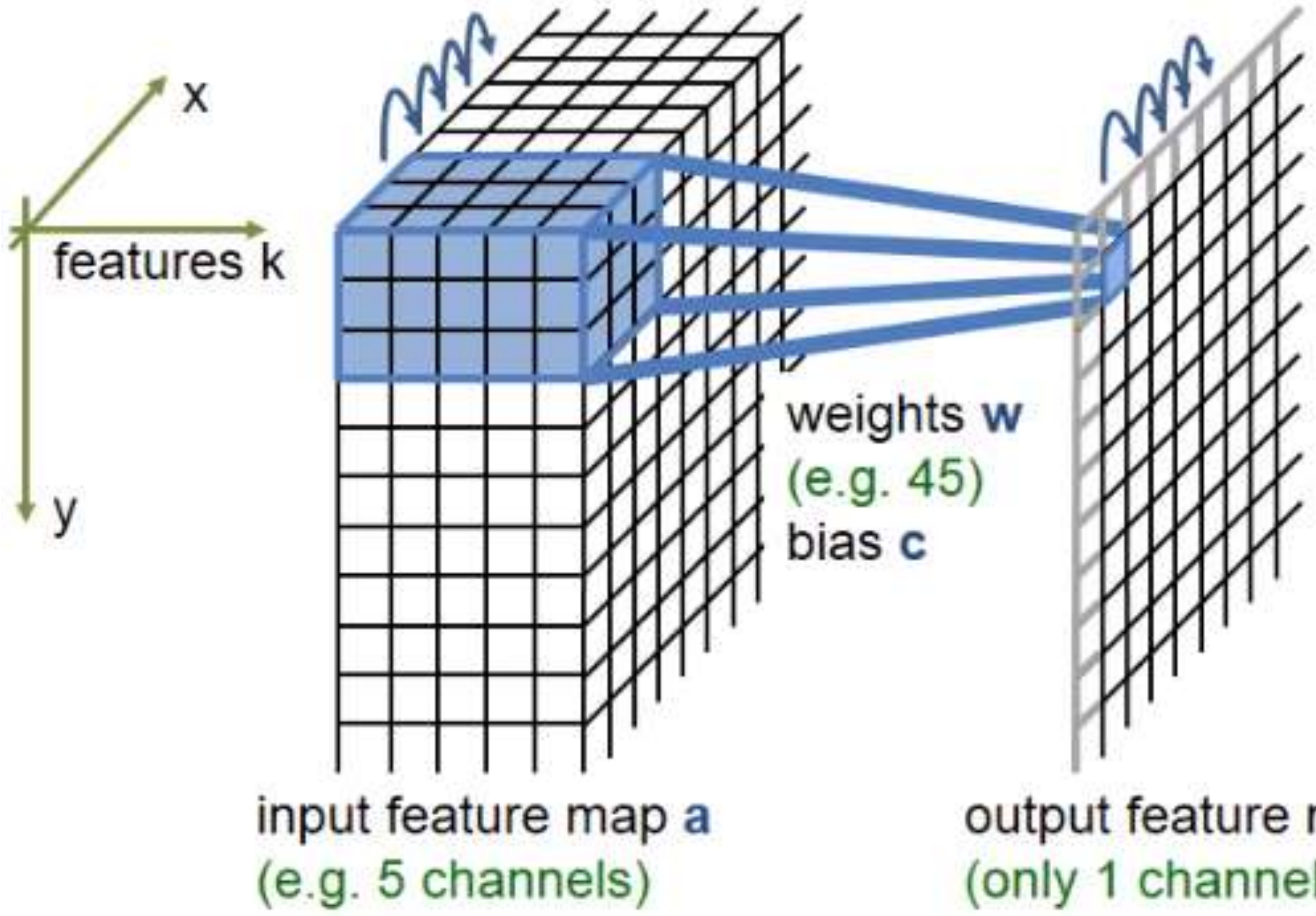
Back to full detail: The U-Net

Input
image
tile

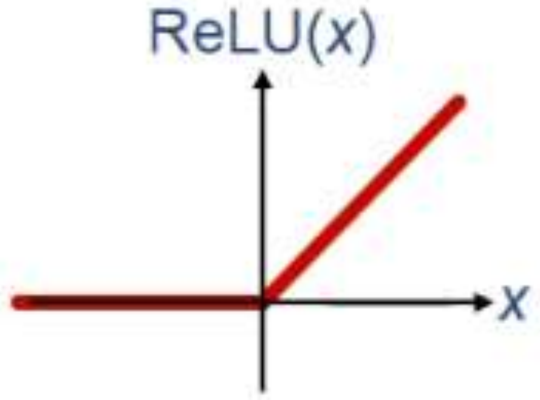
Back to full detail: The U-Net

Input image tile





- Only valid part of convolution is used.
- For 3x3 convolutions a 1-pixel border is lost



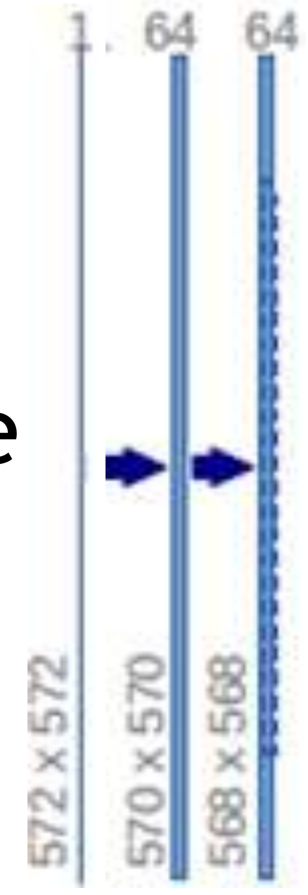
$$b_{x,y,l} = \mathrm{ReLU}\Big(\sum_{\substack{i\in\{-1,0,1\} \\ j\in\{-1,0,1\} \\ k\in\{1,\dots,K\}}} w_{i,j,k,l} \cdot a_{x+i,y+j,k} + c_l \Big)$$

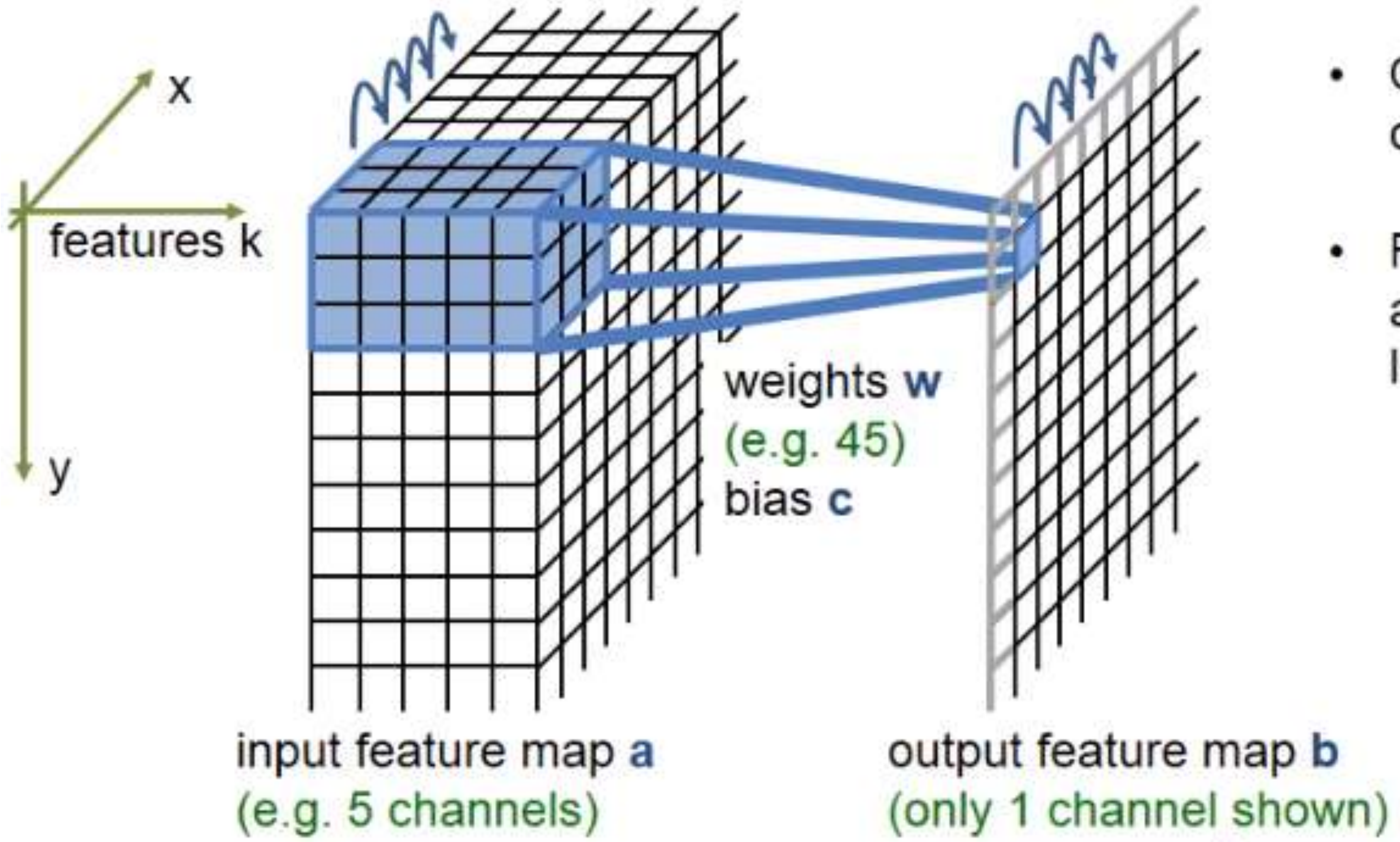
➡ conv 3x3, ReLU

http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

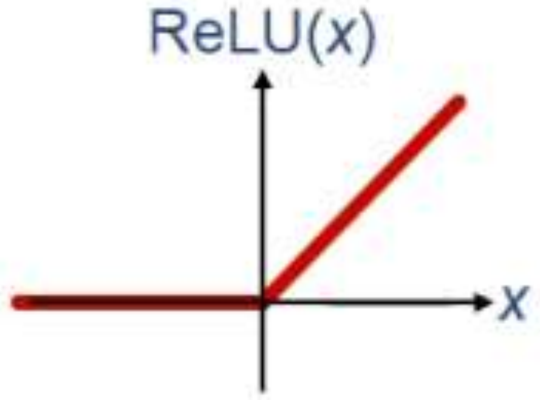
Back to full detail: The U-Net

Input
image
tile





- Only valid part of convolution is used.
- For 3x3 convolutions a 1-pixel border is lost



$$b_{x,y,l} = \mathrm{ReLU}\Big(\sum_{\substack{i\in\{-1,0,1\}\\ j\in\{-1,0,1\}\\ k\in\{1,\ldots,K\}}} w_{i,j,k,l} \cdot a_{x+i,y+j,k} + c_l \Big)$$

➡ conv 3x3, ReLU

W - Input volume size

F – Receptive field size (Filter Size)

P - Zero padding used on the border

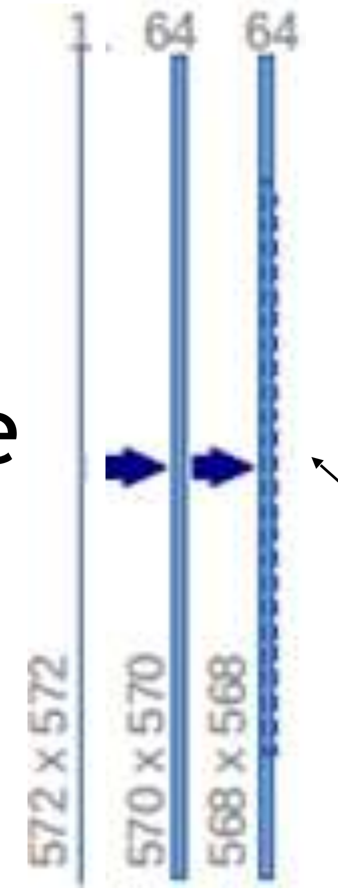
S - Stride

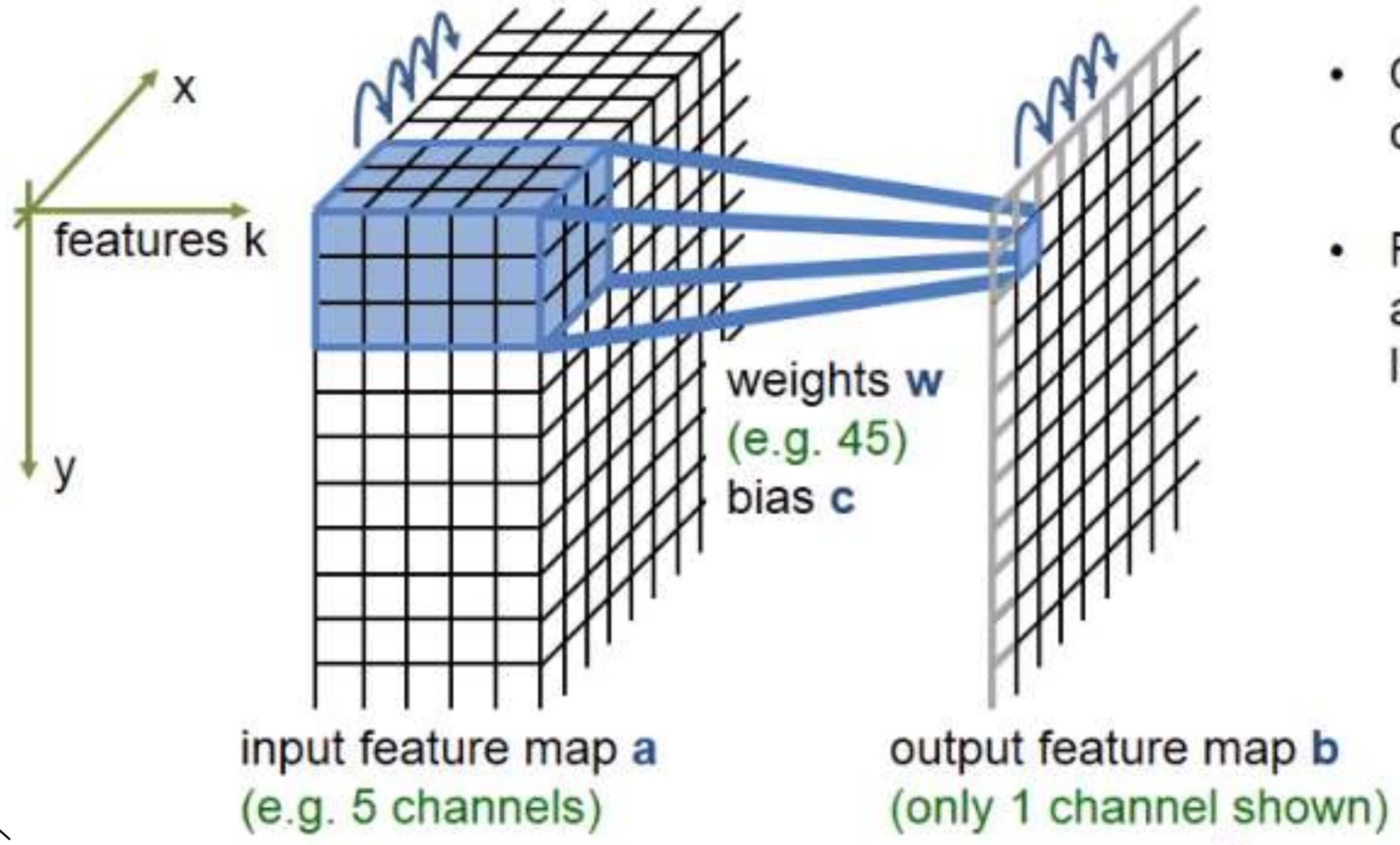
Output Size = (W−F+2P)/S+1

http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

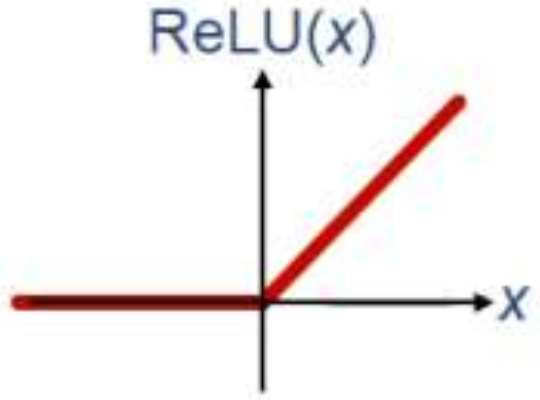
Back to full detail: The U-Net

Input image tile





- Only valid part of convolution is used.
- For 3x3 convolutions a 1-pixel border is lost



$$b_{x,y,l} = \mathrm{ReLU}\Big(\sum_{\substack{i\in\{-1,0,1\} \\ j\in\{-1,0,1\} \\ k\in\{1,\ldots,K\}}} w_{i,j,k,l} \cdot a_{x+i,y+j,k} + c_l \Big)$$

➡ conv 3x3, ReLU

Output Size (second conv)

= (570 – 3 +2*0)/1 + 1 = 568

→ 568 x 568

Output Size (first conv)

= (572 – 3 +2*0)/1 + 1 = 570

→ 570 x 570

W - Input volume size

F – Receptive field size (Filter Size)

P - Zero padding used on the border

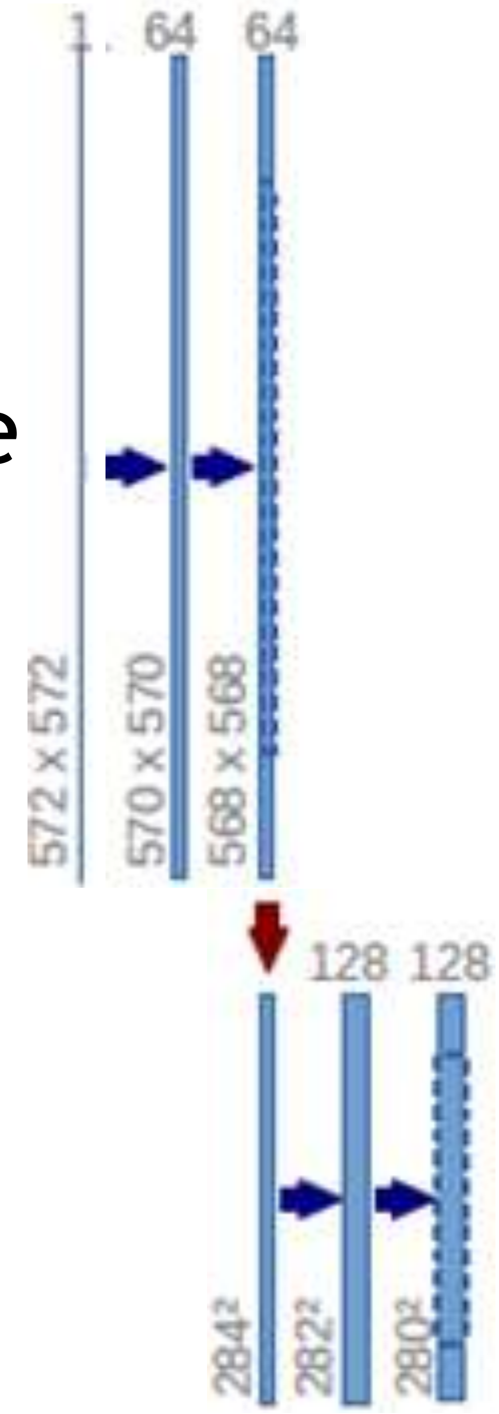
S - Stride

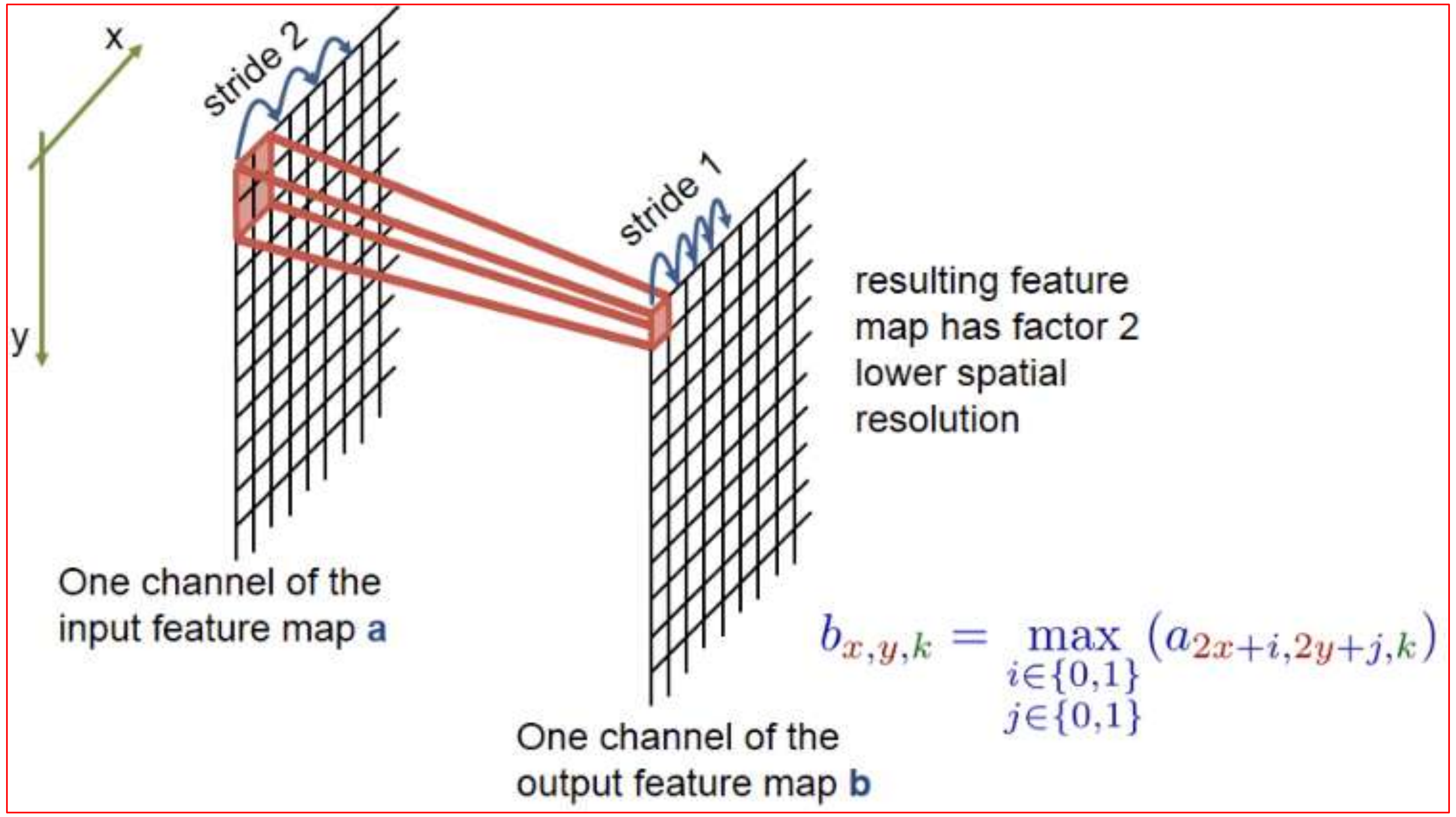
Output Size = (W−F+2P)/S+1

http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Back to full detail: The U-Net

Input
image
tile



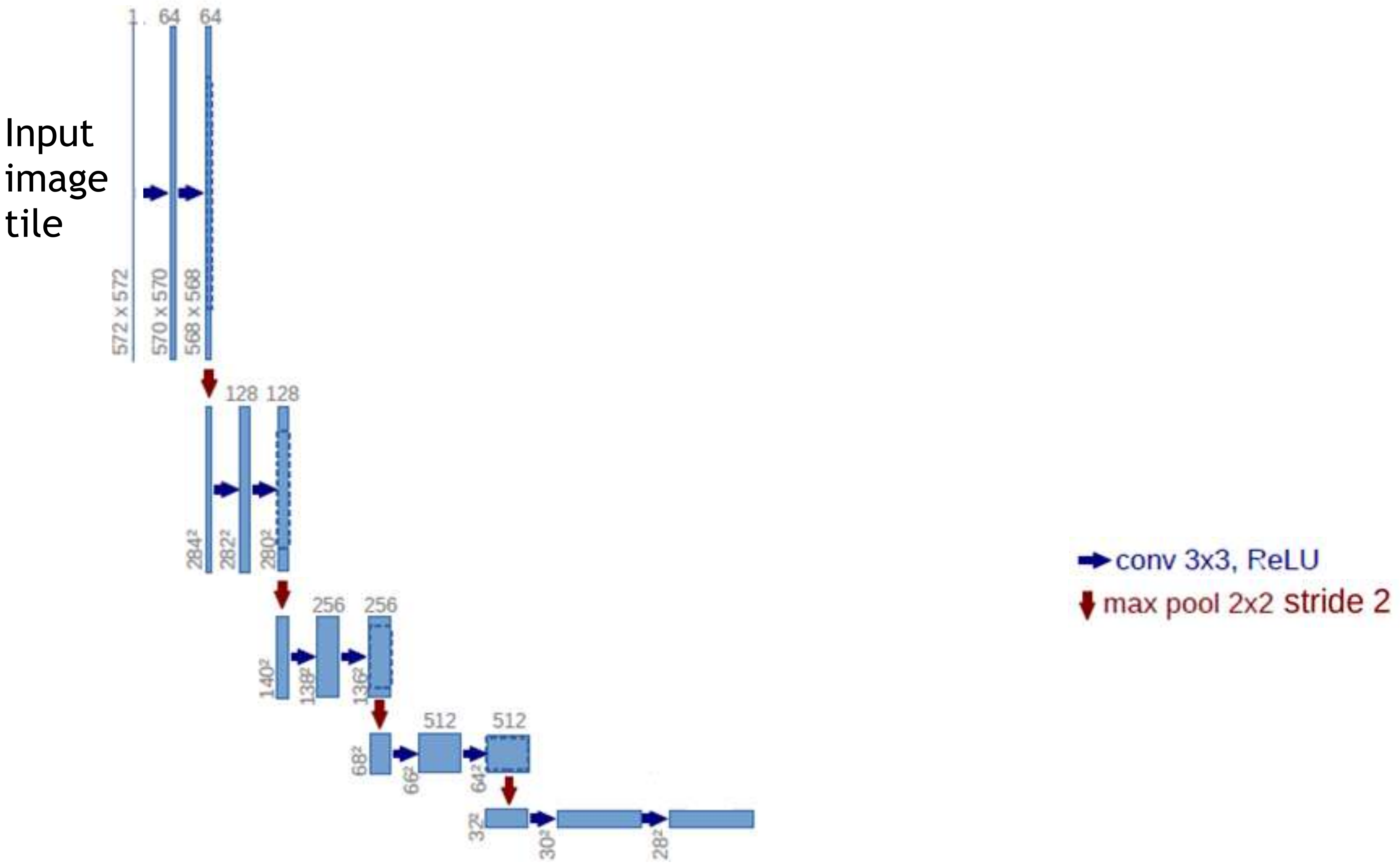


conv 3x3, ReLU

max pool 2x2 stride 2

http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

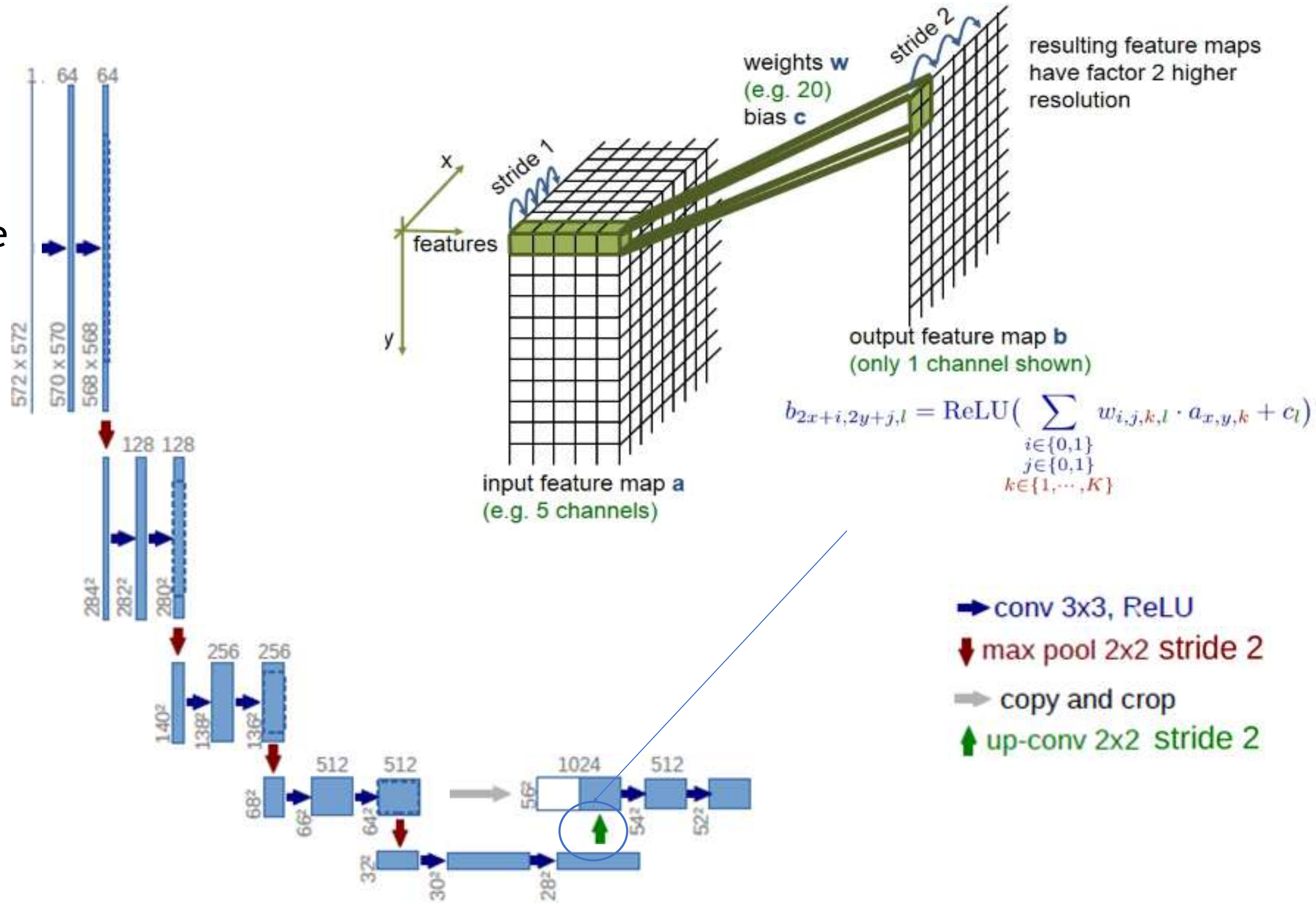
Back to full detail: The U-Net



http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Back to full detail: The U-Net

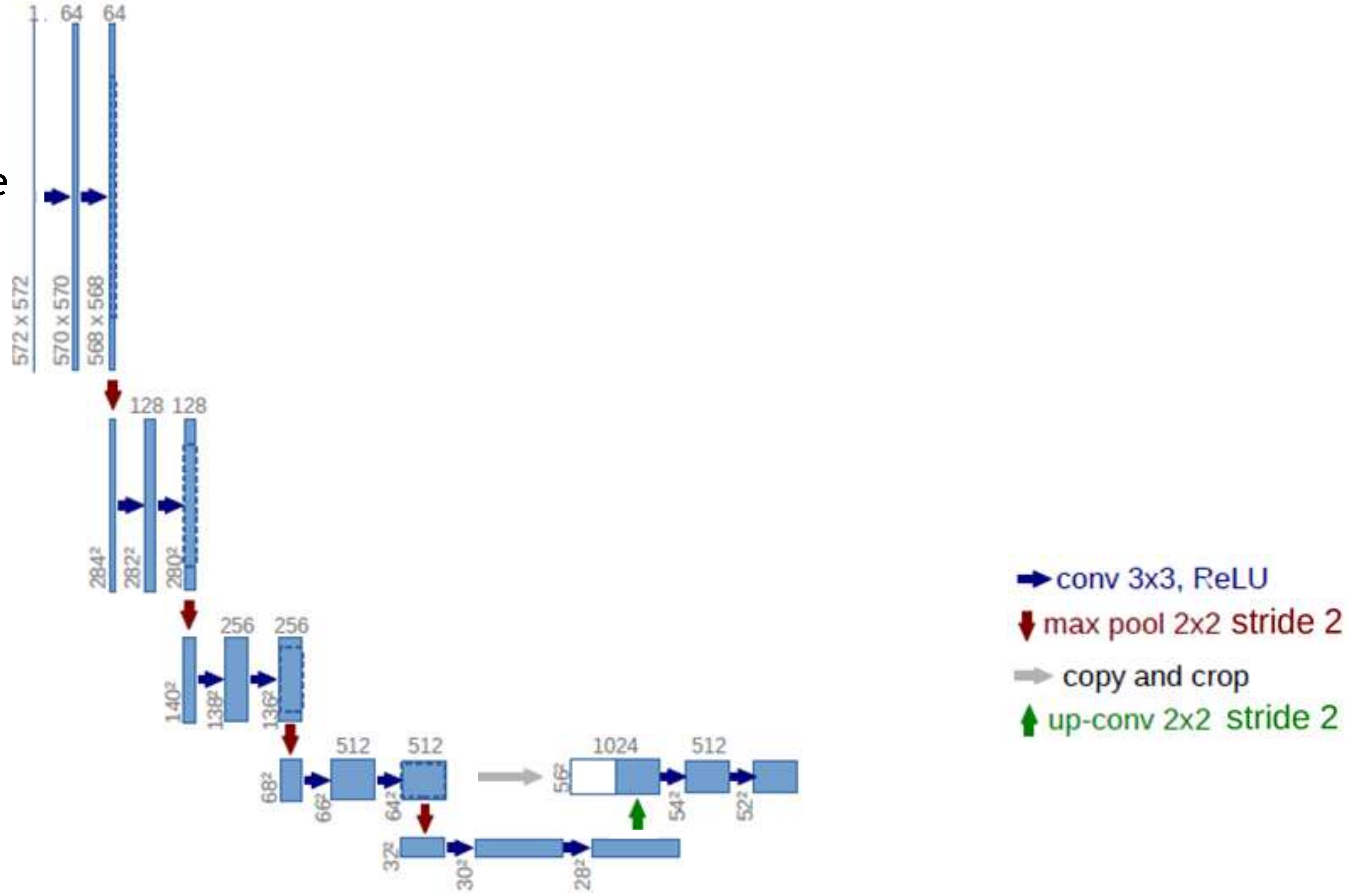
Input image tile



http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

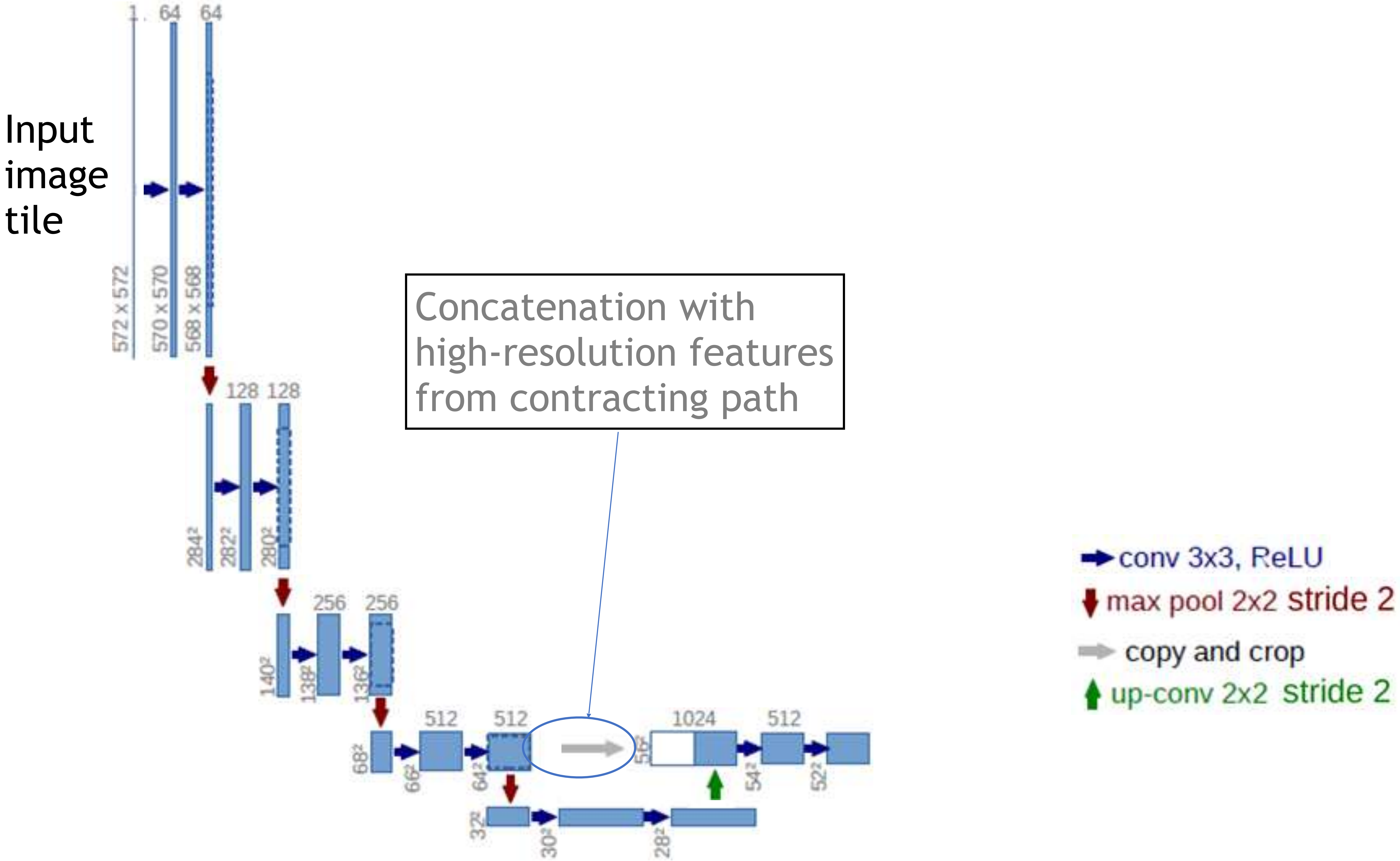
Back to full detail: The U-Net

Input
image
tile



http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

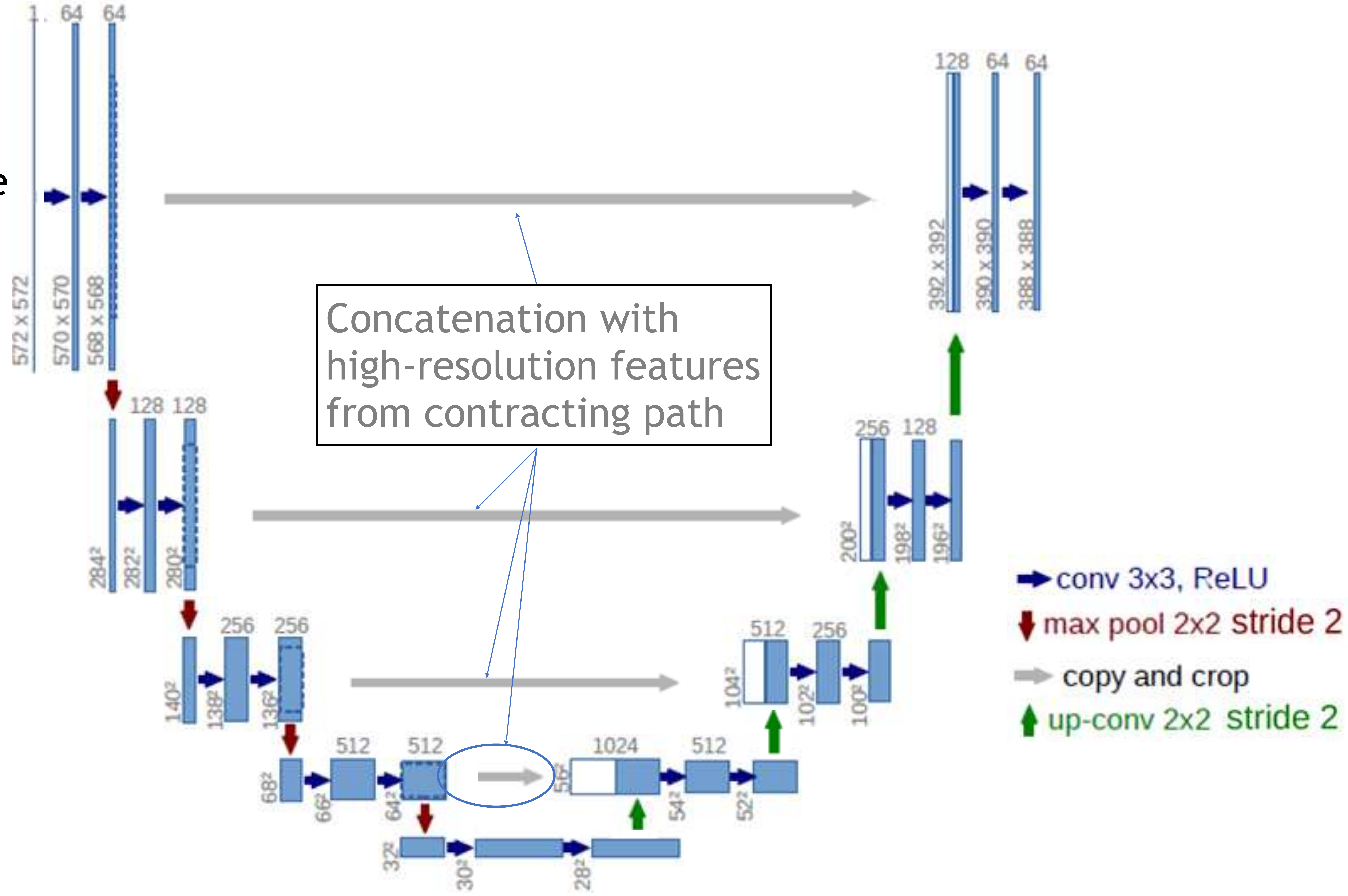
Back to full detail: The U-Net



http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

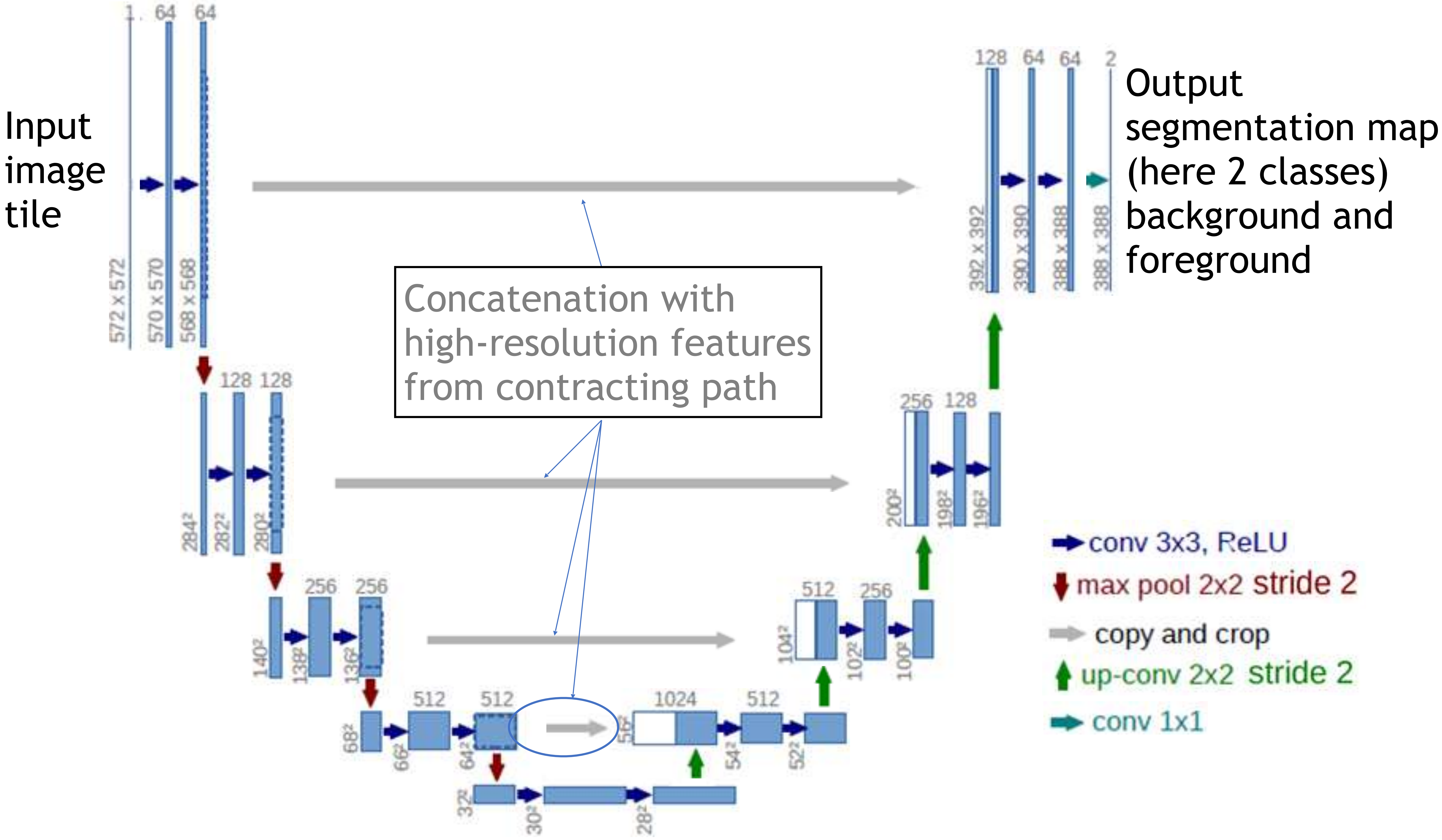
Back to full detail: The U-Net

Input
image
tile



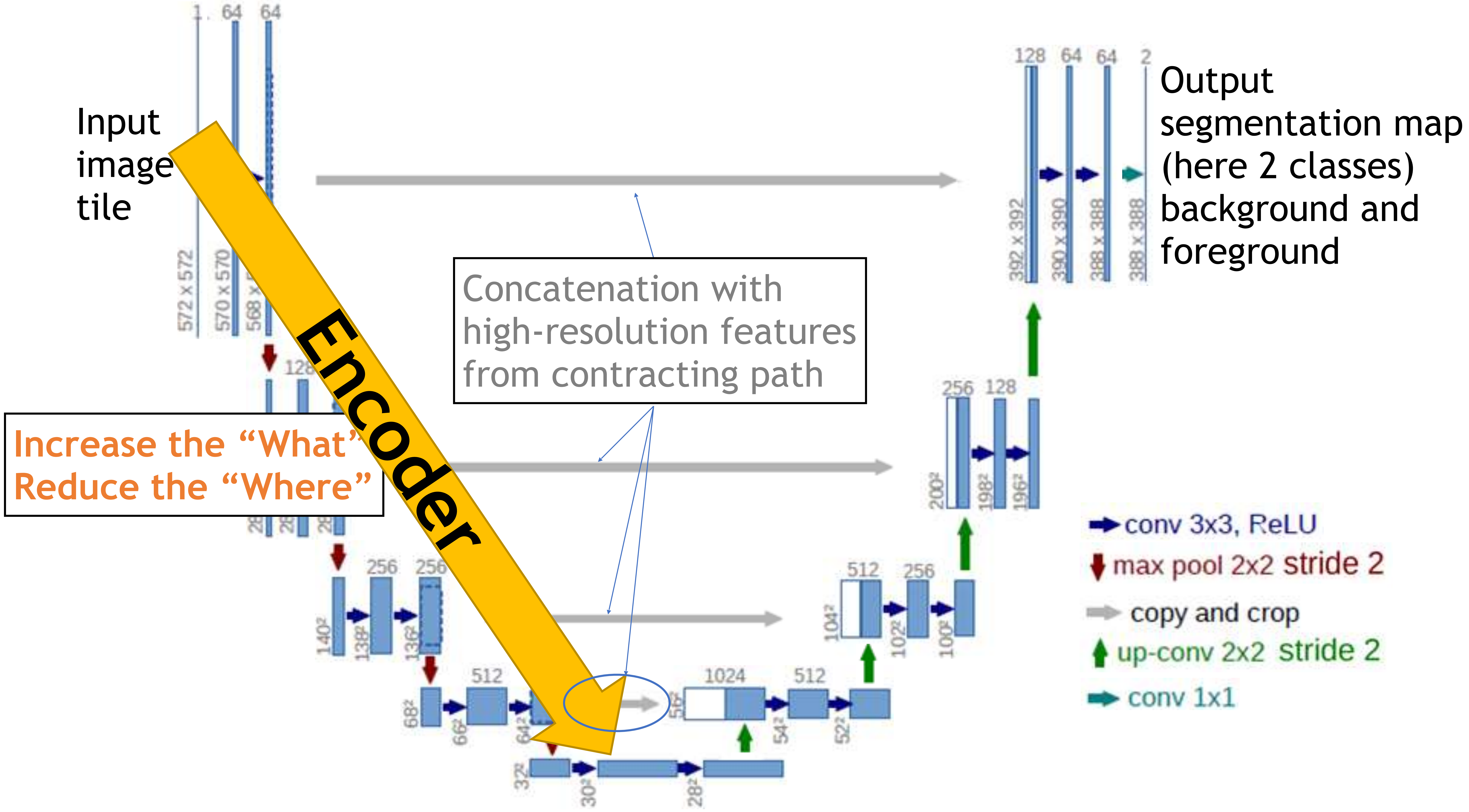
http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Back to full detail: The U-Net



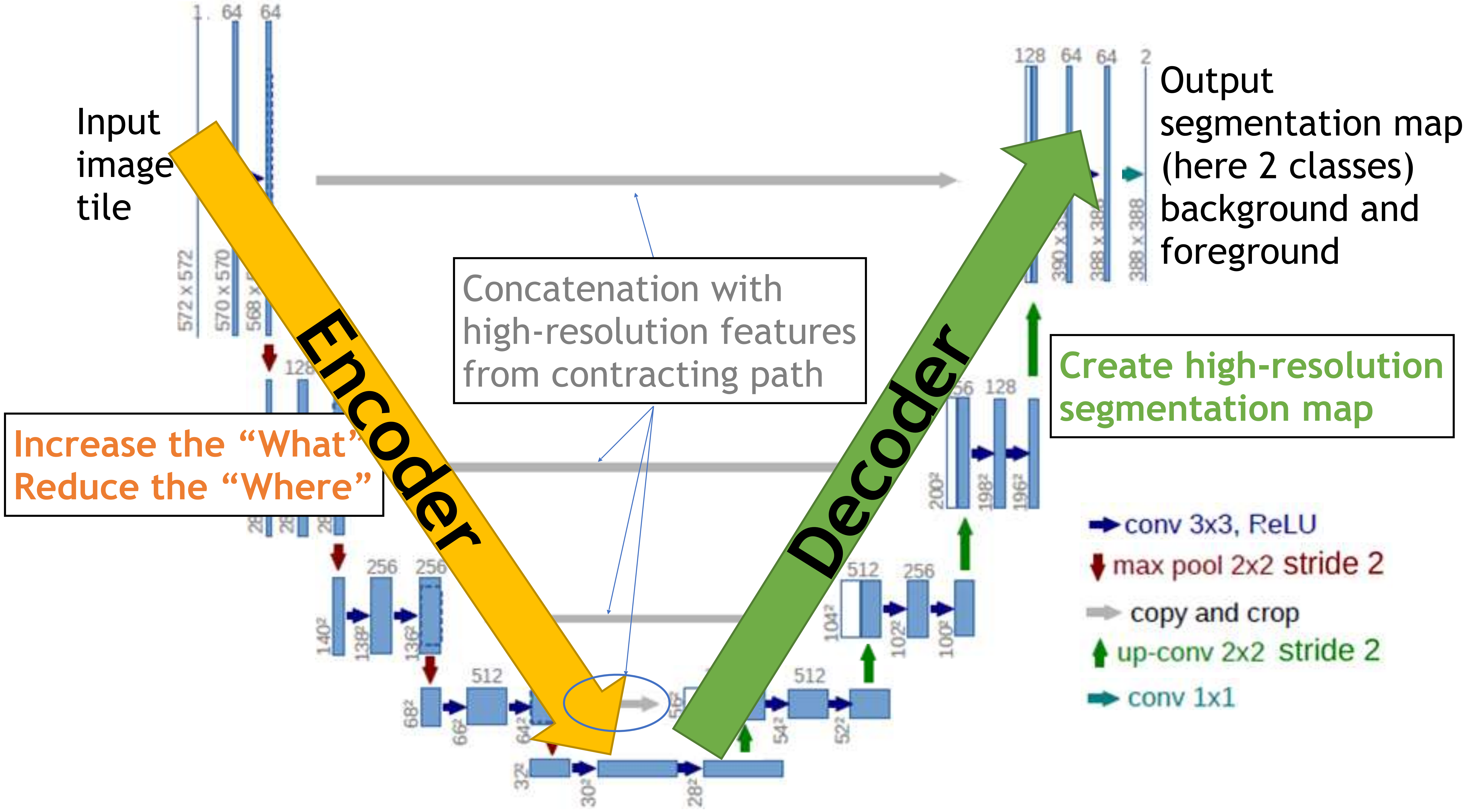
http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Back to full detail: The U-Net



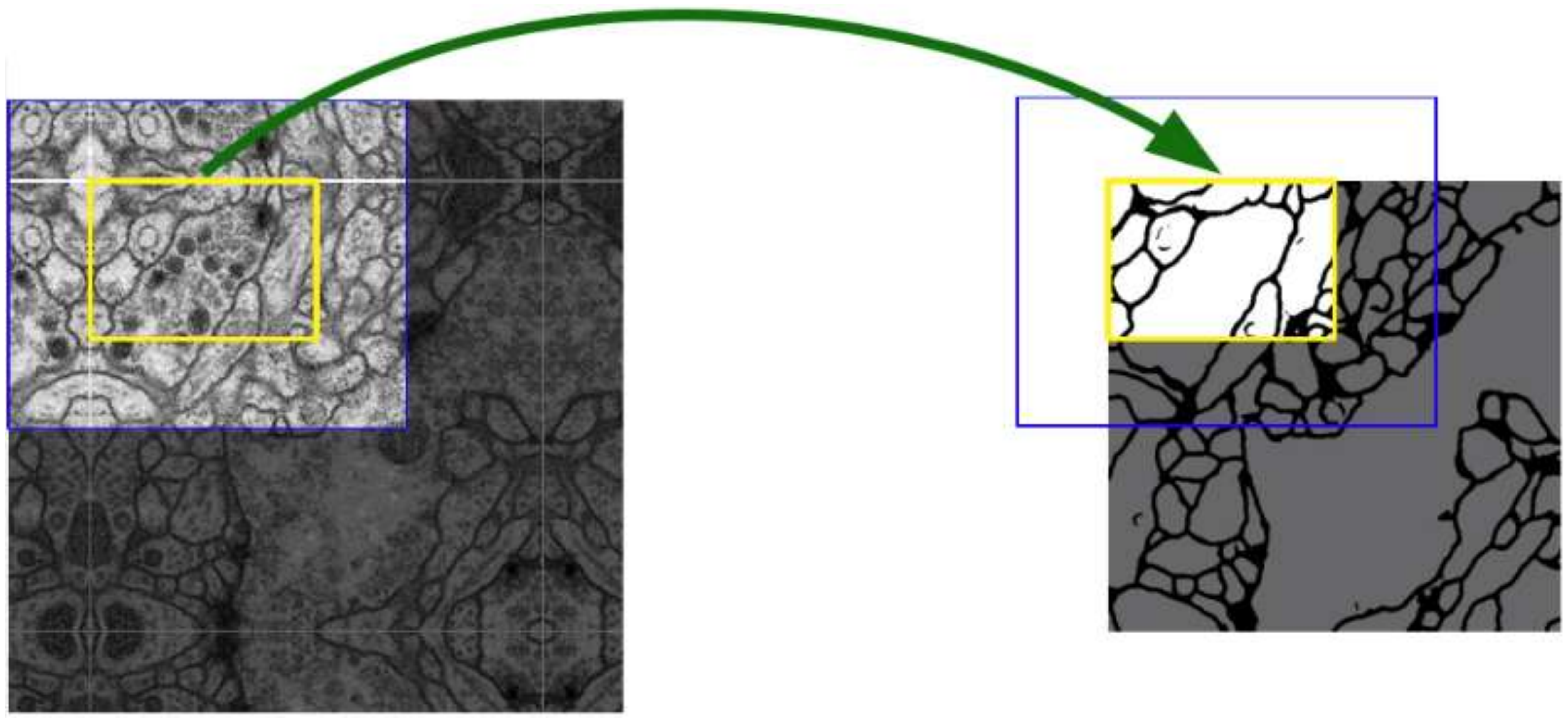
http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Back to full detail: The U-Net



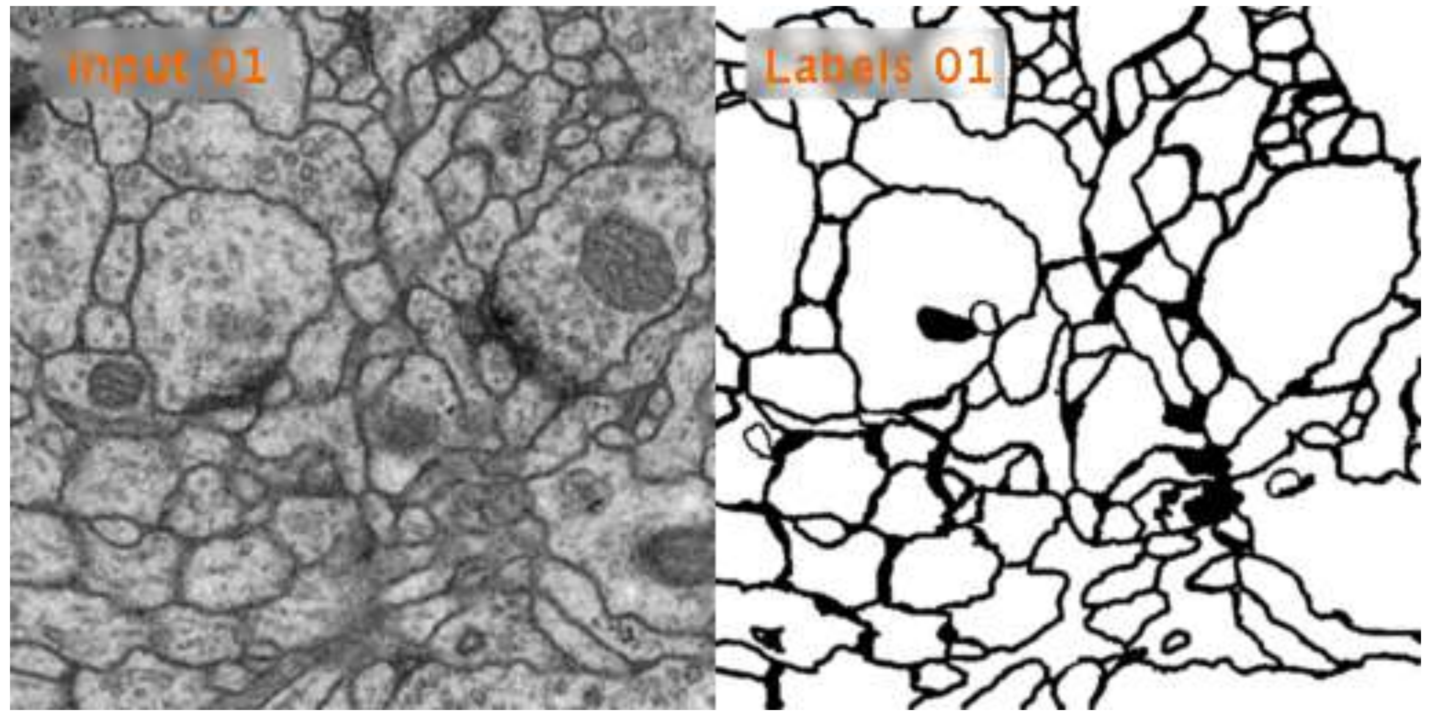
http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Tiling strategy for arbitrary large images



- Segmentation of the yellow area uses input data of the blue area
- Raw data extrapolation by mirroring

U-Net takes the lead in the ISBI 2012 EM segmentation challenge

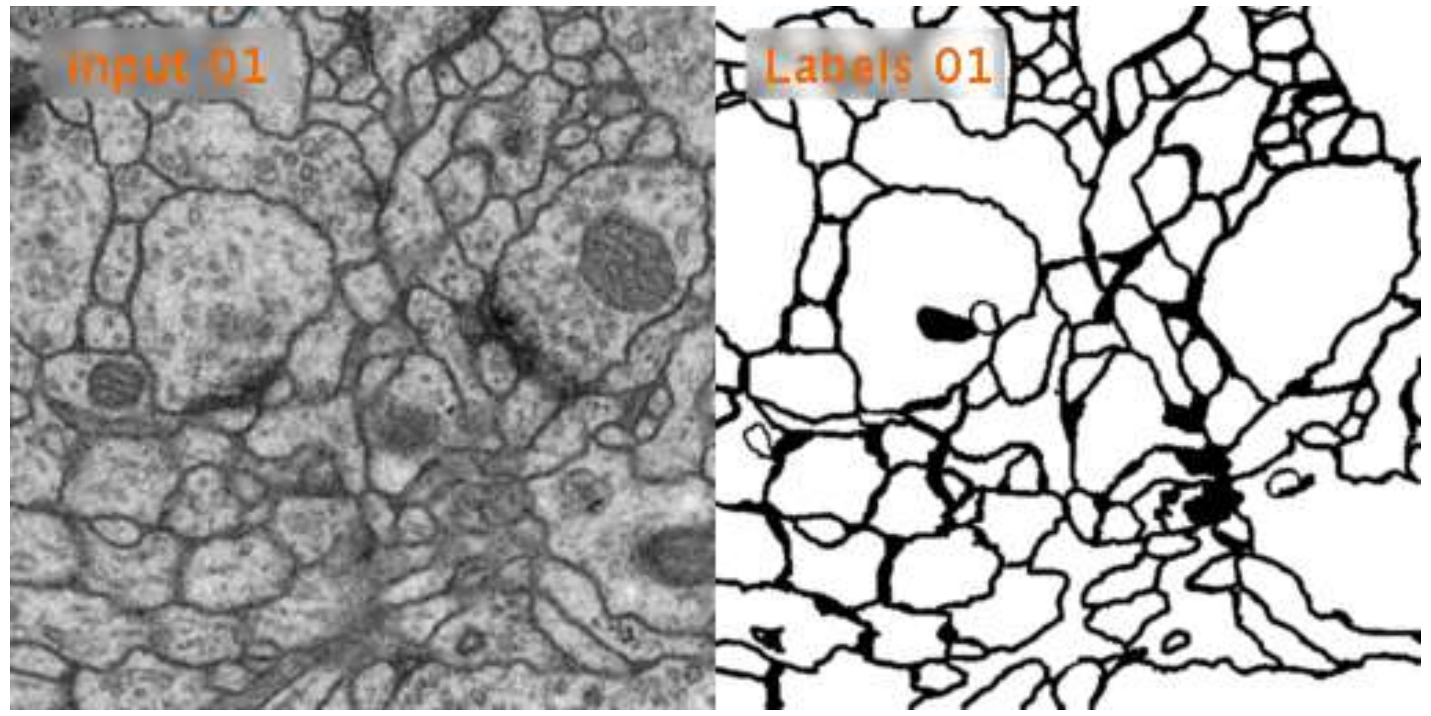


Raw image Ground truth

Rank	Group name	Warping Error	Rand Error	Pixel Error
	** human values **	0.000005	0.0021	0.0010
1.	u-net	**0.000353**	0.0382	0.0611
2.	DIVE-SCI	0.000355	0.0305	0.0584
3.	IDSIA [1]	0.000420	0.0504	0.0613
4.	DIVE	0.000430	0.0545	**0.0582**
⋮				
10.	IDSIA-SCI	0.000653	**0.0189**	0.1027

- outperforms the sliding window CNN which was leading the challenge from 2012 until 2015

U-Net takes the lead in the ISBI 2012 EM segmentation challenge



Raw image Ground truth

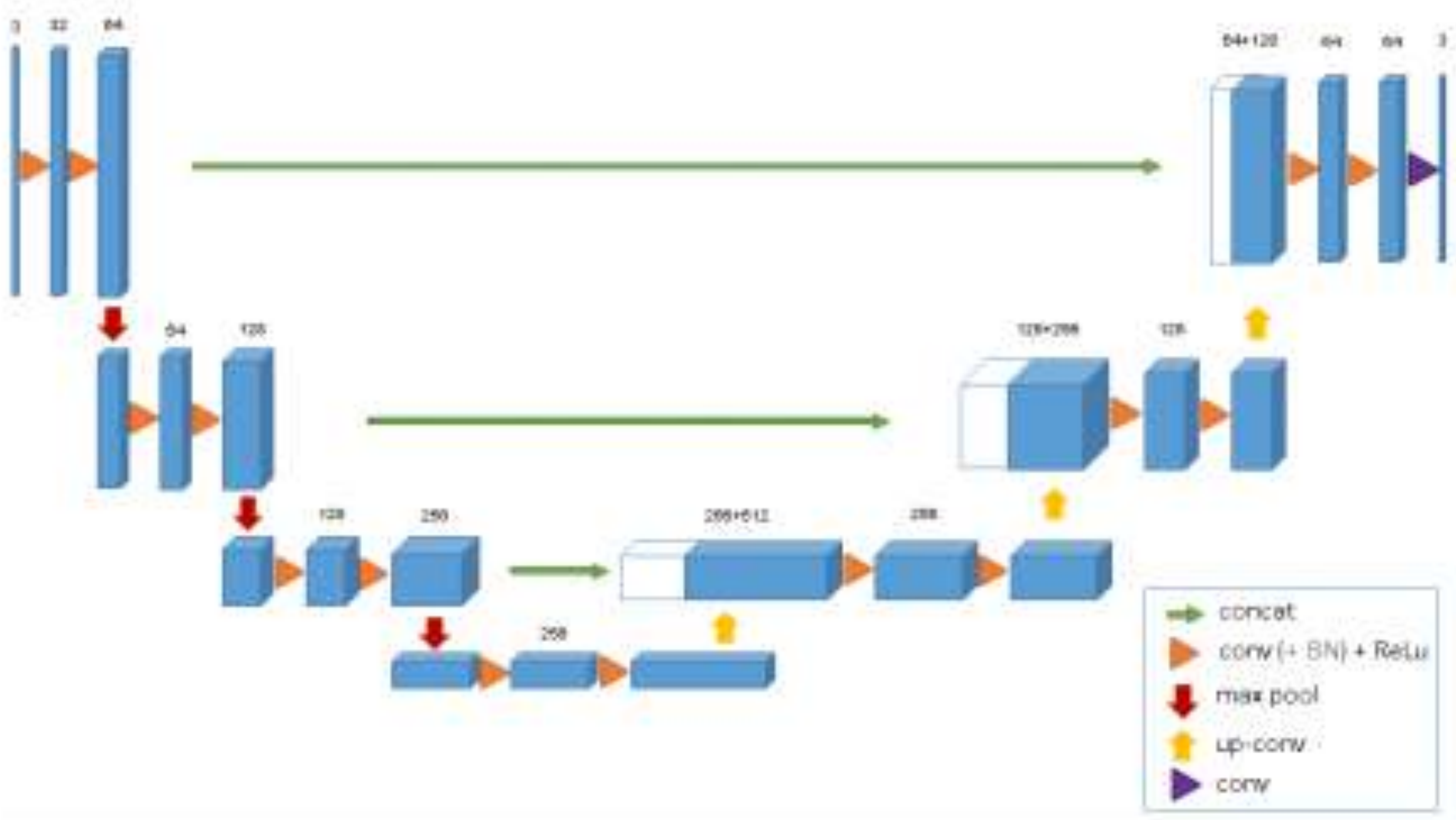
Rank	Group name	Warping Error	Rand Error	Pixel Error
	** human values **	0.000005	0.0021	0.0010
1.	u-net	**0.000353**	0.0382	0.0611
2.	DIVE-SCI	0.000355	0.0305	0.0584
3.	IDSIA [1]	0.000420	0.0504	0.0613
4.	DIVE	0.000430	0.0545	**0.0582**
⋮				
10.	IDSIA-SCI	0.000653	**0.0189**	0.1027

- outperforms the sliding window CNN which was leading the challenge from 2012 until 2015

3D U-Net Architecture

Volumetric segmentation with the 3D U-Net
→ 3x3x3 convolutions, 2x2x2 max pooling, 2x2x2 upconvolutions

Input:
132x132x116
voxel



Output:
44x44x28
voxel

https://arxiv.org/abs/1606.06650

U-Net Summary

U-Net Summary

U-Net advantages

- Flexible use for any pixel-wise prediction task
- Doesn’t contain any fully connected layers
- Faster than sliding-window
- Powerful segmentation tool also in scenarios with limited data
- Competitive accuracy (given proper training, dataset and training time)
- Achieves best performance in a broad range of biomedical applications

U-Net Summary

nature methods ARTICLES
https://doi.org/10.1038/s41592-020-01008-z
Check for updates

nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation

Fabian Isensee[1,2,6], Paul F. Jaeger[1,6], Simon A. A. Kohl[1,3], Jens Petersen[1,4] and Klaus H. Maier-Hein[1,5]

Biomedical imaging is a driver of scientific discovery and a core component of medical care and is being stimulated by the field of deep learning. While semantic segmentation algorithms enable image analysis and quantification in many applications, the design of respective specialized solutions is non-trivial and highly dependent on dataset properties and hardware conditions. We developed nnU-Net, a deep learning-based segmentation method that automatically configures itself, including preprocessing, network architecture, training and post-processing for any new task. The key design choices in this process are modeled as a set of fixed parameters, interdependent rules and empirical decisions. Without manual intervention, nnU-Net surpasses most existing approaches, including highly specialized solutions on 23 public datasets used in international biomedical segmentation competitions. We make nnU-Net publicly available as an out-of-the-box tool, rendering state-of-the-art segmentation accessible to a broad audience by requiring neither expert knowledge nor computing resources beyond standard network training.

published 2021

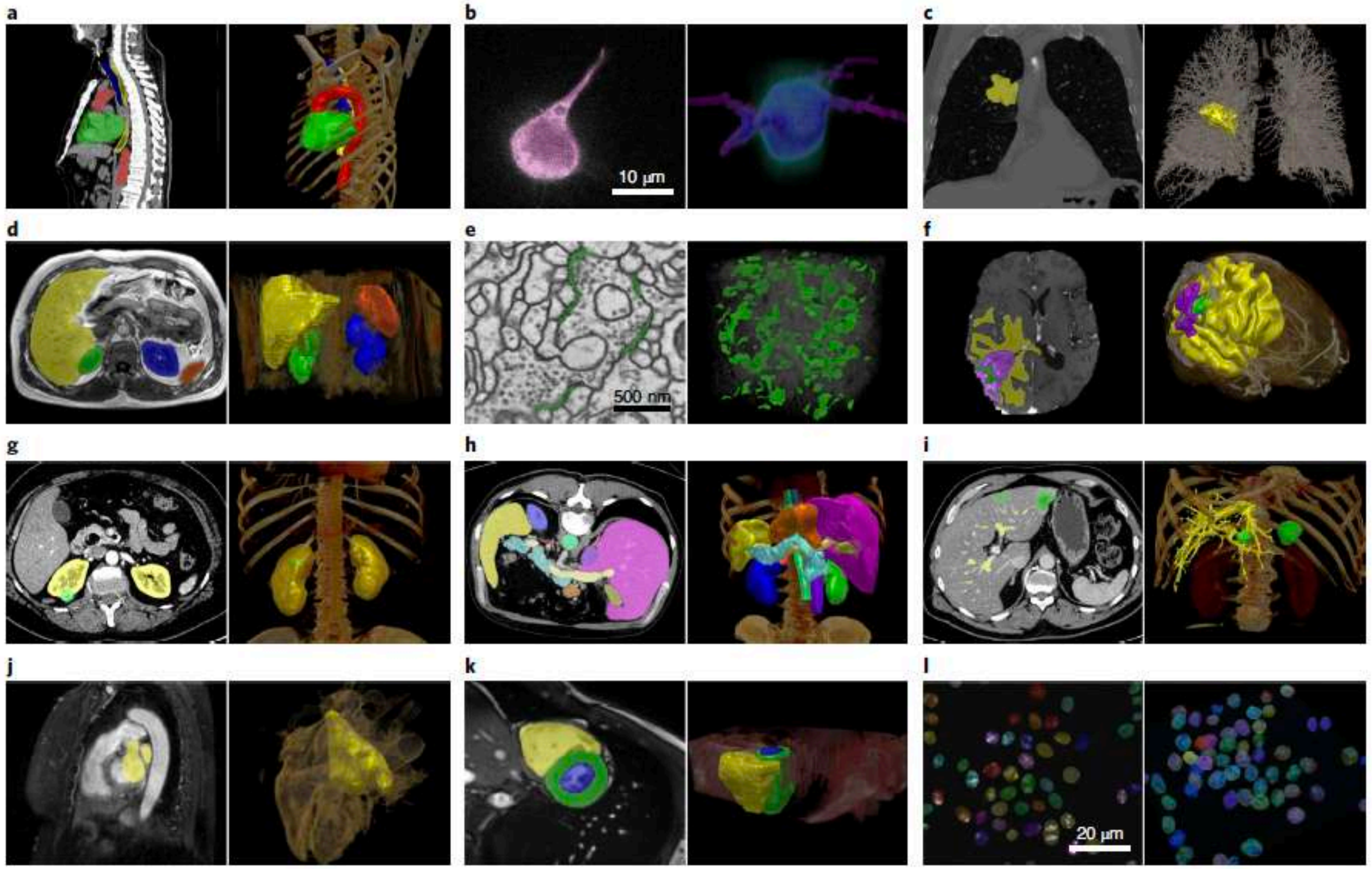


Fig. 1 | nnU-Net handles a broad variety of datasets and target image properties. All examples originate from test sets of different international segmentation challenges to which nnU-Net was applied. Target structures for each dataset are shown in 2D projected onto the raw data (left) and in 3D together with a volume rendering of the raw data (right). All visualizations were created with the MITK Workbench[35]. **a**, Heart (green), aorta (red), trachea (blue) and esophagus (yellow) in CT images (dataset (D)18)[14]. **b**, A549 lung cancer cells (purple) in FM (D22)[36,37]. **c**, Lung nodules (yellow) in CT images (D6)[14]. **d**, Liver (yellow), spleen (orange), left and right kidneys (blue and green, respectively) in T1 in-phase MRI (D16)[20]. **e**, Synaptic clefts (green) in EM scans (D19) (https://cremi.org/). **f**, Edema (yellow), enhancing tumor (purple), necrosis (green) in MRI (T1, T1 with contrast agent, T2, FLAIR) (D1)[14,38]. **g**, Kidneys (yellow) and kidney tumors (green) in CT images (D17)[21]. **h**, Thirteen abdominal organs in CT images (D11)[16]. **i**, Hepatic vessels (yellow) and liver tumors (green) in CT (D8)[14]. **j**, Left ventricle (yellow) in MRI (D2)[14]. **k**, Right ventricle (yellow), left ventricular cavity (blue) and myocardium of left ventricle (green) in cine MRI (D13)[6]. **l**, HL60 cell nuclei (instance segmentation, one color per instance) in FM (D21)[39].

U-Net Summary

nature methods
ARTICLES
https://doi.org/10.1038/s41592-020-01008-z
Check for updates

nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation

Fabian Isensee[1,2,6], Paul F. Jaeger[1,6], Simon A. A. Kohl[1,3], Jens Petersen[1,4] and Klaus H. Maier-Hein[1,5]

Biomedical imaging is a driver of scientific discovery and a core component of medical care and is being stimulated by the field of deep learning. While semantic segmentation algorithms enable image analysis and quantification in many applications, the design of respective specialized solutions is non-trivial and highly dependent on dataset properties and hardware conditions. We developed nnU-Net, a deep learning-based segmentation method that automatically configures itself, including preprocessing, network architecture, training and post-processing for any new task. The key design choices in this process are modeled as a set of fixed parameters, interdependent rules and empirical decisions. Without manual intervention, nnU-Net surpasses most existing approaches, including highly specialized solutions on 23 public datasets used in international biomedical segmentation competitions. We make nnU-Net publicly available as an out-of-the-box tool, rendering state-of-the-art segmentation accessible to a broad audience by requiring neither expert knowledge nor computing resources beyond standard network training.

published 2021

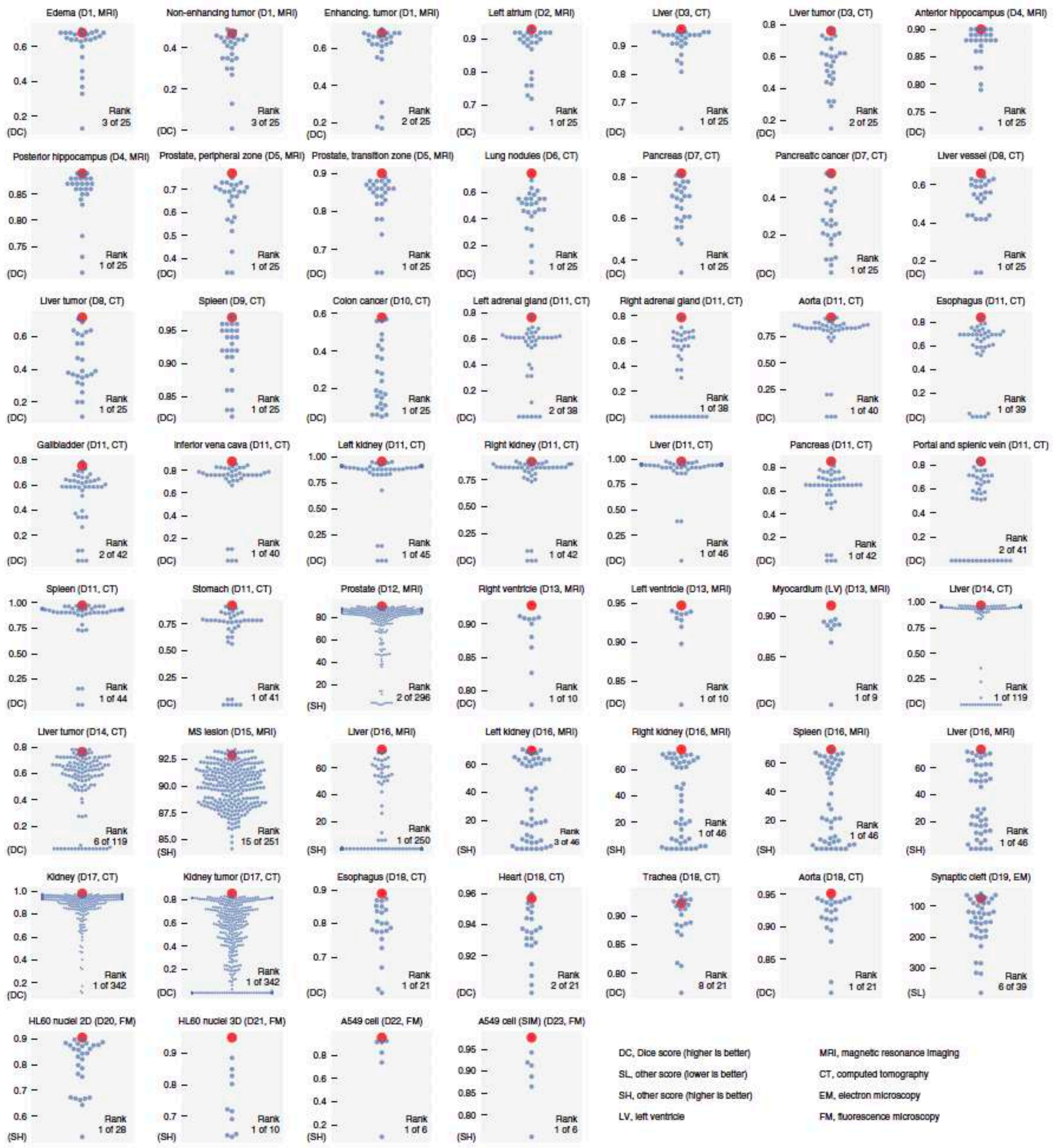


Fig. 3 | nnU-Net outperforms most specialized deep learning pipelines. Quantitative results from all international challenges that nnU-Net competed in. For each segmentation task, results achieved by nnU-Net are highlighted in red; competing teams are shown in blue. For each segmentation task, nnU-Net's rank as well as the total number of competing algorithms is displayed in the bottom-right corner of each plot. Note that for the CHAOS challenge (D16), we only participated in two of the five subtasks for reasons outlined in section 9 of Supplementary Note 6. The Cell Tracking Challenge leaderboard (D20–D23) was last accessed on 30 July 2020; all remaining leaderboards were last accessed on 12 December 2019. SIM, simulated.

U-Net Summary

[Submitted on 15 Apr 2024]

nnU-Net Revisited: A Call for Rigorous Validation in 3D Medical Image Segmentation

Fabian Isensee, Tassilo Wald, Constantin Ulrich, Michael Baumgartner, Saikat Roy, Klaus Maier-Hein, Paul F. Jaeger

The release of nnU-Net marked a paradigm shift in 3D medical image segmentation, demonstrating that a properly configured U-Net architecture could still achieve state-of-the-art results. Despite this, the pursuit of novel architectures, and the respective claims of superior performance over the U-Net baseline, continued. In this study, we demonstrate that many of these recent claims fail to hold up when scrutinized for common validation shortcomings, such as the use of inadequate baselines, insufficient datasets, and neglected computational resources. By meticulously avoiding these pitfalls, we conduct a thorough and comprehensive benchmarking of current segmentation methods including CNN-based, Transformer-based, and Mamba-based approaches. In contrast to current beliefs, we find that the recipe for state-of-the-art performance is 1) employing CNN-based U-Net models, including ResNet and ConvNeXt variants, 2) using the nnU-Net framework, and 3) scaling models to modern hardware resources. These results indicate an ongoing innovation bias towards novel architectures in the field and underscore the need for more stringent validation standards in the quest for scientific progress.

Subjects: **Computer Vision and Pattern Recognition (cs.CV)**
Cite as: arXiv:2404.09556 **[cs.CV]**
(or arXiv:2404.09556v1 **[cs.CV]** for this version)
https://doi.org/10.48550/arXiv.2404.09556

U-Net Summary

U-Net Summary

U-Net advantages

- Flexible use for any pixel-wise prediction task
- Doesn’t contain any fully connected layers
- Faster than sliding-window
- Powerful segmentation tool also in scenarios with limited data
- Competitive accuracy (given proper training, dataset and training time)
- Achieves best performance in a broad range of biomedical applications

U-Net Summary

U-Net advantages

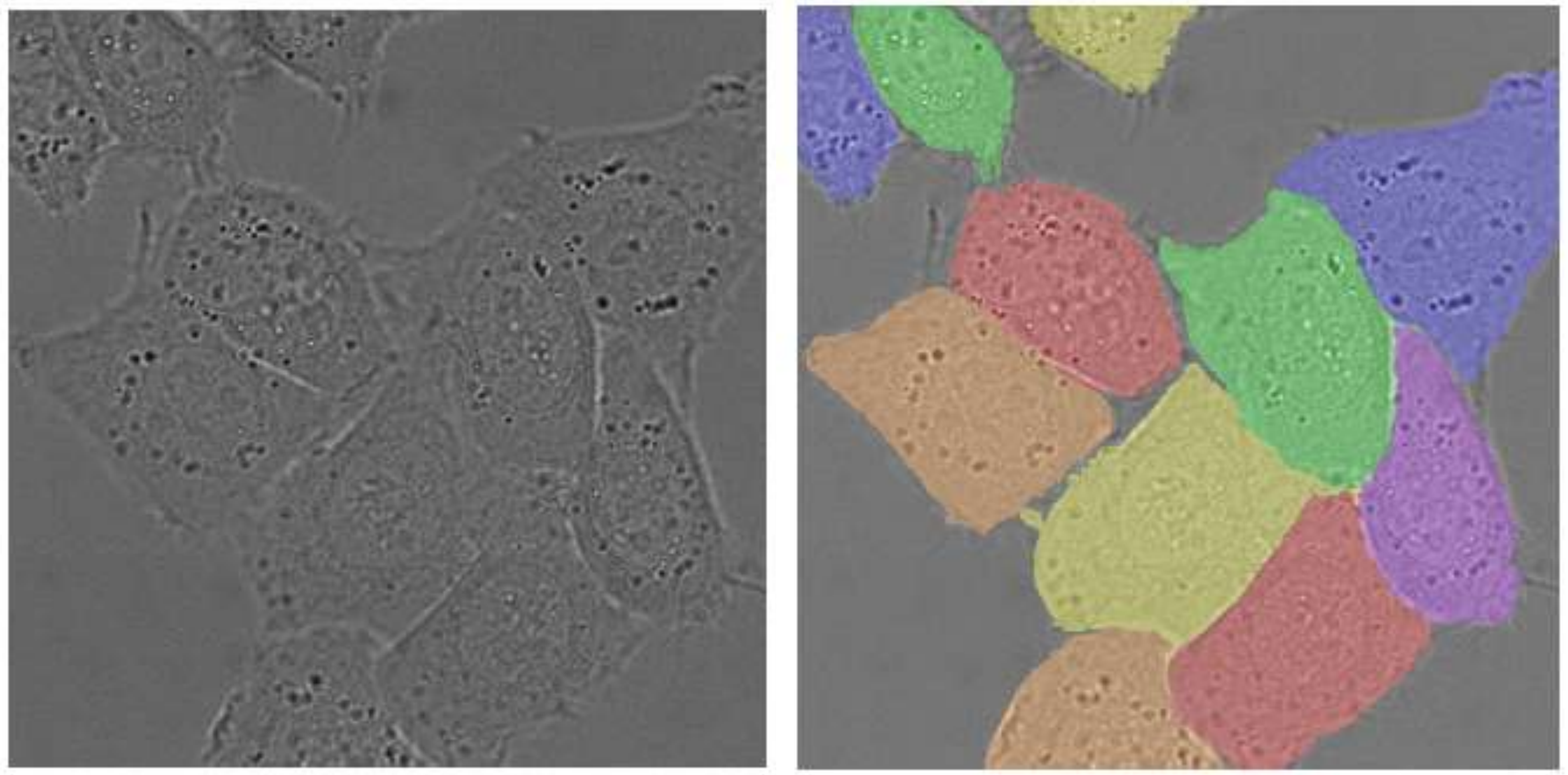
- Flexible use for any pixel-wise prediction task
- Doesn’t contain any fully connected layers
- Faster than sliding-window
- Powerful segmentation tool also in scenarios with limited data
- Competitive accuracy (given proper training, dataset and training time)
- Achieves best performance in a broad range of biomedical applications

U-Net disadvantages

- Larger images need lots of GPU memory
- Takes significant amount of time to train
- Pre-trained models not widely available (but coming up)

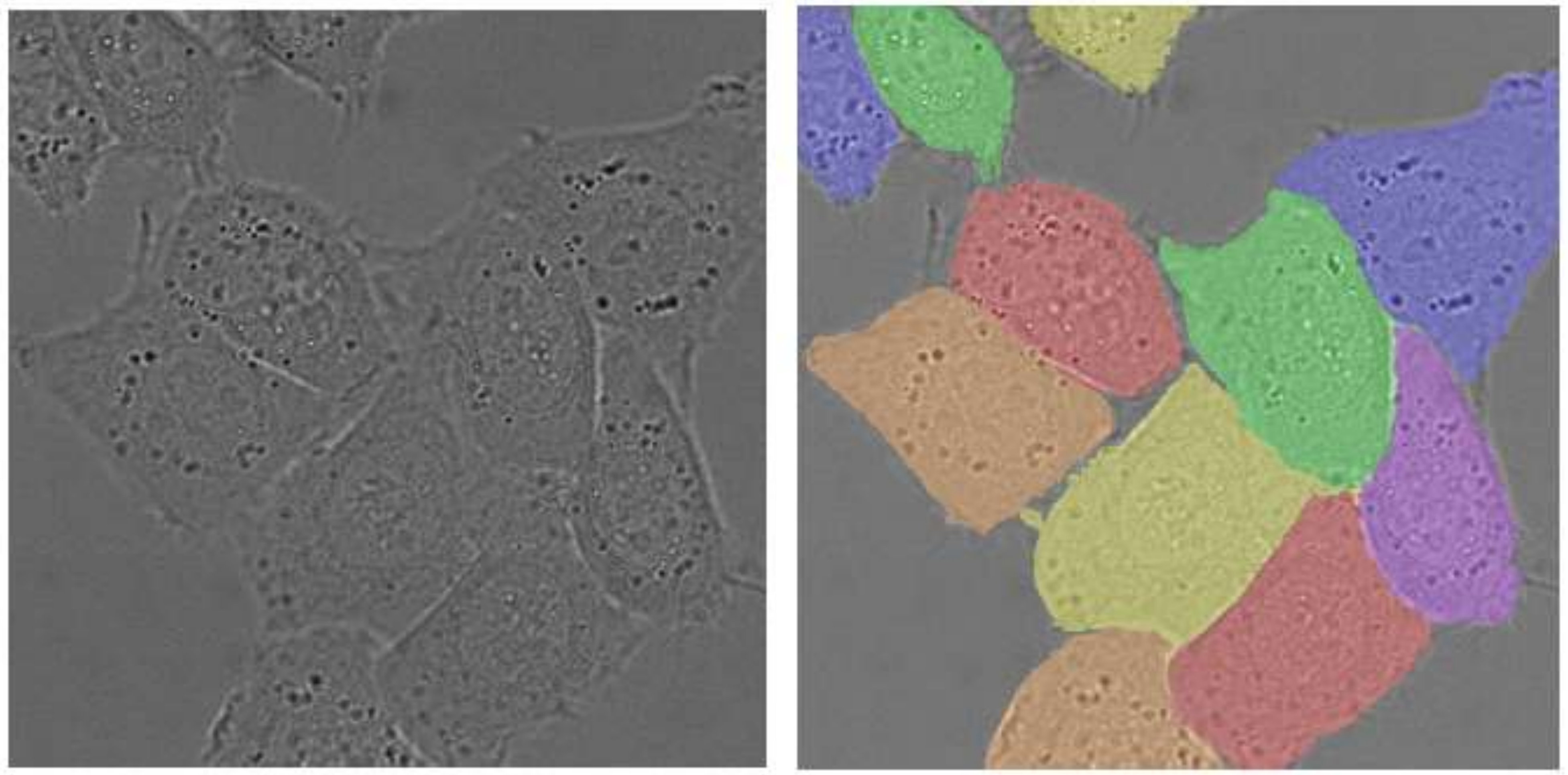
CNNs for Instance Segmentation

Assign one instance label to each pixel in an image



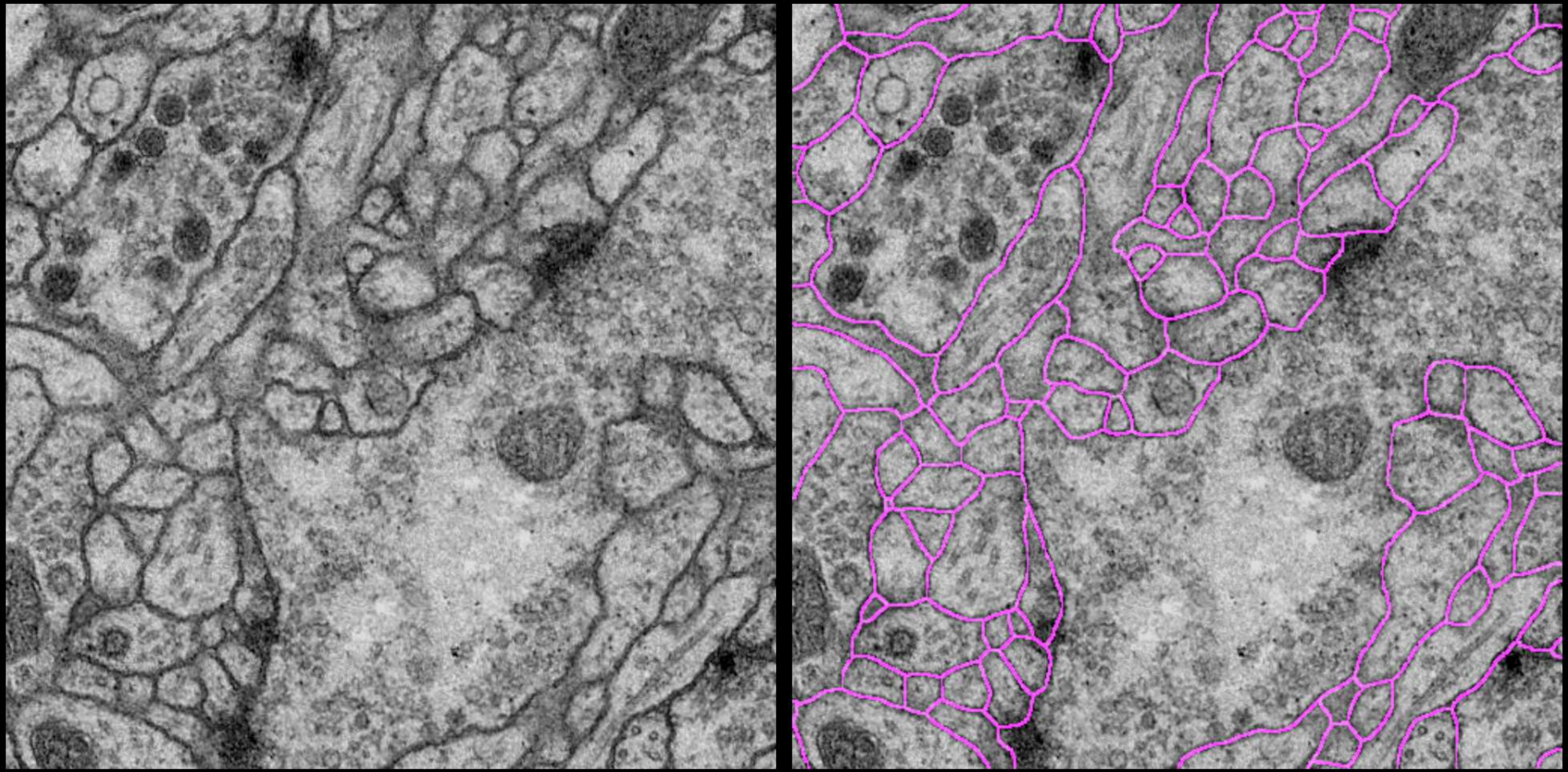
CNNs for Instance Segmentation

Assign one instance label to each pixel in an image



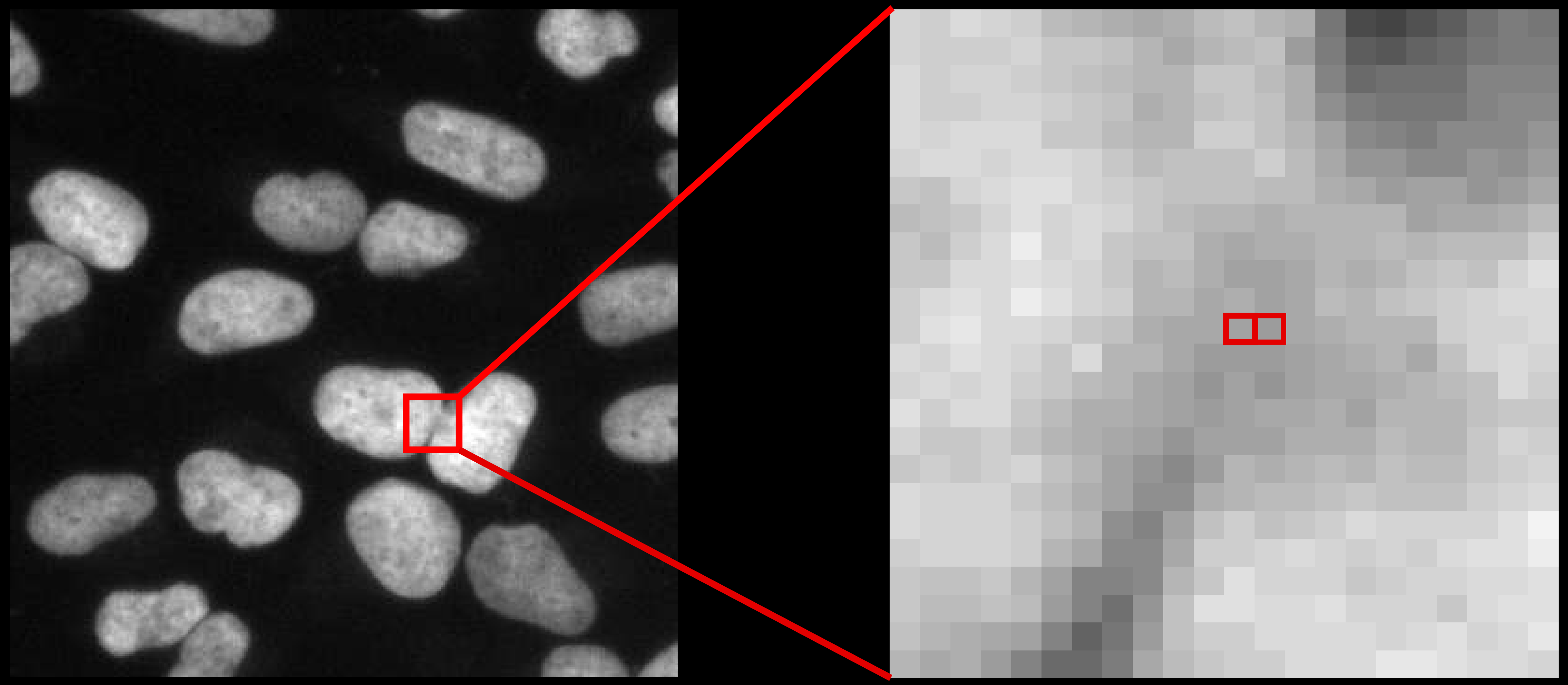
CNN-based “Proposal-free” Instance Segmentation

Lead on ISBI 2012 EM Segmentation Challenge in 2015:
U-Net for boundary pixel classification



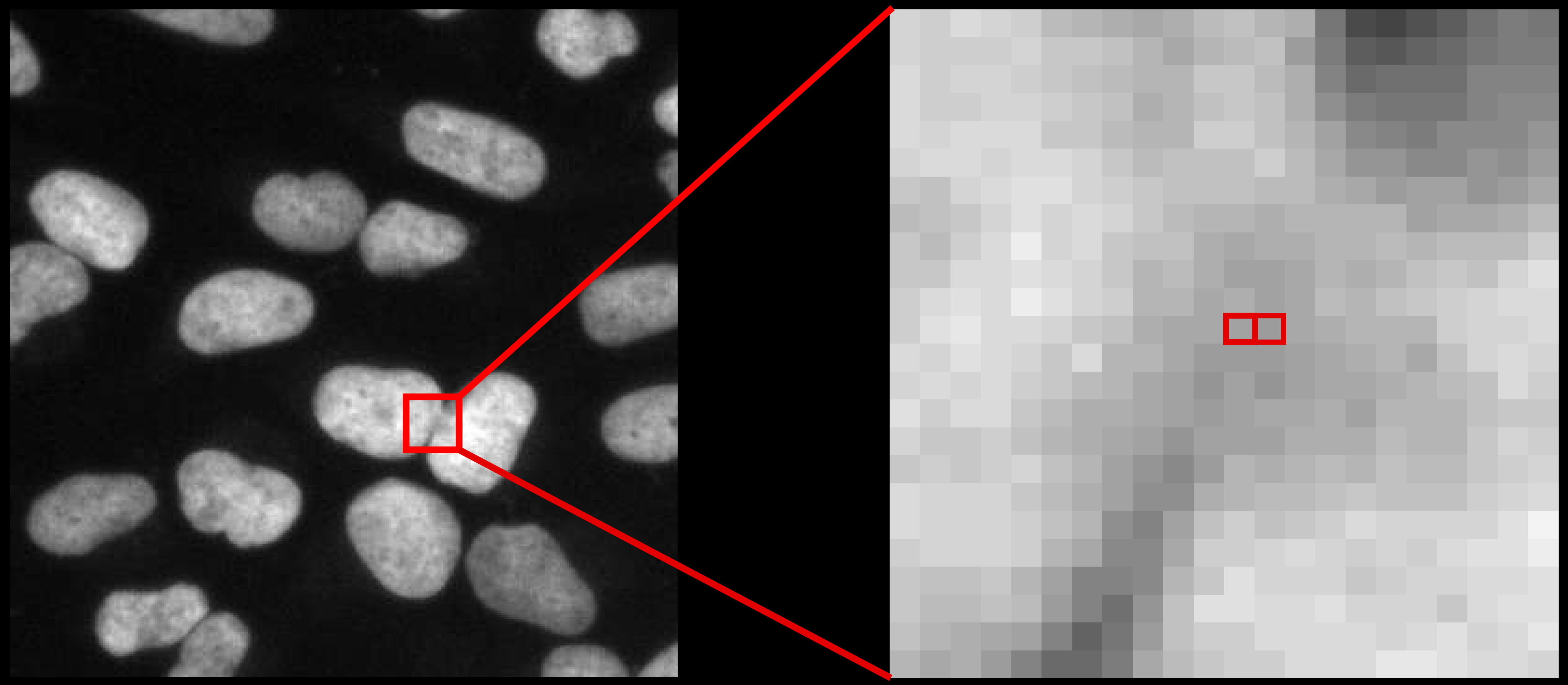
Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. MICCAI 2015 (pp. 234-241).

CNN-based Proposal-free Instance Segmentation



Instances with Background:

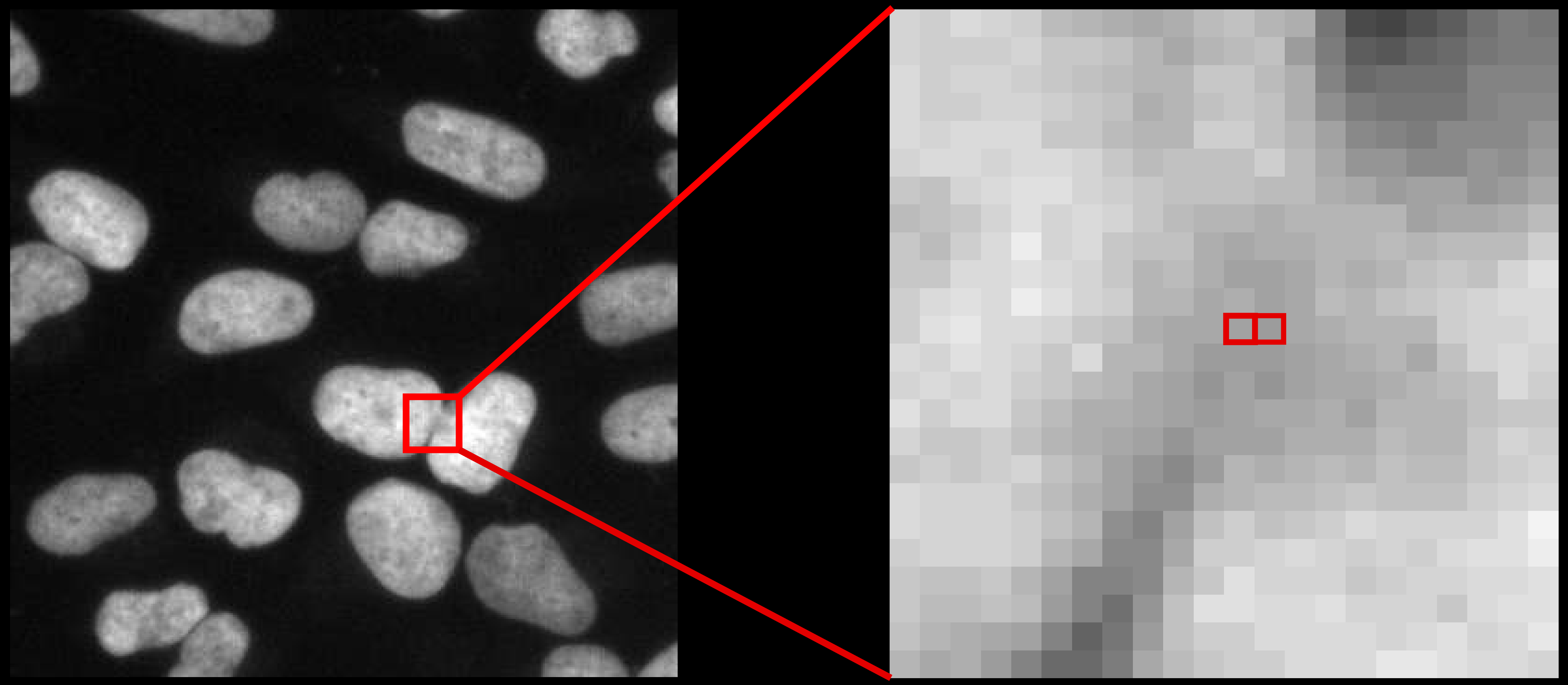
CNN-based Proposal-free Instance Segmentation



Instances with Background:

Two labels: Fg, Bg

CNN-based Proposal-free Instance Segmentation

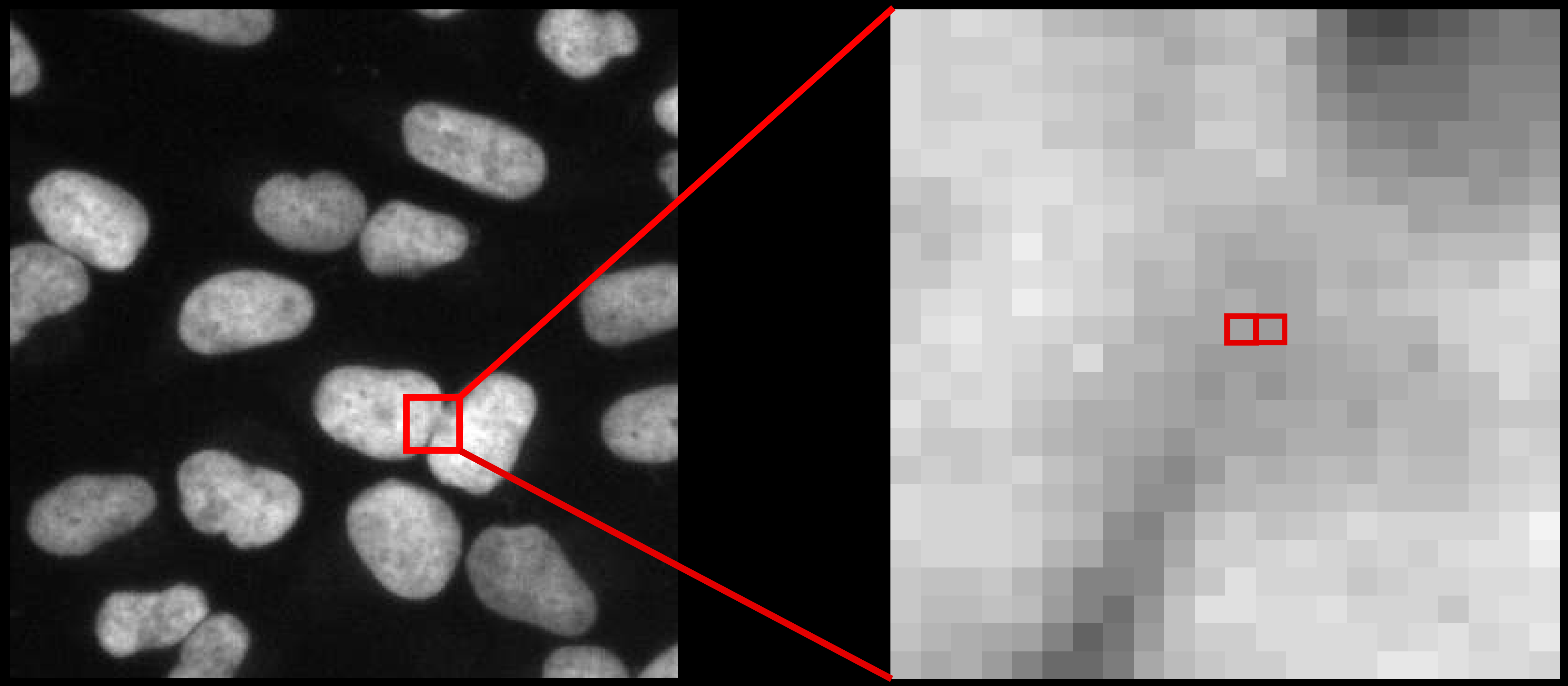


Instances with Background:

Two labels: Fg, Bg

Three labels: Fg, Bg, Boundary

CNN-based Proposal-free Instance Segmentation



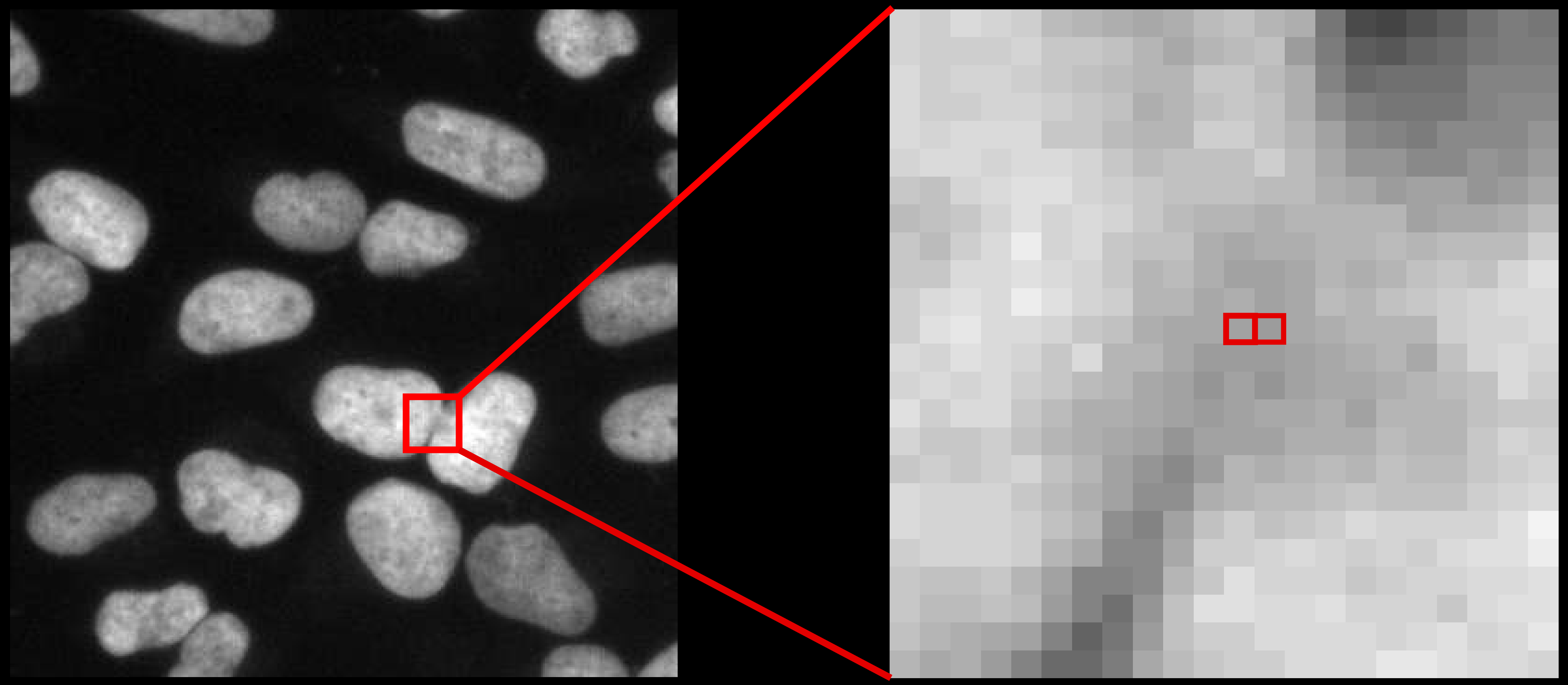
Instances with Background:

Two labels: Fg, Bg

Three labels: Fg, Bg, Boundary

Problem:

CNN-based Proposal-free Instance Segmentation



Instances with Background:

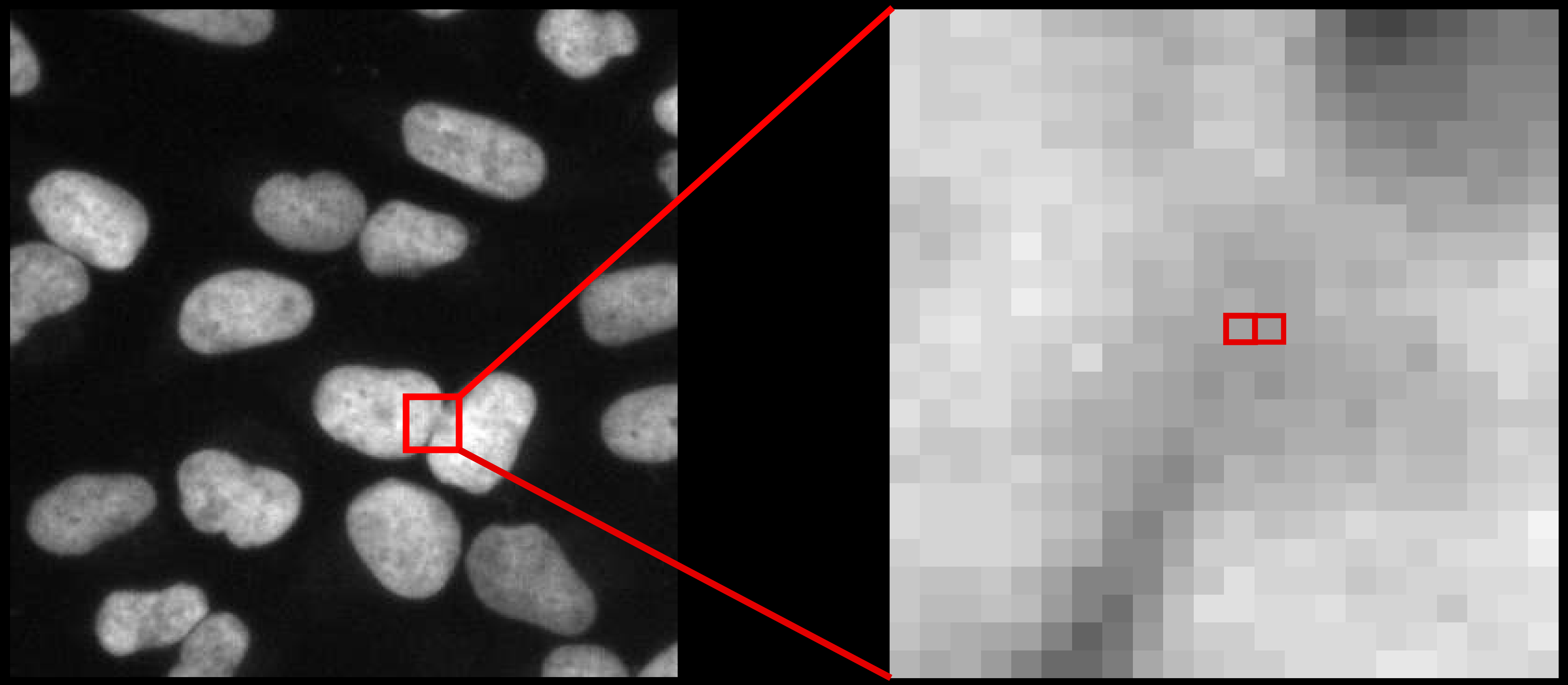
Two labels: Fg, Bg

Three labels: Fg, Bg, Boundary

Problem:

Tiny change in prediction / loss can cause huge (topological) change in segmentation

CNN-based Proposal-free Instance Segmentation



Instances with Background:

Two labels: Fg, Bg

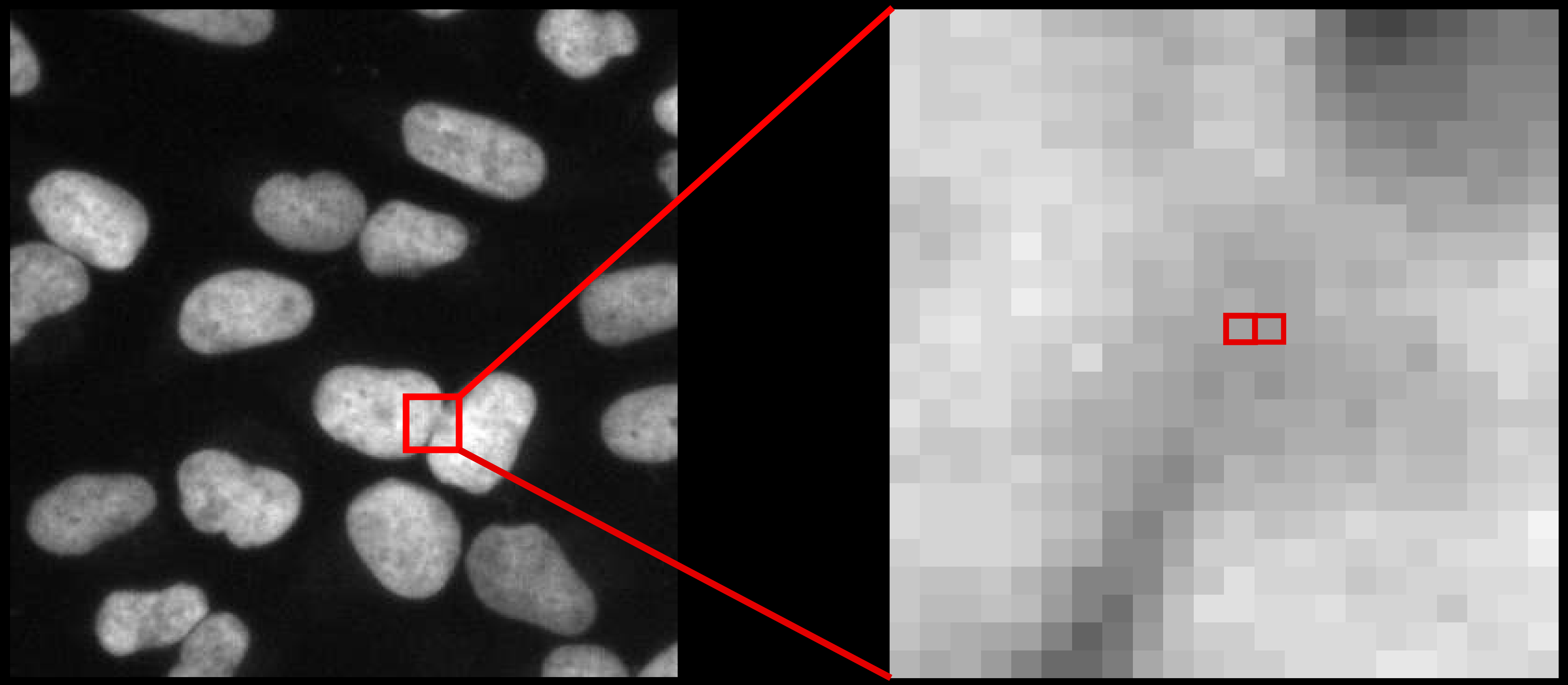
Three labels: Fg, Bg, Boundary

Problem:

Tiny change in prediction / loss can cause huge (topological) change in segmentation

Funke, J., Tschopp, F., Grisaitis, W., Sheridan, A., Singh, C., Saalfeld, S., & Turaga, S. C. (2018). Large scale image segmentation with structured loss based deep learning for connectome reconstruction. IEEE transactions on pattern analysis and machine intelligence, 41(7), 1669-1680.

CNN-based Proposal-free Instance Segmentation



Instances with Background:

Two labels: Fg, Bg

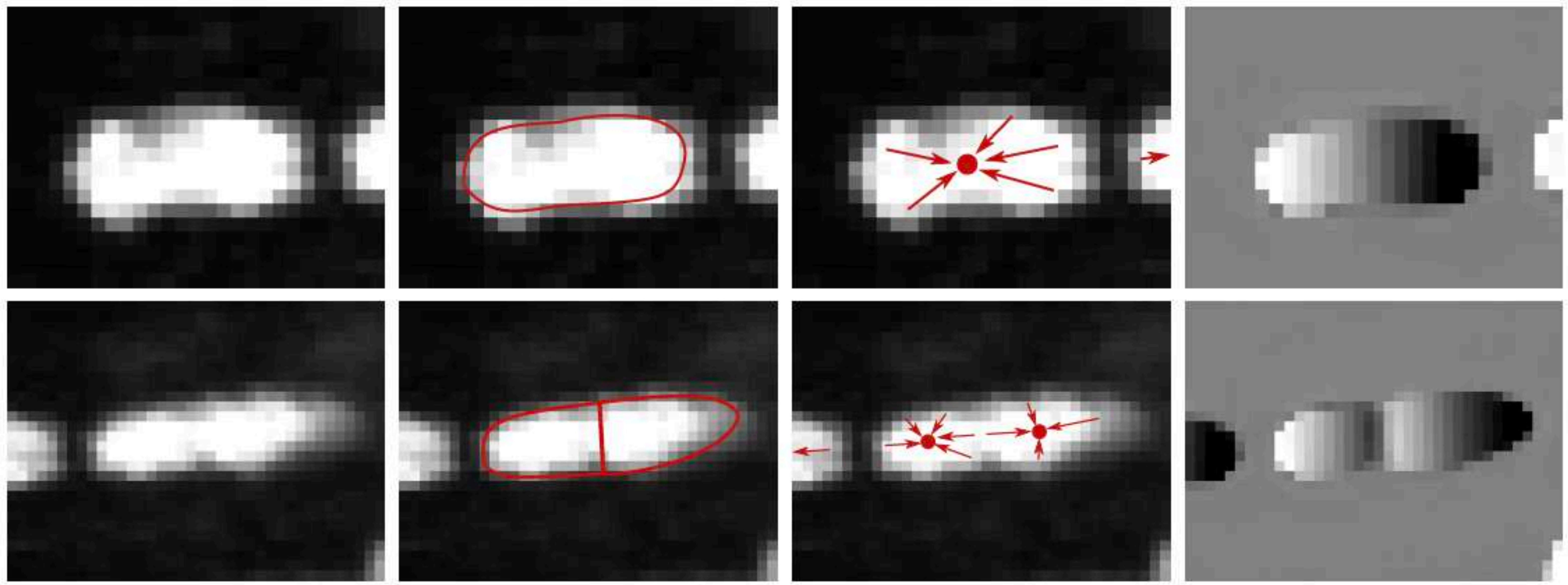
Three labels: Fg, Bg, Boundary

Problem:

Tiny change in prediction / loss can cause huge (topological) change in segmentation

Hirsch, Kainmueller: *An Auxiliary Task for Learning Nuclei Segmentation in 3D Microscopy Images.* MIDL 2020

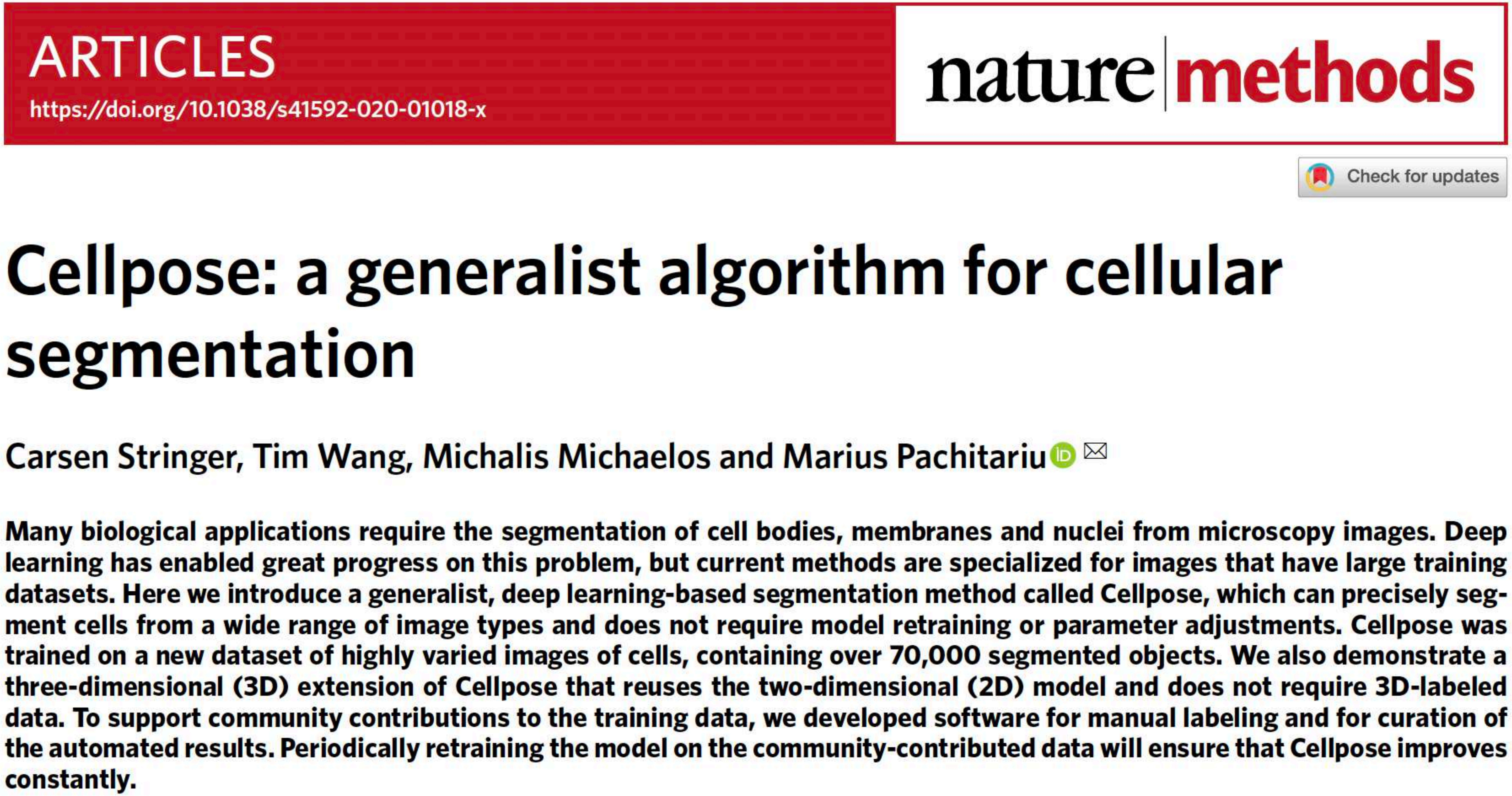
CNN-based Proposal-free Instance Segmentation



(A) Exemplary nuclei (B) Boundary label (C) Center point vectors (D) Prediction

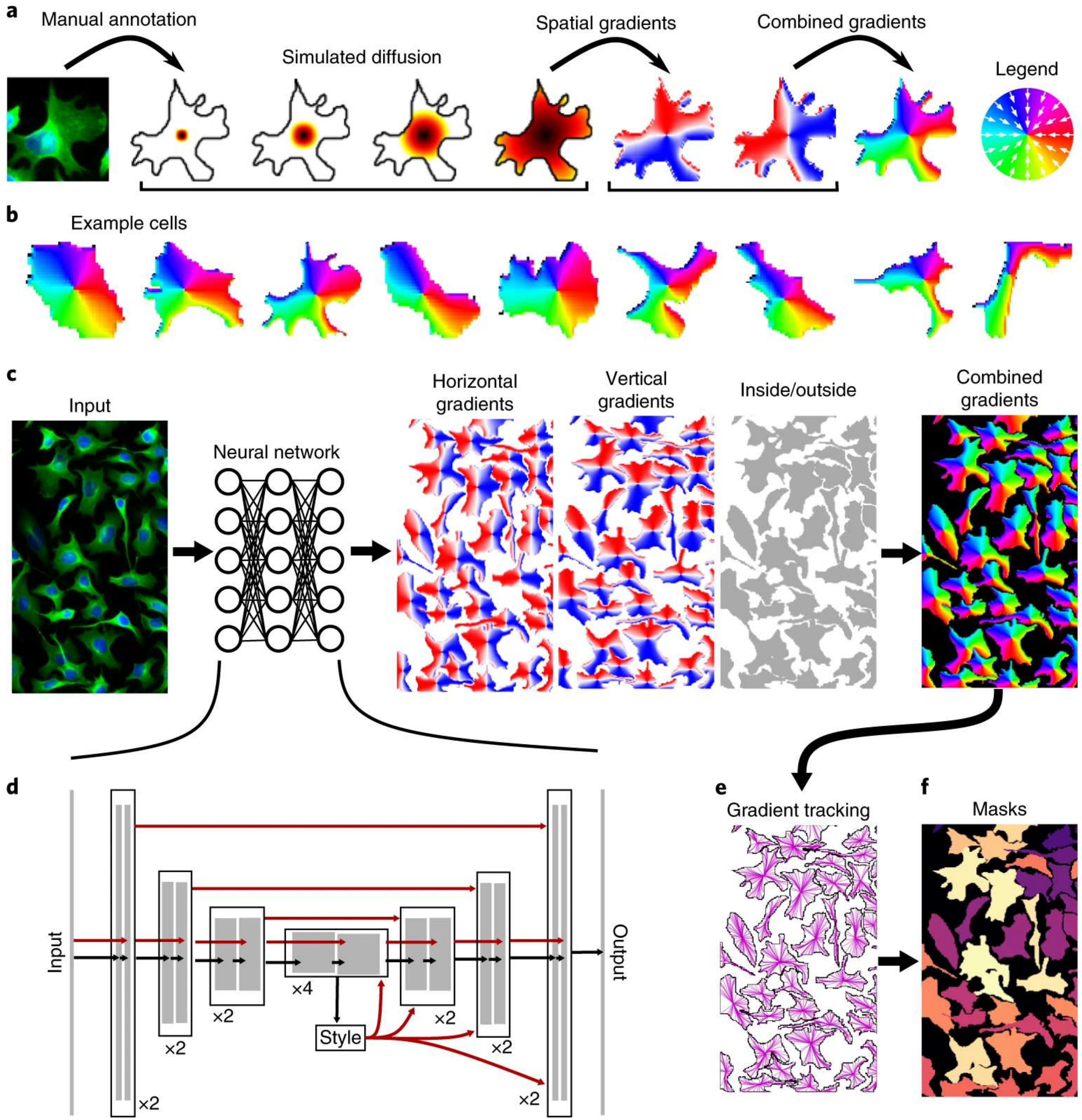
Hirsch, Kainmueller: *An Auxiliary Task for Learning Nuclei Segmentation in 3D Microscopy Images.* MIDL 2020

CNN-based Proposal-free Instance Segmentation



published Dec. 2020

CNN-based Proposal-free Instance Segmentation



a, Procedure for transforming manually annotated masks into a vector flow representation that can be predicted by a neural network. A simulated diffusion process starting from the center of the mask is used to derive spatial gradients that point toward the center of the cell, potentially indirectly around corners. The x and y gradients are combined in a single direction from 0° to 360°. **b**, Example spatial gradients for cells from the training dataset. **c**, A neural network is trained to predict the horizontal and vertical gradients, as well as whether a pixel belongs to any cell. The three predicted maps are combined into a gradient vector field. **d**, The details of the neural network that contains a standard backbone U-Net[3] to downsample and then upsample the feature maps, with skip connections between layers of the same size and global skip connections from the image styles, computed at the lowest resolution, to all successive computations. **e**, At test time, the predicted gradient vector fields are used to construct a dynamical system with fixed points whose basins of attraction represent the predicted masks. Informally, every pixel 'tracks the gradients' toward their eventual fixed point. **f**, All the pixels that converge to the same fixed point are assigned to the same mask.

CNN-based Proposal-free Instance Segmentation

Analysis | Published: 26 March 2024

The multimodality cell segmentation challenge: toward universal solutions

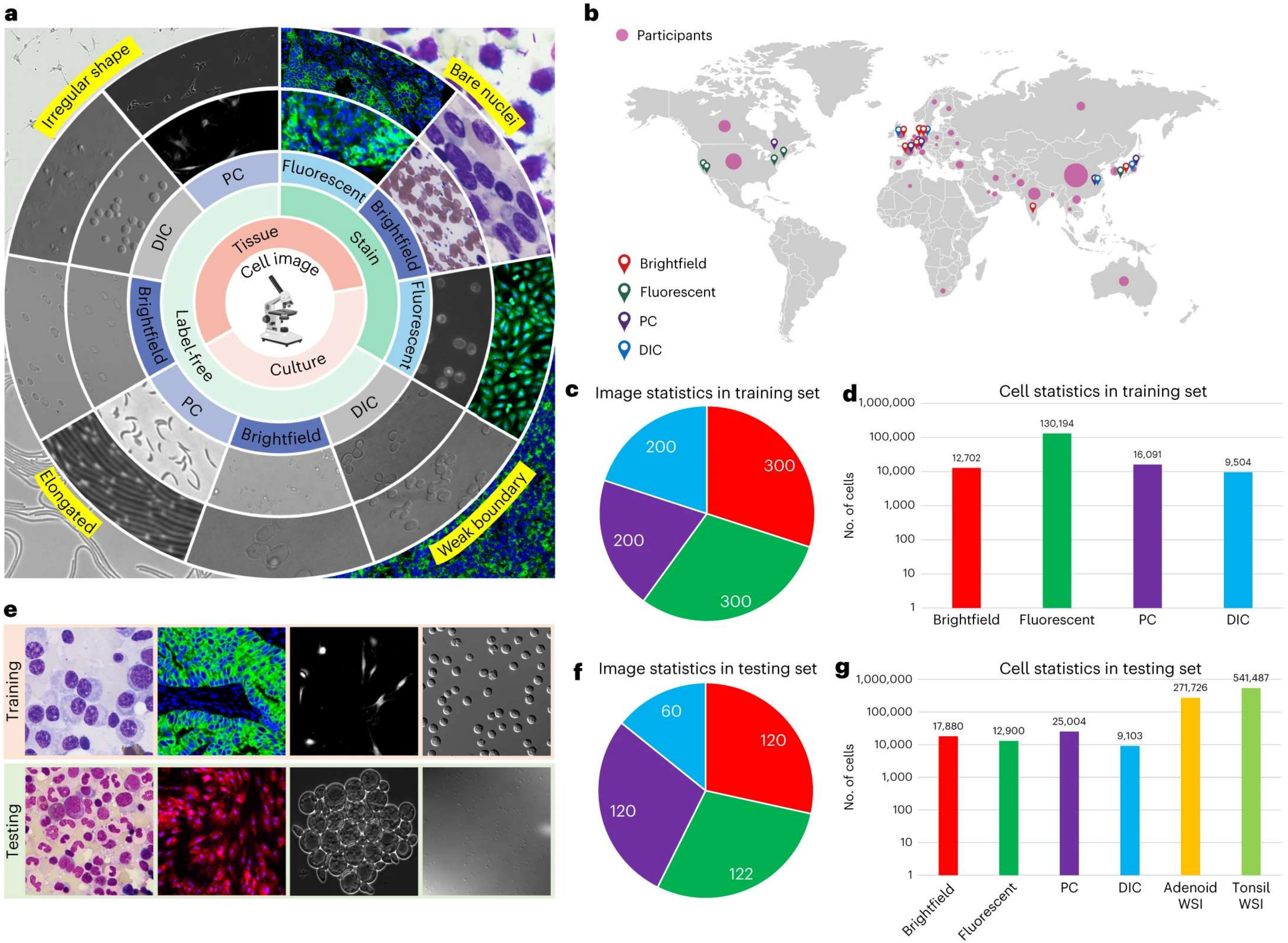
Jun Ma, Ronald Xie, Shamini Ayyadhury, Cheng Ge, Anubha Gupta, Ritu Gupta, Song Gu, Yao Zhang, Gihun Lee, Joonkee Kim, Wei Lou, Haofeng Li, Eric Upschulte, Timo Dickscheid, José Guilherme de Almeida, Yixin Wang, Lin Han, Xin Yang, Marco Labagnara, Vojislav Gligorovski, Maxime Scheder, Sahand Jamal Rahi, Carly Kempster, Alice Pollitt, ... Bo Wang + Show authors

Nature Methods (2024) | Cite this article

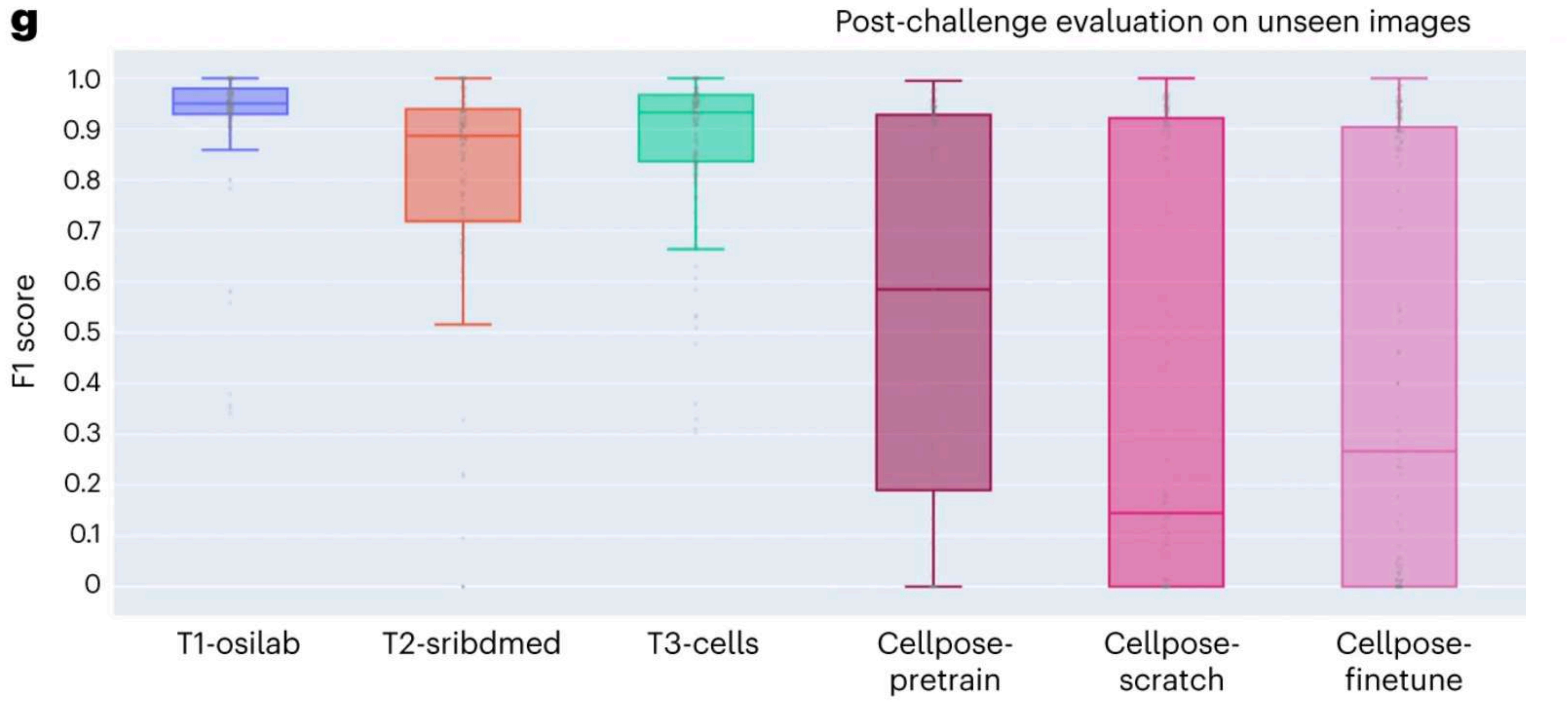
Abstract

Cell segmentation is a critical step for quantitative single-cell analysis in microscopy images. Existing cell segmentation methods are often tailored to specific modalities or require manual interventions to specify hyper-parameters in different experimental settings. Here, we present a multimodality cell segmentation benchmark, comprising more than 1,500 labeled images derived from more than 50 diverse biological experiments. The top participants developed a Transformer-based deep-learning algorithm that not only exceeds existing methods but can also be applied to diverse microscopy images across imaging platforms and tissue types without manual parameter adjustments. This benchmark and the improved algorithm offer promising avenues for more accurate and versatile cell analysis in microscopy imaging.

CNN-based Proposal-free Instance Segmentation



CNN-based Proposal-free Instance Segmentation



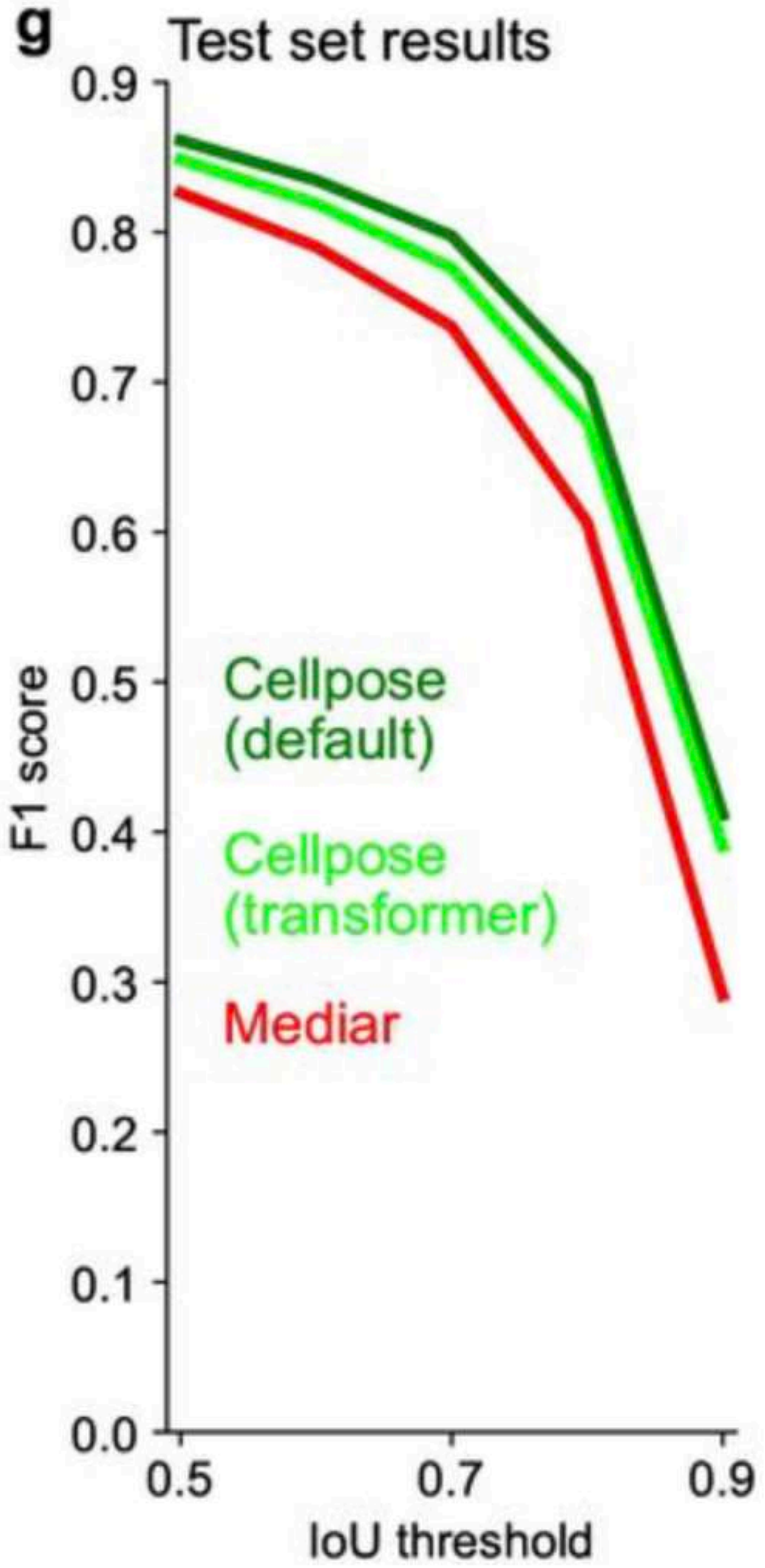
CNN-based Proposal-free Instance Segmentation

Transformers do not outperform Cellpose

Carsen Stringer[†], Marius Pachitariu[†]
HHMI Janelia Research Campus, Ashburn, VA, USA
[†] correspondence to (stringerc, pachitarium) @ janelia.hhmi.org

In a recent publication, Ma et al [1] claim that a transformer-based cellular segmentation method called Mediar [2] — which won a Neurips challenge — outperforms Cellpose [3] (0.897 vs 0.543 median F1 score). Here we show that this result was obtained by artificially impairing Cellpose in multiple ways. When we removed these impairments, Cellpose outperformed Mediar (0.861 vs 0.826 median F1 score on the updated test set). To further investigate the performance of transformers for cellular segmentation, we replaced the Cellpose backbone with a transformer. The transformer-Cellpose model also did not outperform the standard Cellpose (0.848 median F1 test score). Our results suggest that transformers do not advance the state-of-the-art in cellular segmentation.

CNN-based Proposal-free Instance Segmentation



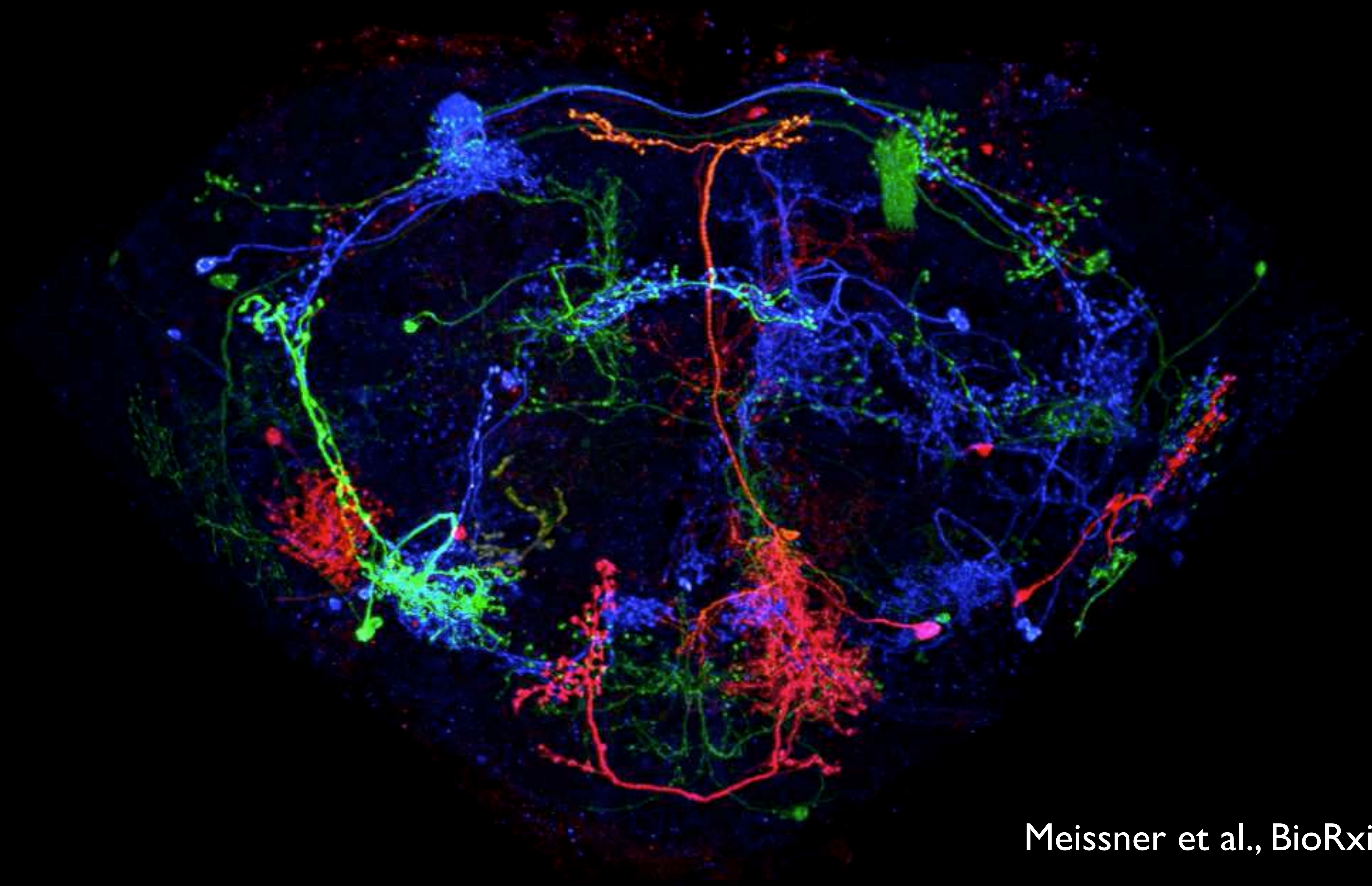
CNN-based Proposal-free Instance Segmentation

CNN-based Proposal-free Instance Segmentation

Flaws in *Cell Segmentation Challenge* Benchmarking of CellPose

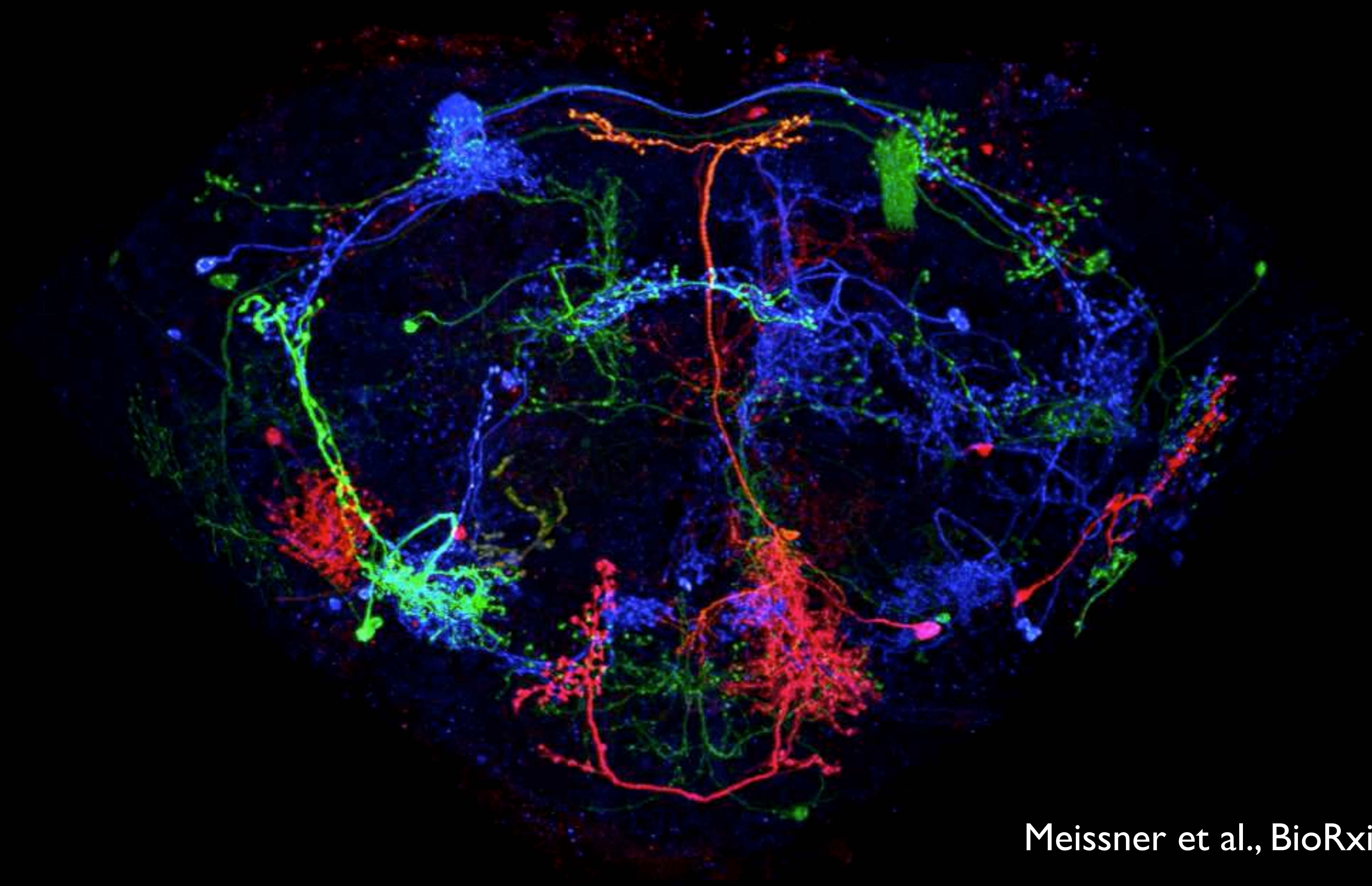
1. CellPose trained on grayscale images instead of full-channel images
2. Cell size normalization pipeline run at training time but not at test time
3. Training only on challenge training data instead of incl. CellPose dataset
4. CellPose run without test-time augmentations

Fluorescence Microscopy Data of Fly Brains



Meissner et al., BioRxiv2020

Fluorescence Microscopy Data of Fly Brains



Meissner et al., BioRxiv2020

FISBe:
The FlyLight Instance Segmentation Benchmark Dataset

A dataset for neuron instance segmentation in 3d multicolor light microscopy data of fruit fly brains, with 30 completely- and 71 partly labeled images, altogether comprising ~600 expert-labeled neuron instances. It constitutes, to the best of our knowledge, the first publicly available multi-neuron LM dataset with pixel-wise annotations and the first real-world benchmark dataset for instance segmentation of long thin filamentous objects.

Download Dataset

Lisa Mais*, Peter Hirsch*, Claire Managan, Ramya Kandarpa, Josef Lorenz Rumberger, Annika Reinke, Lena Maier-Hein, Gudrun Ihrke, Dagmar Kainmueller

* shared first authors

[Project] [Paper] [Data] [Documentation] [Metrics] [Leaderboard] [BibTeX] [Changelog]

Convolutional Neural Networks (CNNs)

- A bit of CNN history
- CNN Layers
  - Convolutions, Pooling, Fully Connected Layers, Normalization
- Training CNNs
  - Weight initialization
  - Advanced training ingredients: Regularization: Dropout; Optimization: Adam
  - Extended training data: Augmentation, Transfer Learning
  - Sanity-checking the learning process; Hyperparameter tuning strategies
- Famous CNN architectures
- CNNs for image segmentation; CNNs for object detection

Course Outline

- Introduction to Image Analysis
- Machine Learning Basics: Linear Regression, Basic Classifiers
- Neural Networks
- **Convolutional Neural Networks**
- Recurrent Neural Networks, Transformers, Diffusion
- Model Interpretability
- Probabilistic Machine Learning
- Generative Models