Semantic Segmentation

So far, Image Classification

This image is CC0 public domain

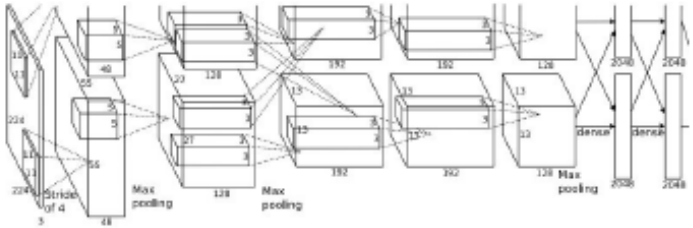


Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

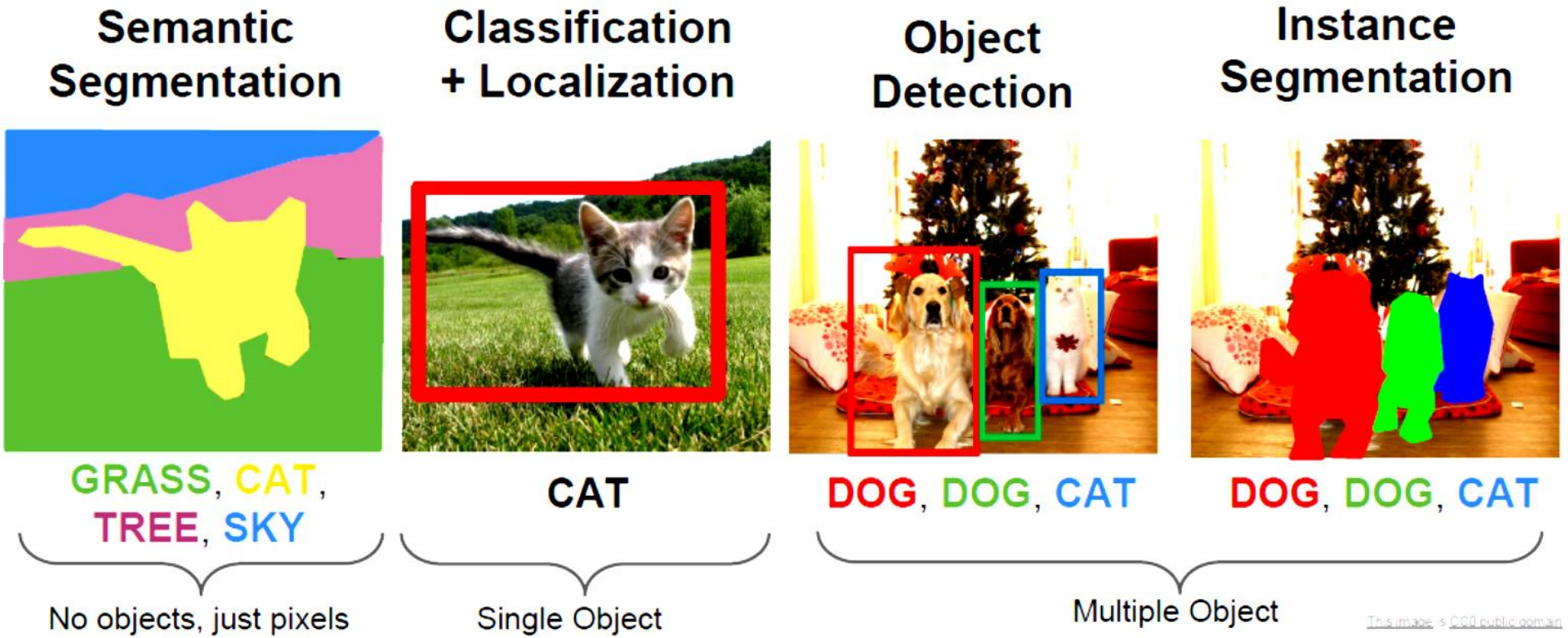
Vector:
4096

Fully-Connected:
4096 to 1000

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

Image Source:
http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Scenarios



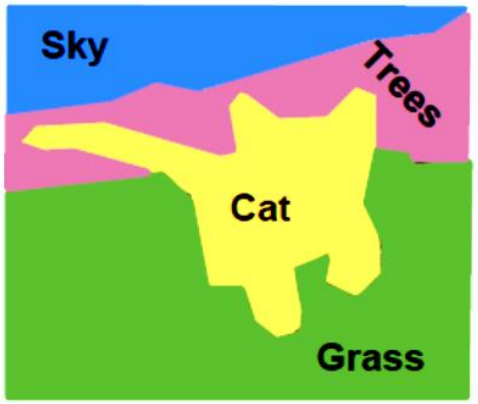
http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

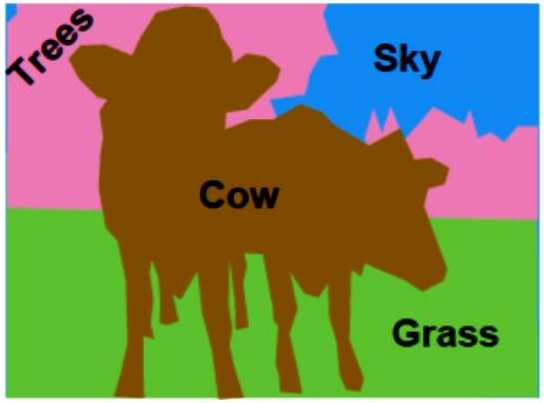
Semantic Segmentation

Semantic Segmentation

Label each pixel in the image with a category label

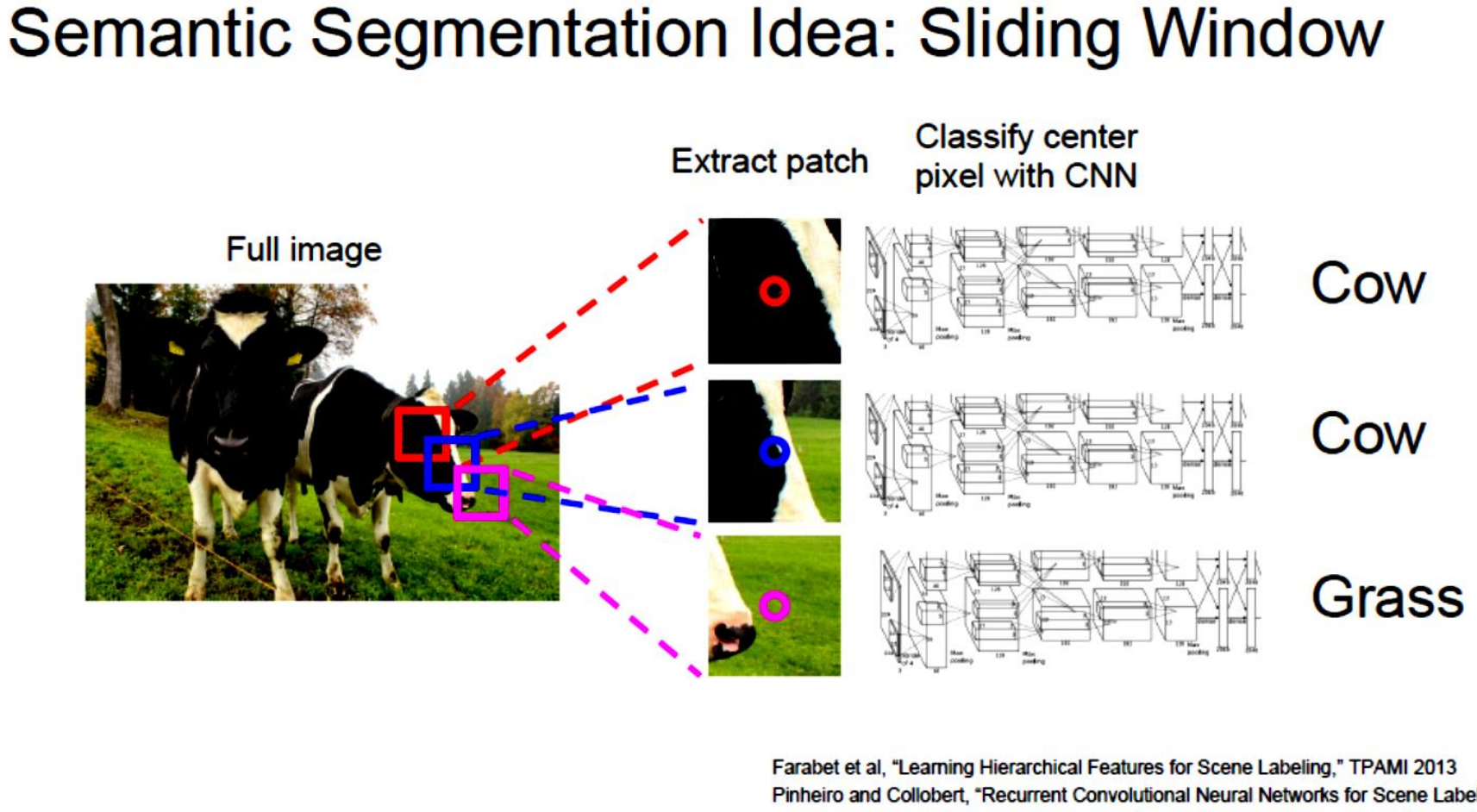
Don't differentiate instances, only care about pixels





http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

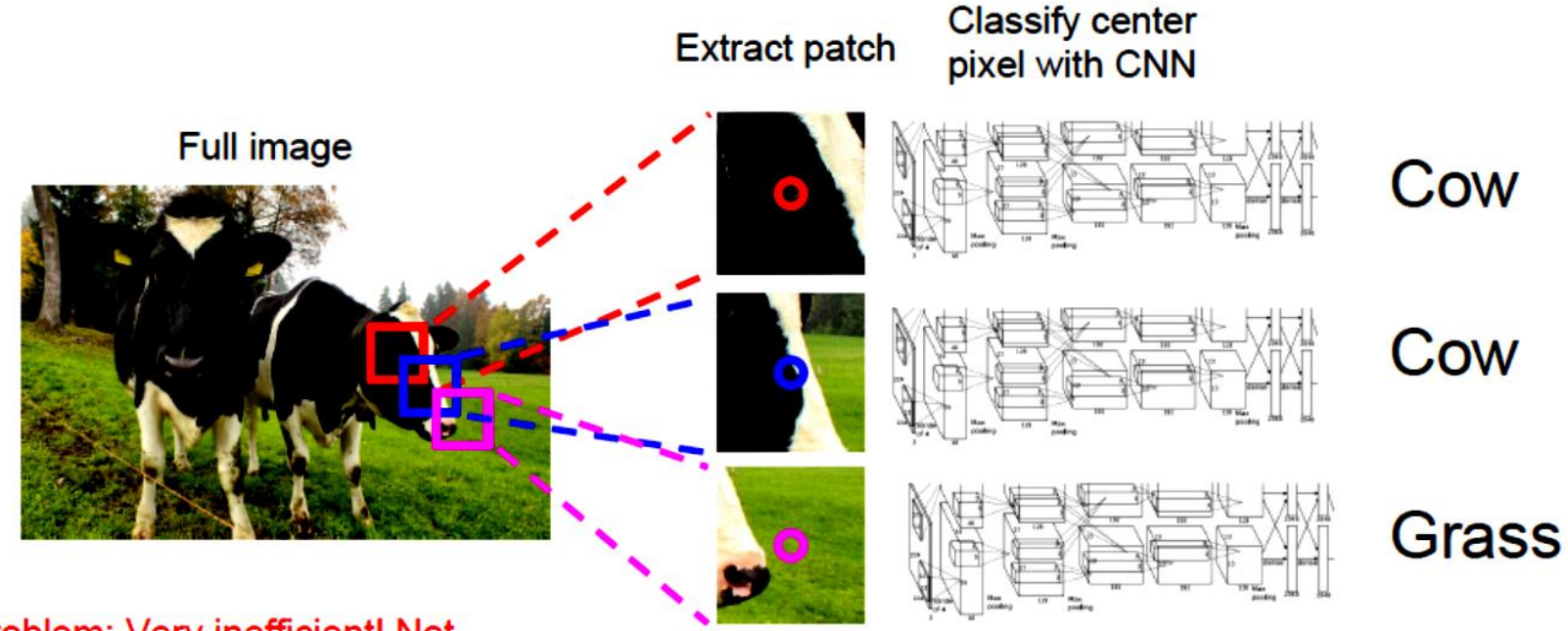
Semantic Segmentation



http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

Semantic Segmentation Idea: Sliding Window



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

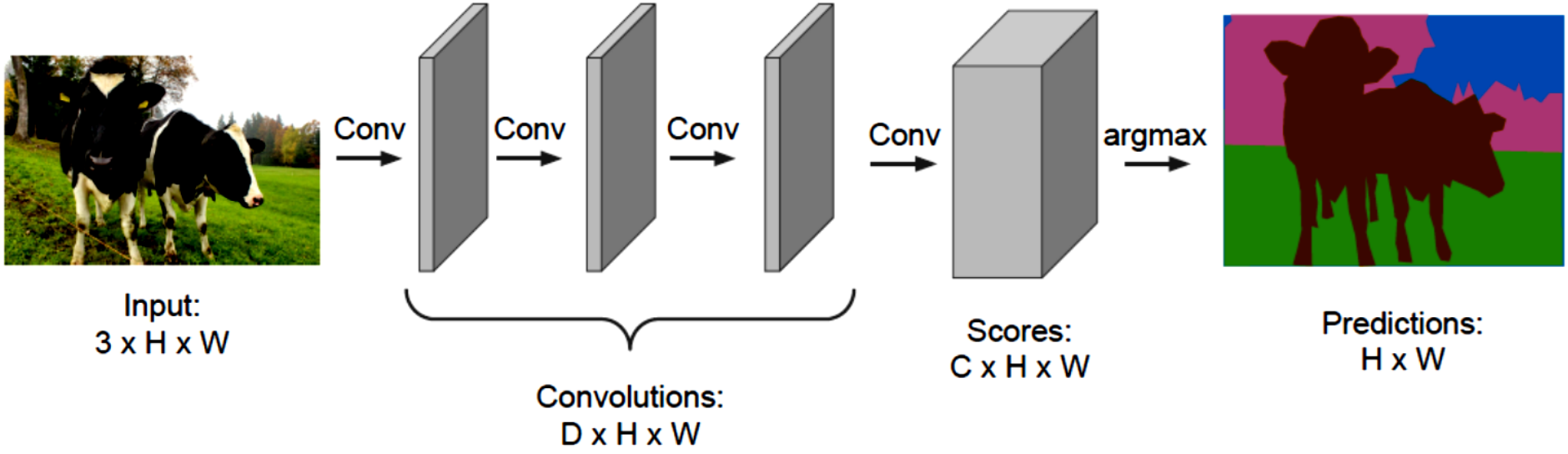
Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

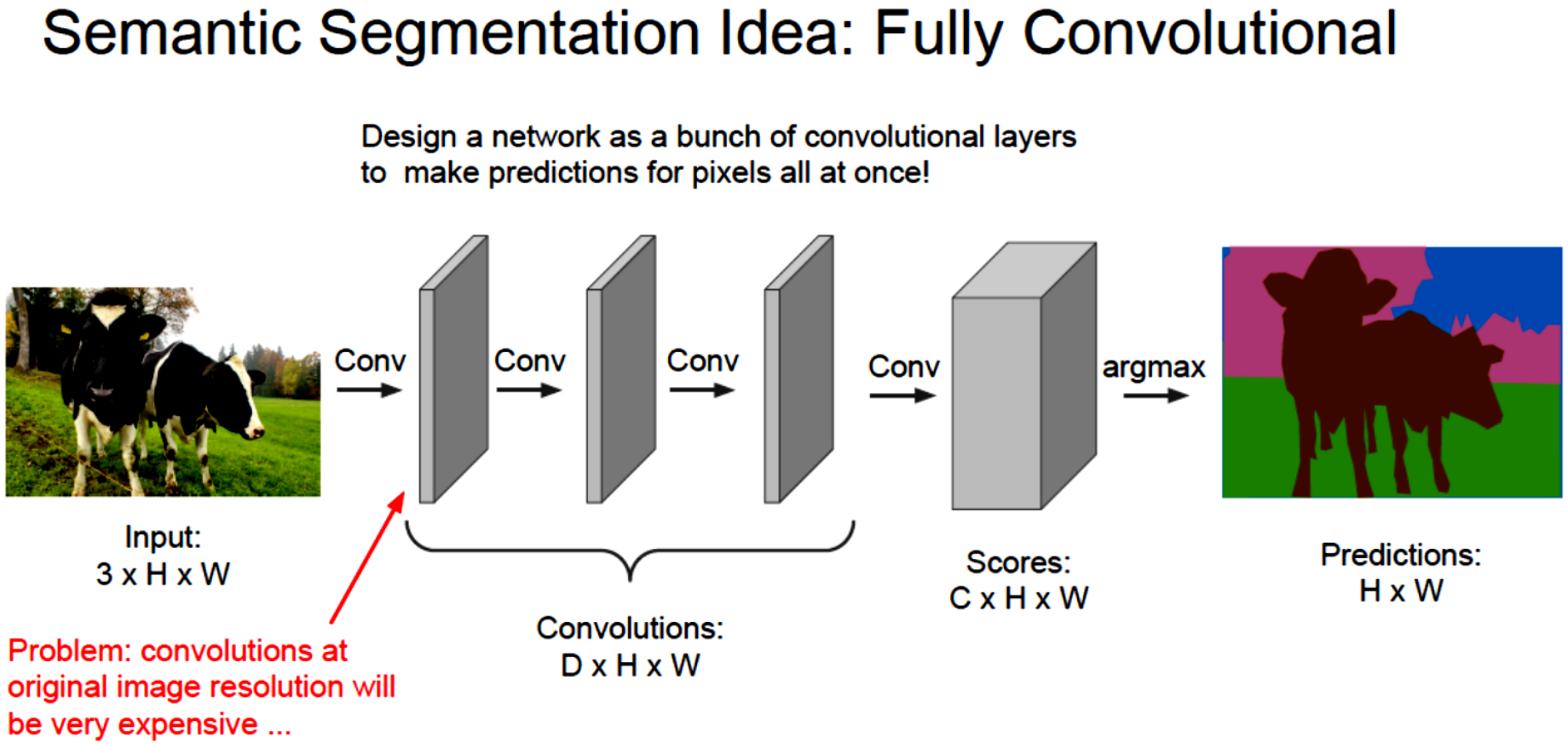
Semantic Segmentation Idea: Fully Convolutional

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



http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation



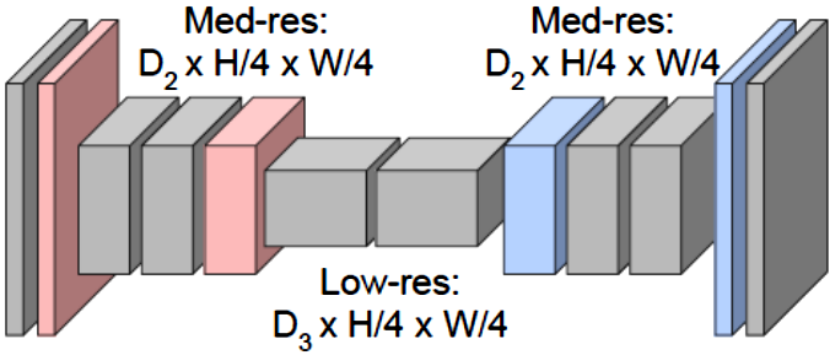
http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

Semantic Segmentation Idea: Fully Convolutional

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

Input:
3 x H x W



High-res:
D1 x H/2 x W/2

High-res:
D1 x H/2 x W/2

Predictions:
H x W

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

http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

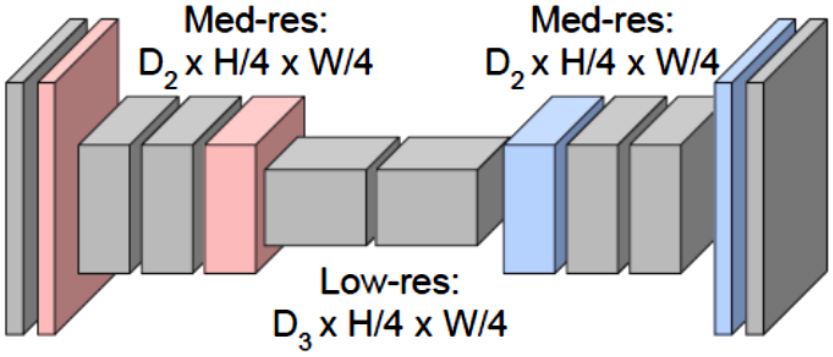
Semantic Segmentation Idea: Fully Convolutional

Downsampling: Pooling, strided convolution

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

Upsampling: ???

Input:
3 x H x W



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

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

Predictions:
H x W

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

http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

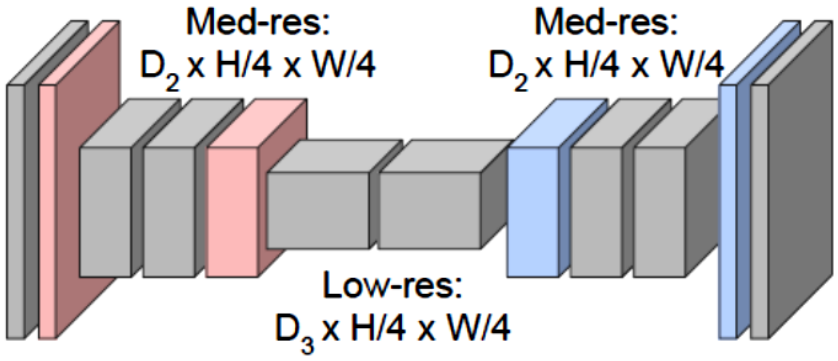
Semantic Segmentation Idea: Fully Convolutional

Downsampling:
Pooling, strided convolution

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

Upsampling:
Unpooling or strided transpose convolution

Input:
3 x H x W



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

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

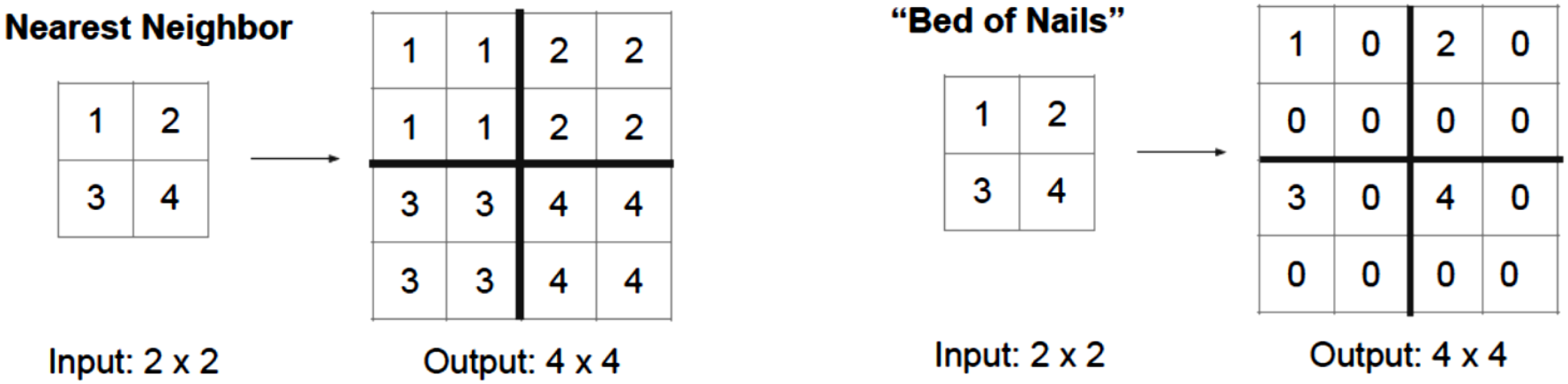
Predictions:
H x W

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

http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

In-Network upsampling: “Unpooling”



http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

Max Pooling
Remember which element was max!

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

Input: 4 x 4

5	6
7	8

Output: 2 x 2

→ . . . → Rest of the network

Max Unpooling
Use positions from pooling layer

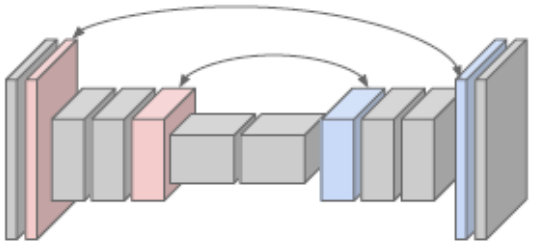
1	2
3	4

Input: 2 x 2

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

Output: 4 x 4

Corresponding pairs of downsampling and upsampling layers

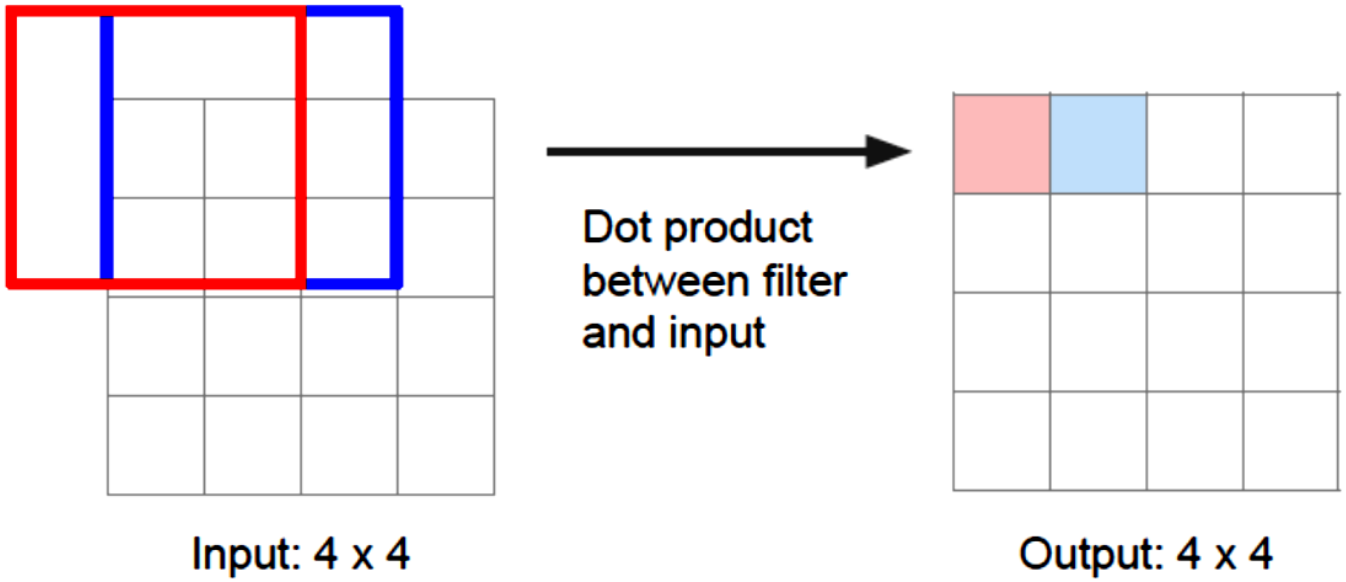


http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

Learnable Upsampling: Transpose Convolution

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

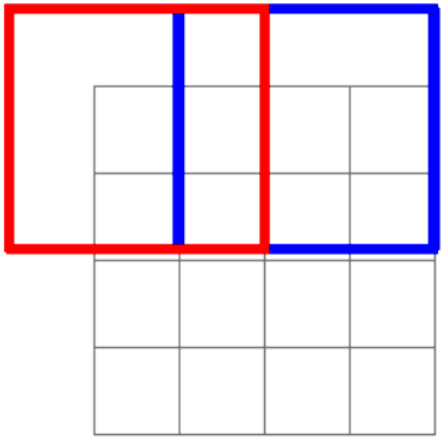


http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

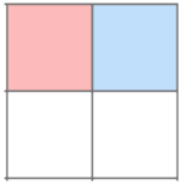
Learnable Upsampling: Transpose Convolution

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



Input: 4 x 4

Dot product between filter and input



Output: 2 x 2

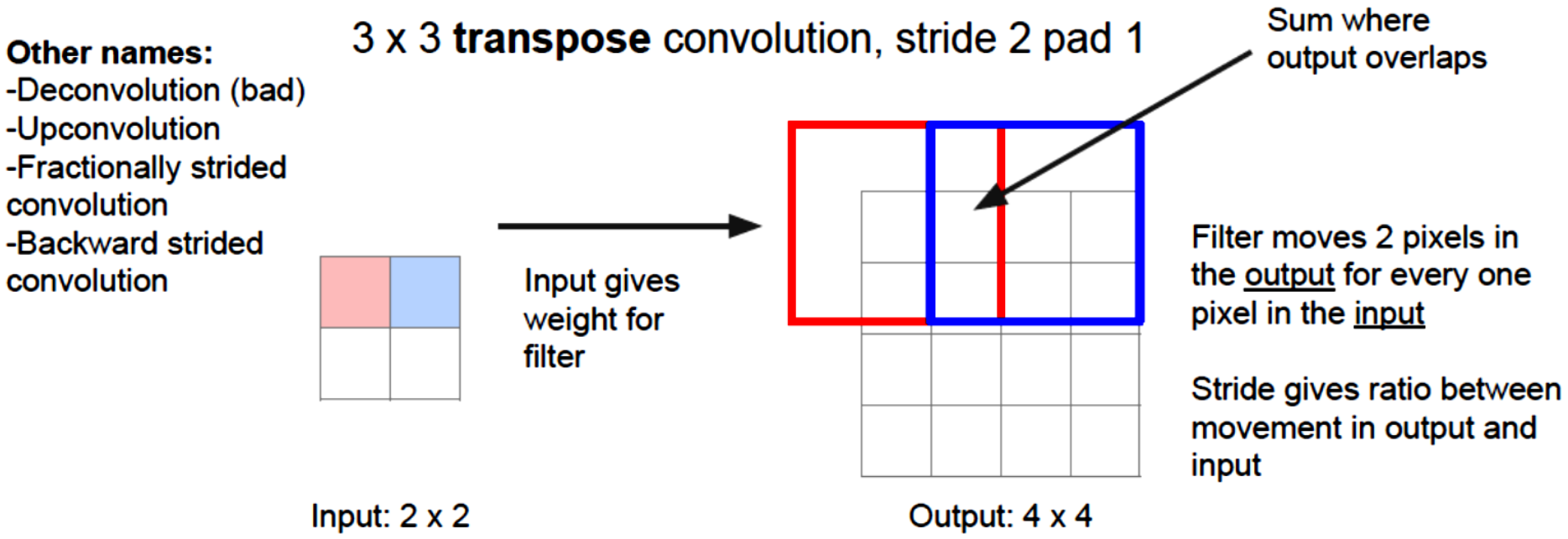
Filter moves 2 pixels in the input for every one pixel in the output

Stride gives ratio between movement in input and output

http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

Learnable Upsampling: Transpose Convolution



http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

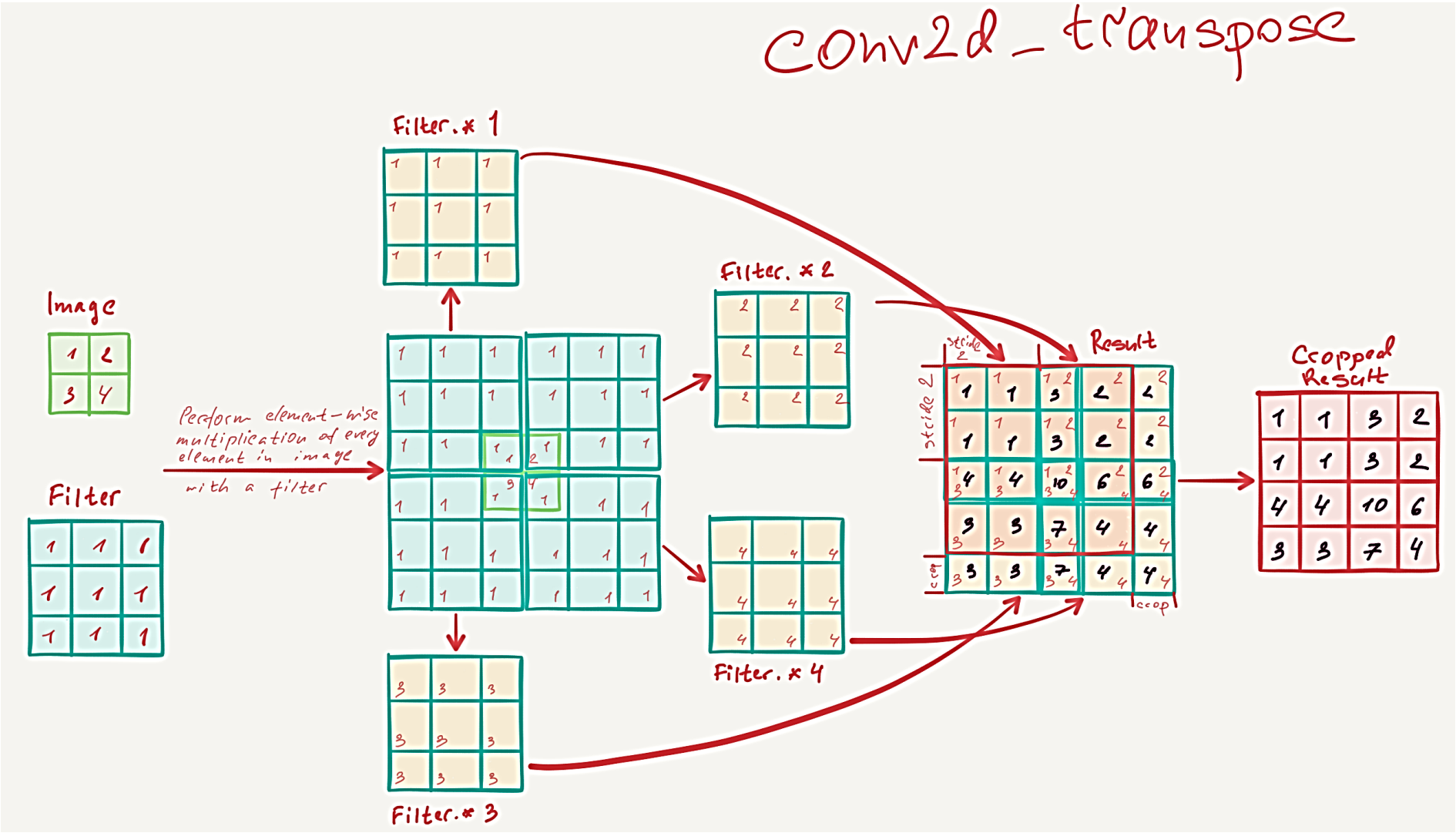


Image Source:

UNET [1]

Objective:

- **There is large consent that successful training of deep networks requires many thousand annotated training samples.**
- In UNET paper, authors presented a network and training strategy that relied on the strong **use** of **data augmentation to use the available annotated samples more efficiently.**
- The architecture consisted of a contracting path to capture context and a symmetric expanding path that enabled precise localization.
- **They showed that such a network could be trained end-to-end from very few images** and outperformed the prior best method on the ISBI challenge for segmentation of neuronal structures in electron microscopic stacks.

UNET [1]

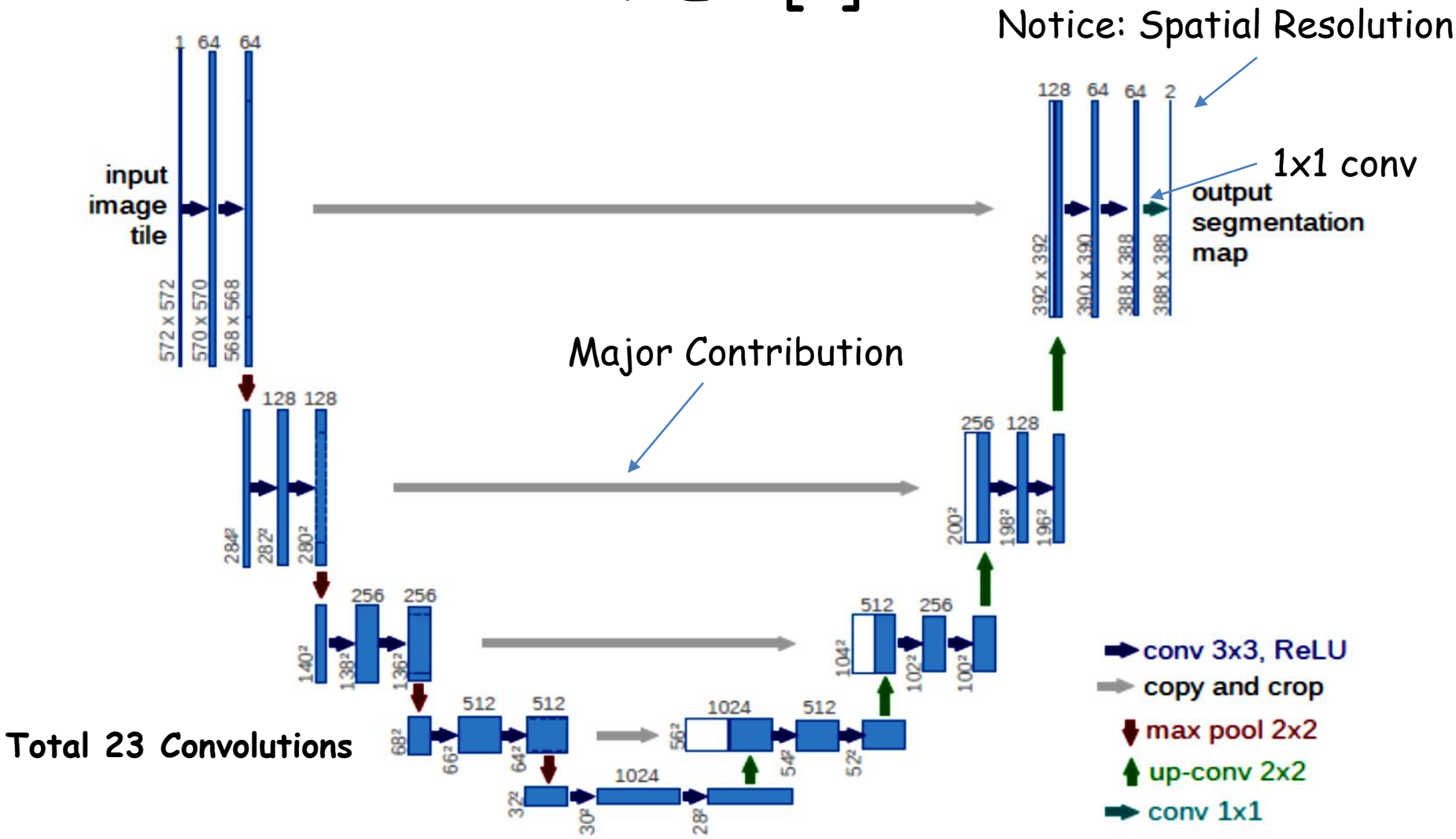


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows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

UNET [1]

To allow a seamless tiling of the output segmentation map (see Figure 2), it is important to select the input tile size such that all 2x2 max-pooling operations are applied to a layer with an even x- and y-size.

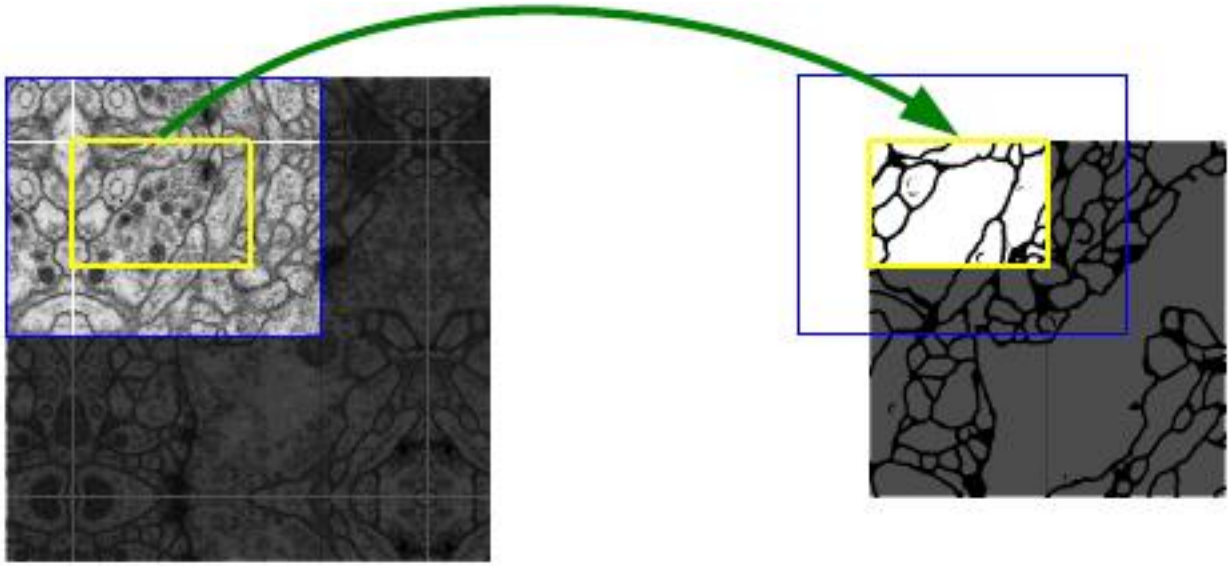


Fig. 2. Overlap-tile strategy for seamless segmentation of arbitrary large images (here segmentation of neuronal structures in EM stacks). Prediction of the segmentation in the yellow area, requires image data within the blue area as input. Missing input data is extrapolated by mirroring

UNET – Training [1]

- The input images and their corresponding segmentation maps were used to train the network with the **stochastic gradient descent with momentum.**

- Due to the unpadded convolutions, the output image was smaller than the input by a constant border width.

- Accordingly, they used a high **momentum (0.99)** such that a large number of the previously seen training samples determine the update in the current optimization step.

- The energy function was computed by a pixel-wise soft-max over the final feature map combined with the cross entropy loss function

UNET – Training [1]

- The soft-max is defined as:

The energy function is computed by a pixel-wise soft-max over the final feature map combined with the cross entropy loss function. The soft-max is defined as $p_k(\mathbf{x}) = \exp(a_k(\mathbf{x}))/\left(\sum_{k'=1}^{K}\exp(a_{k'}(\mathbf{x}))\right)$ where $a_k(\mathbf{x})$ denotes the activation in feature channel k at the pixel position $\mathbf{x} \in \Omega$ with $\Omega \subset \mathbb{Z}^2$. K is the number of classes and $p_k(\mathbf{x})$ is the approximated maximum-function. I.e. $p_k(\mathbf{x}) \approx 1$ for the k that has the maximum activation $a_k(\mathbf{x})$ and $p_k(\mathbf{x}) \approx 0$ for all other k. The cross entropy then penalizes at each position the deviation of $p_{\ell(\mathbf{x})}(\mathbf{x})$ from 1 using

$$E = \sum_{\mathbf{x}\in\Omega} w(\mathbf{x})\log(p_{\ell(\mathbf{x})}(\mathbf{x})) \tag{1}$$

where $\ell : \Omega \to \{1,\ldots,K\}$ is the true label of each pixel and $w : \Omega \to \mathbb{R}$ is a weight map that we introduced to give some pixels more importance in the training.

UNET – Training [1]

- They pre-computed the weight map for each ground truth segmentation to compensate the different frequency of pixels from a certain class in the training data set, and to force the network to learn the small separation borders that we introduce between touching cells (See Figure 3c and d).

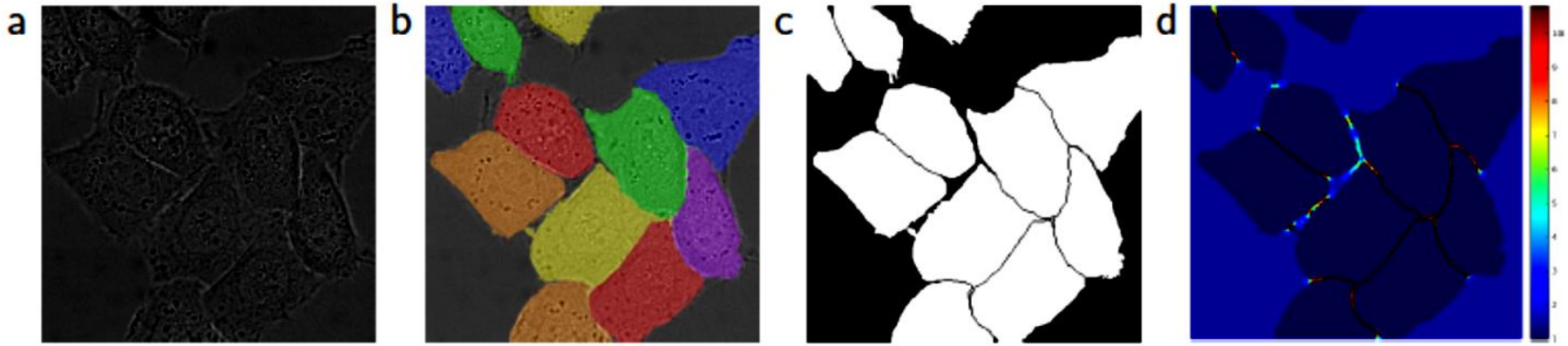


Fig. 3. HeLa cells on glass recorded with DIC (differential interference contrast) microscopy. (**a**) raw image. (**b**) overlay with ground truth segmentation. Different colors indicate different instances of the HeLa cells. (**c**) generated segmentation mask (white: foreground, black: background). (**d**) map with a pixel-wise loss weight to force the network to learn the border pixels.

UNET – Training [1]

The separation border is computed using morphological operations. The weight map is then computed as

$$w(\mathbf{x}) = w_c(\mathbf{x}) + w_0 \cdot \exp\left(-\frac{(d_1(\mathbf{x}) + d_2(\mathbf{x}))^2}{2\sigma^2}\right) \tag{2}$$

where $w_c : \Omega \to \mathbb{R}$ is the weight map to balance the class frequencies, $d_1 : \Omega \to \mathbb{R}$ denotes the distance to the border of the nearest cell and $d_2 : \Omega \to \mathbb{R}$ the distance to the border of the second nearest cell. In our experiments we set $w_0 = 10$ and $\sigma \approx 5$ pixels.

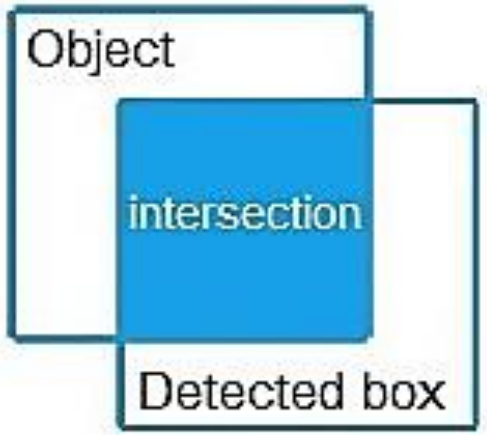
UNET – Training [1]

- Data augmentation is essential to teach the network the desired invariance and robustness properties, when only few training samples are available.

- In case of microscopical images we primarily need shift and rotation invariance as well as robustness to deformations and gray value variations.

Intersection over Union

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



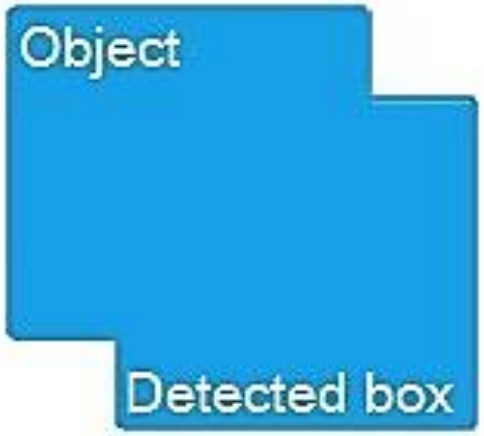


Image Source: https://i.stack.imgur.com/JlHnn.jpg

Intersection over Union

Ground truth

Prediction

$$IoU = \frac{\text{area of overlap}}{\text{area of union}}$$

Image Source: https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173

UNET – Results [1]

Table 2. Segmentation results (IOU) on the ISBI cell tracking challenge 2015.

Name	PhC-U373	DIC-HeLa
IMCB-SG (2014)	0.2669	0.2935
KTH-SE (2014)	0.7953	0.4607
HOUS-US (2014)	0.5323	-
second-best 2015	0.83	0.46
u-net (2015)	**0.9203**	**0.7756**

References

1. https://arxiv.org/pdf/1505.04597.pdf

Disclaimer

- These slides are not original and have been prepared from various sources for teaching purpose.