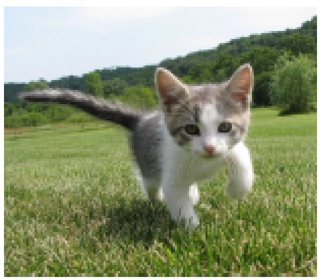
Semantic Segmentation

So far, Image Classification



This image is CC0 public domain

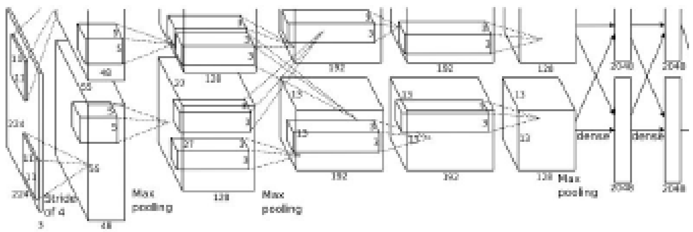


Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

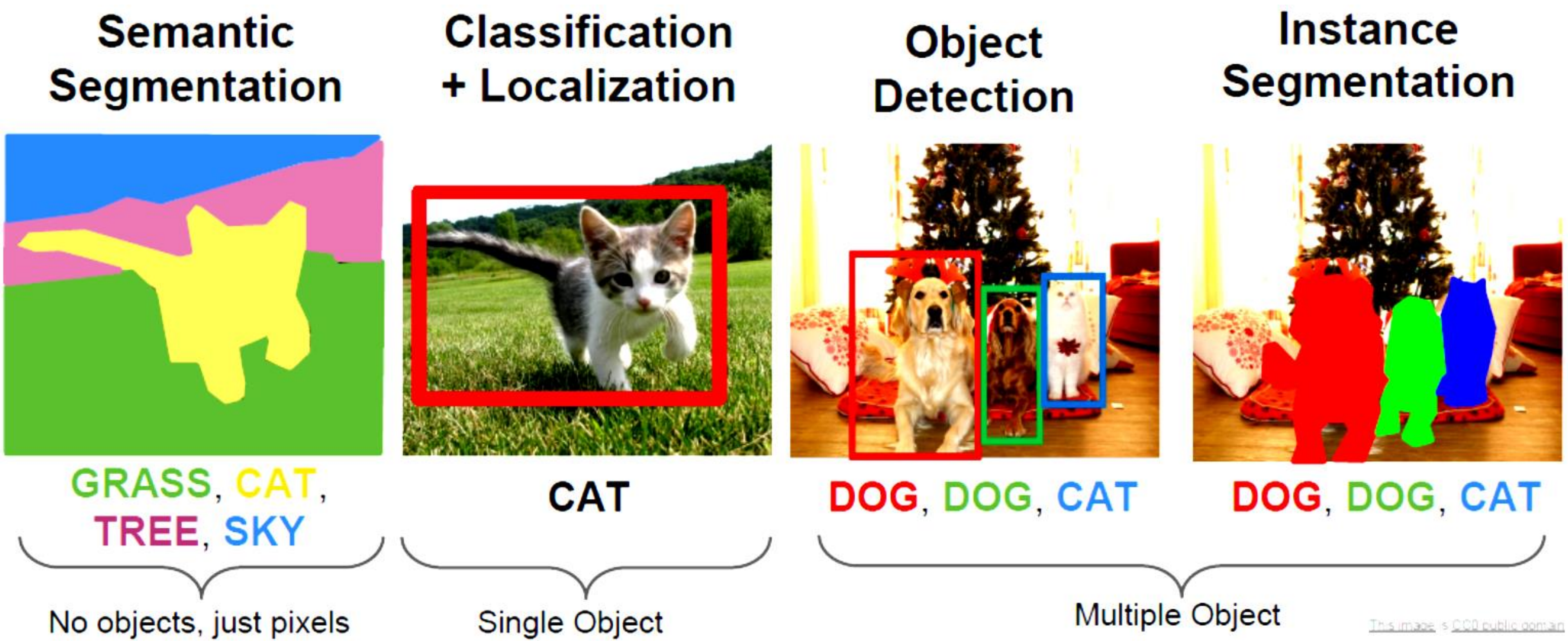
Vector: 4096

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



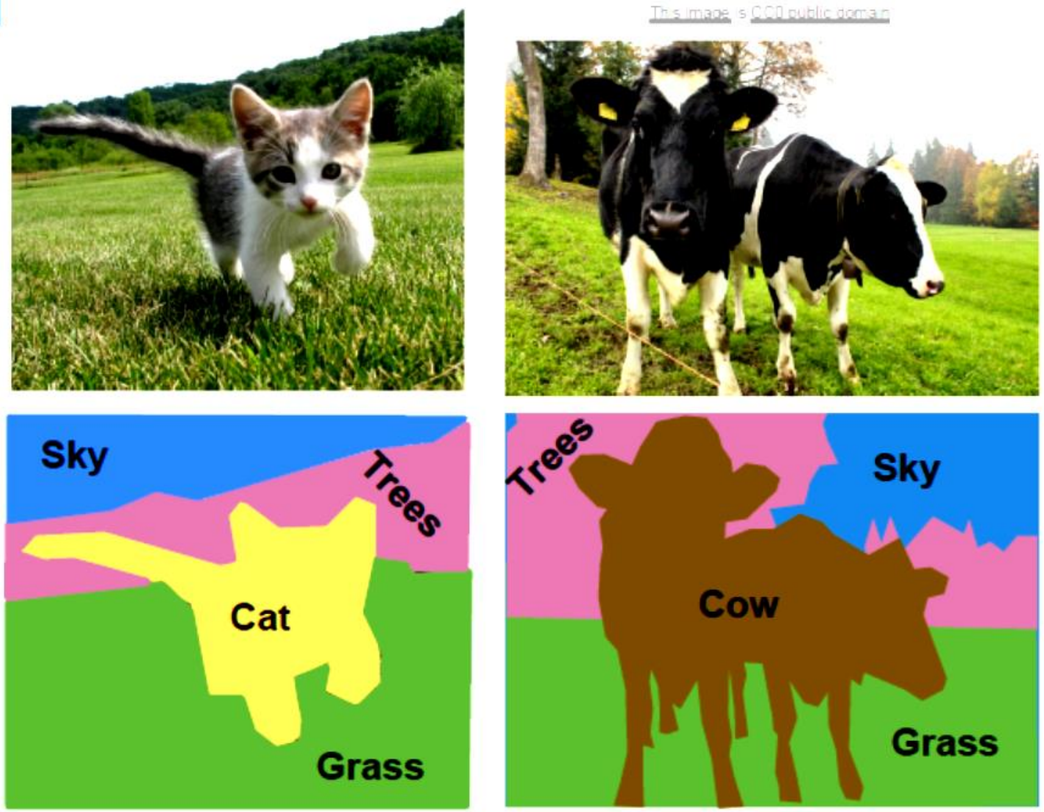
http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

Semantic Segmentation

Label each pixel in the image with a category label

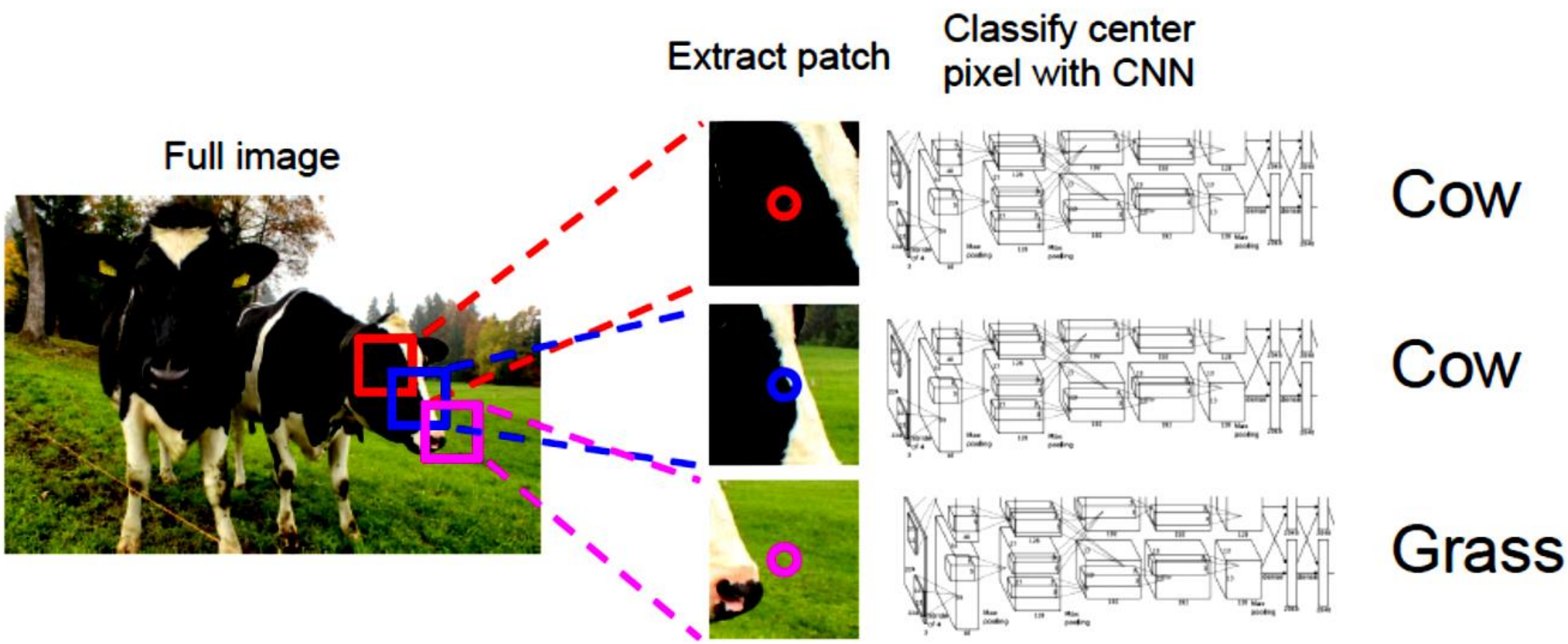
Don't differentiate instances, only care about pixels



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

Semantic Segmentation

Semantic Segmentation Idea: Sliding Window

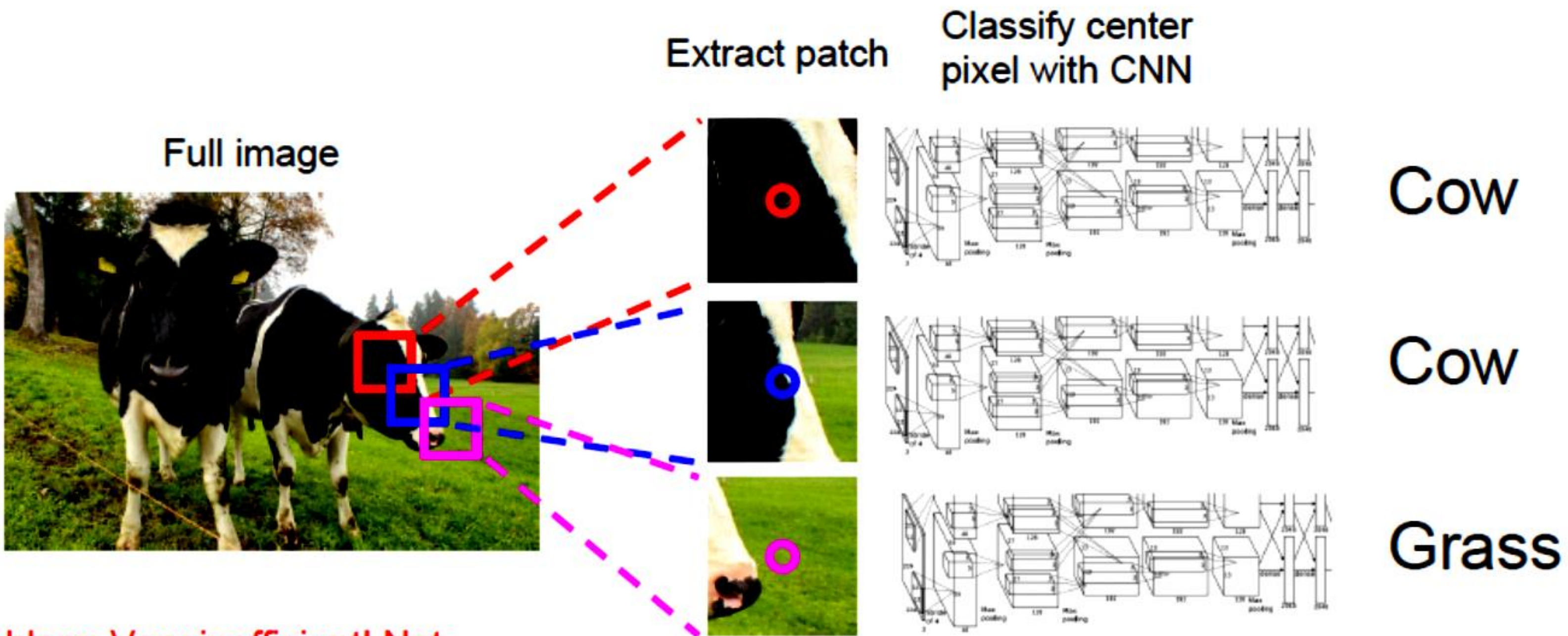


Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

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

Semantic Segmentation

Semantic Segmentation Idea: Sliding Window



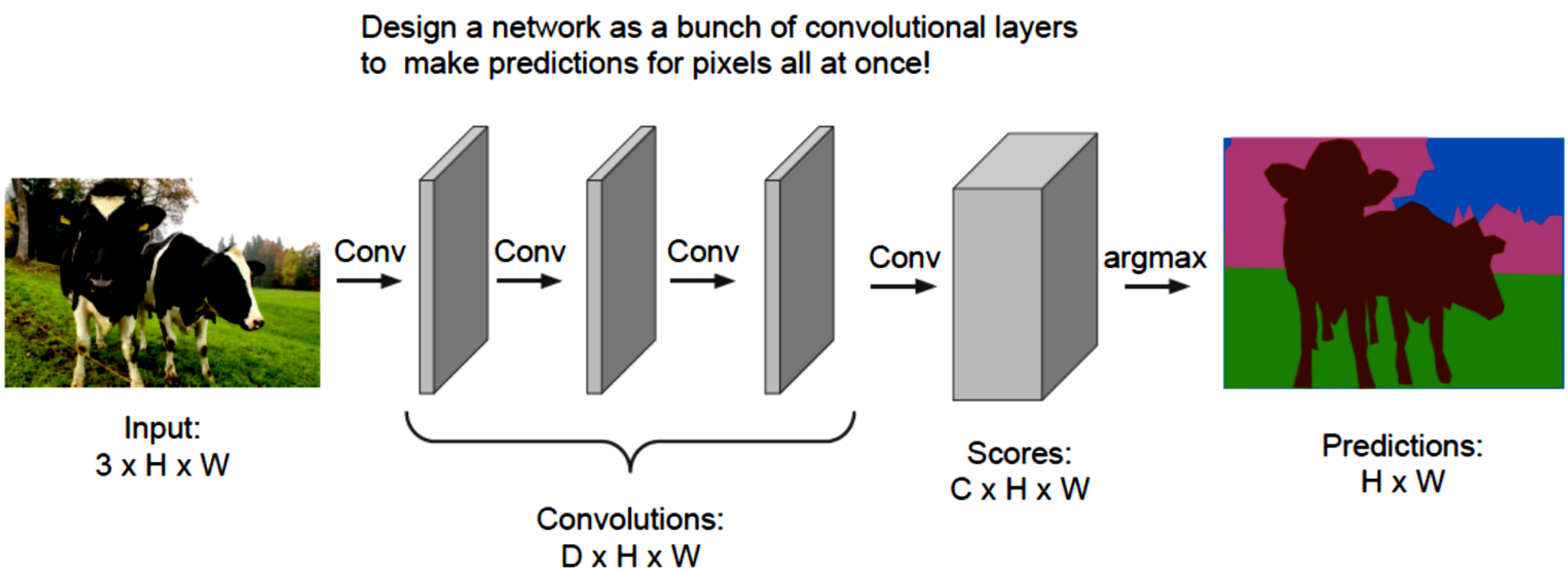
Problem: Very inefficient! Not reusing shared features between overlapping patches

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

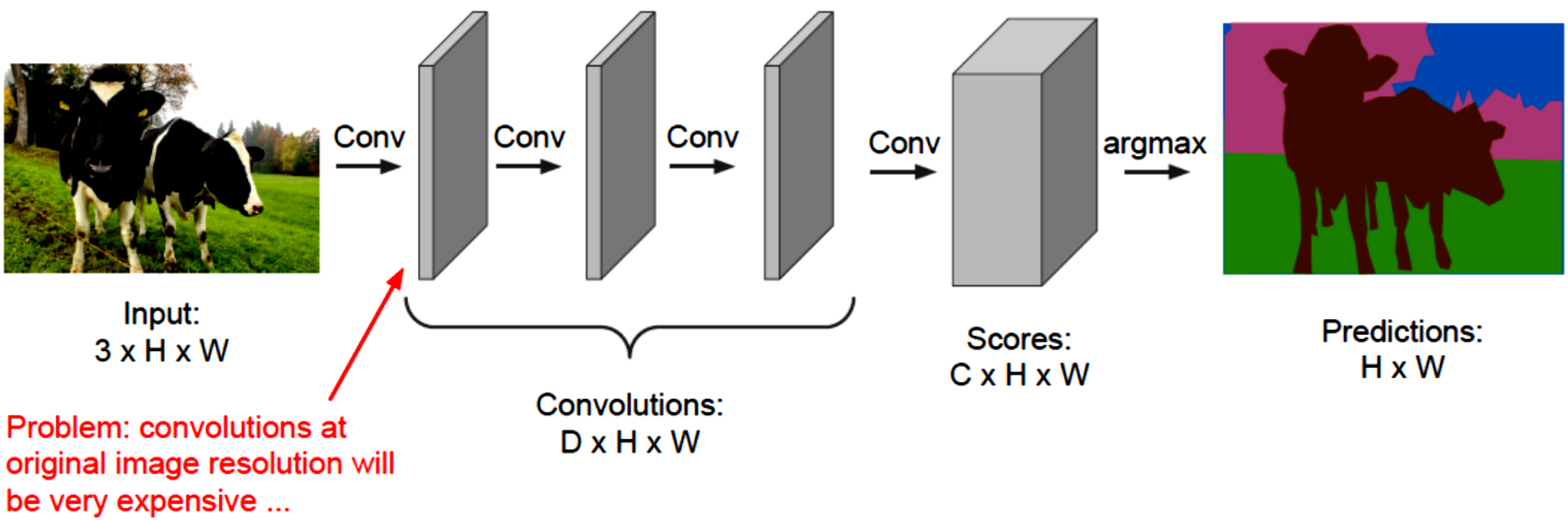


http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

Semantic Segmentation Idea: Fully Convolutional

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



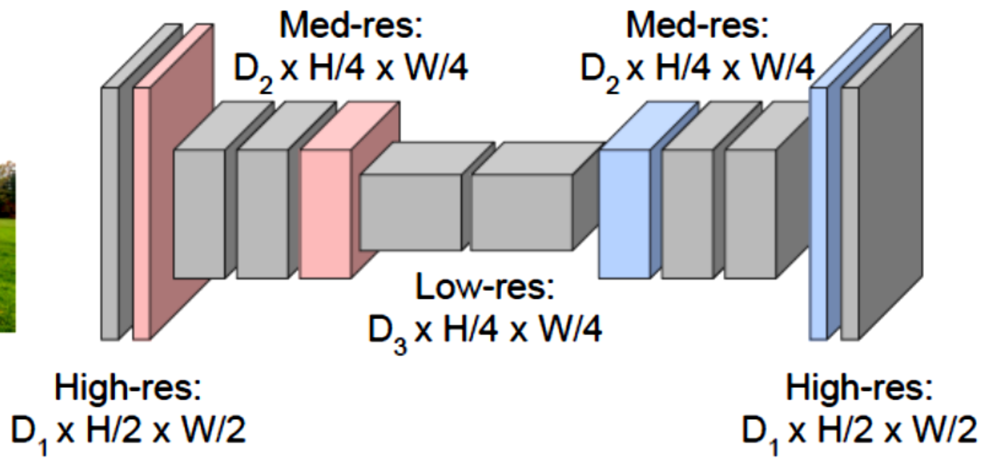
http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

Semantic Segmentation Idea: Fully Convolutional

Design netv a bunch of convolutional layers, with
downsamp d **upsampling** inside the network!

Input:
3 x H x W



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

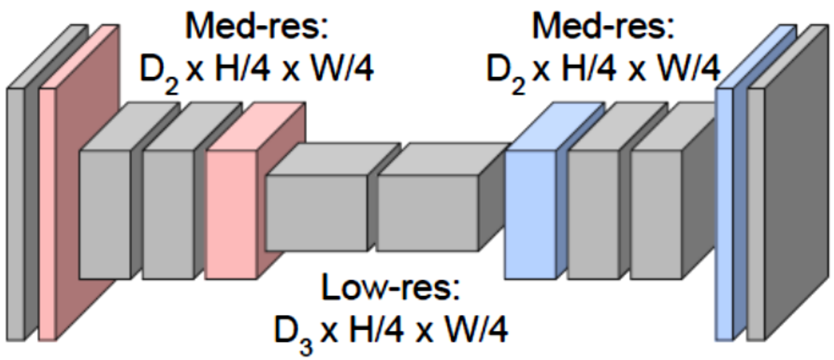
Semantic Segmentation Idea: Fully Convolutional

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

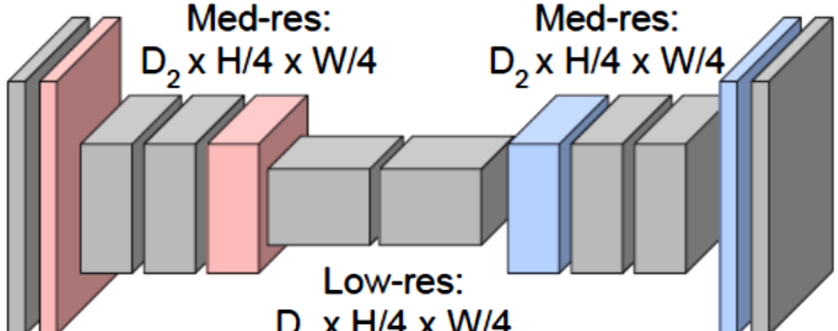
Semantic Segmentation Idea: Fully Convolutional

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



Med-res:
D_2 x H/4 x W/4

Med-res:
D_2 x H/4 x W/4

Low-res:
D_3 x H/4 x W/4

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”

Nearest Neighbor

Input: 2 x 2

1	2
3	4

Output: 4 x 4

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

“Bed of Nails”

Input: 2 x 2

1	2
3	4

Output: 4 x 4

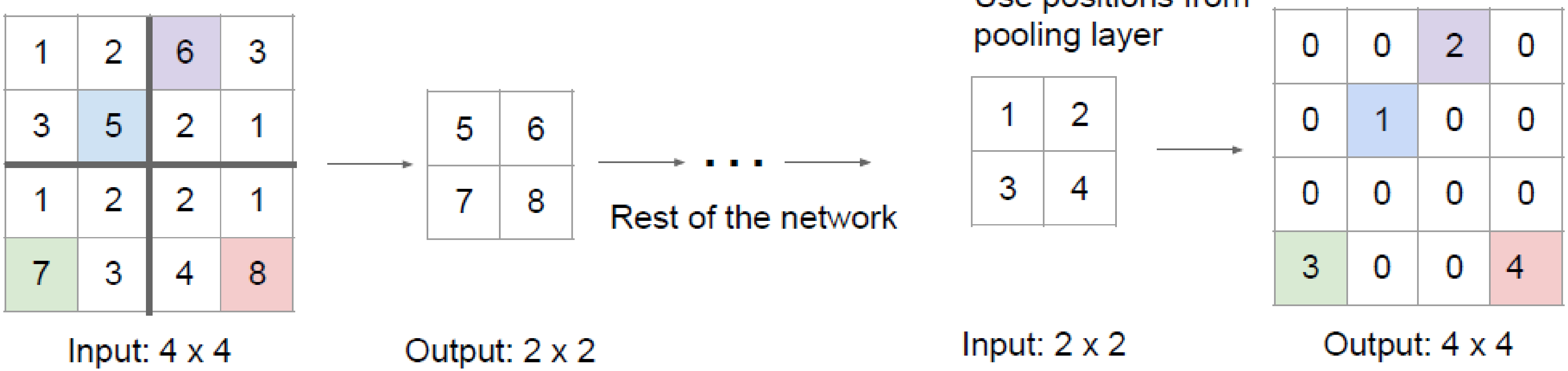
1	0	2	0
0	0	0	0
3	0	4	0
0	0	0	0

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

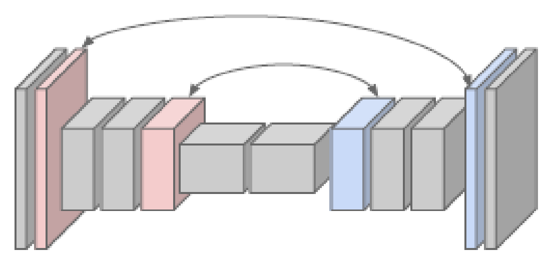
Semantic Segmentation

Max Pooling
Remember which element was max!

Max Unpooling
Use positions from pooling layer



Corresponding pairs of downsampling and upsampling layers

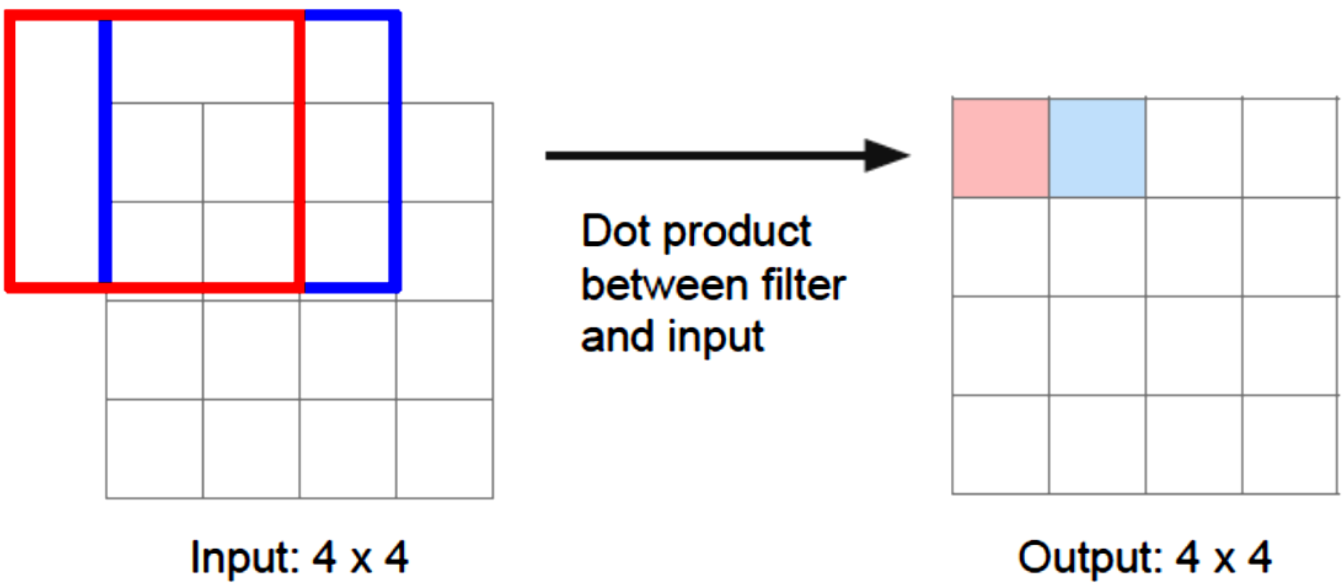


http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

Learnable Upsampling: Transpose Convolution

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

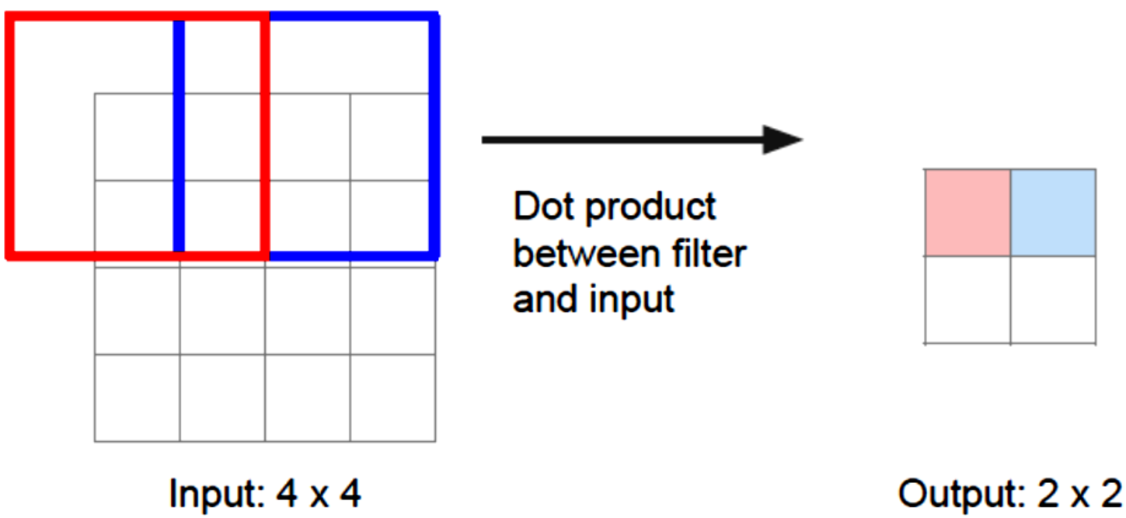


http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

Learnable Upsampling: Transpose Convolution

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



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

Stride gives ratio between movement in input and output

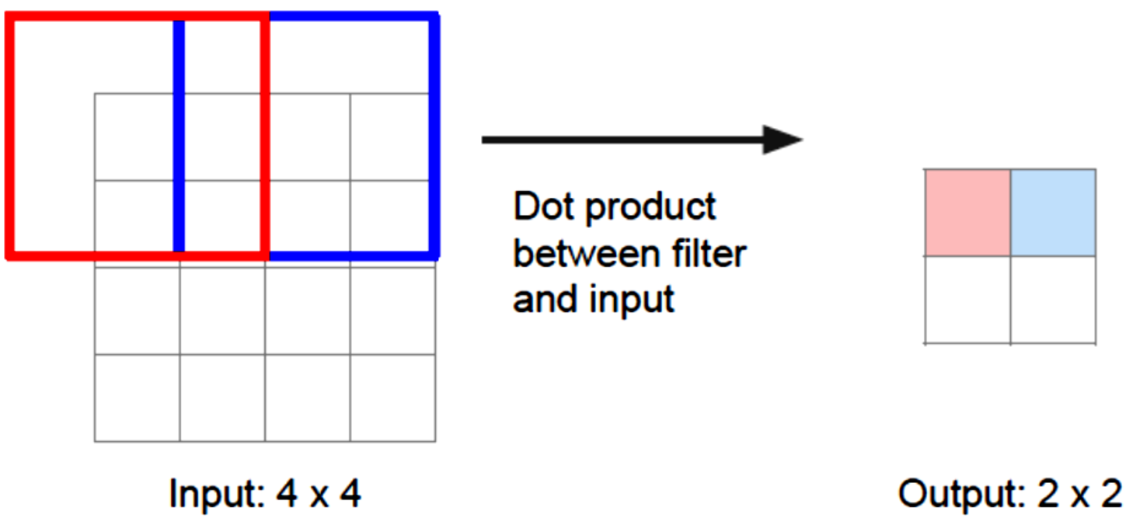
http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

Learnable Upsampling: Transpose Convolution

Recall: Normal 3 x 3 convolution, <u>stride 2</u> pad 1

2P = 1



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

Stride gives ratio between movement in input and output

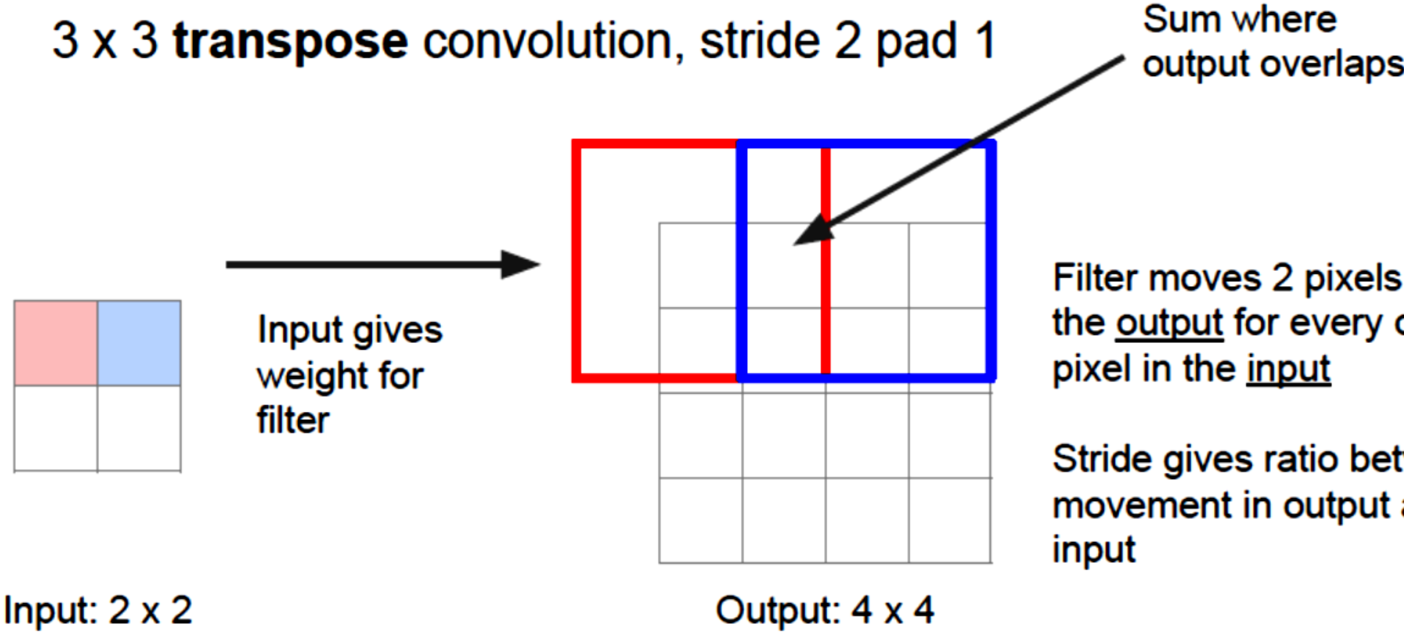
http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture11.pdf

Semantic Segmentation

Learnable Upsampling: Transpose Convolution

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

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



Filter moves 2 pixels in the <u>output</u> for every one pixel in the <u>input</u>

Stride gives ratio between movement in output and input

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

Semantic Segmentation

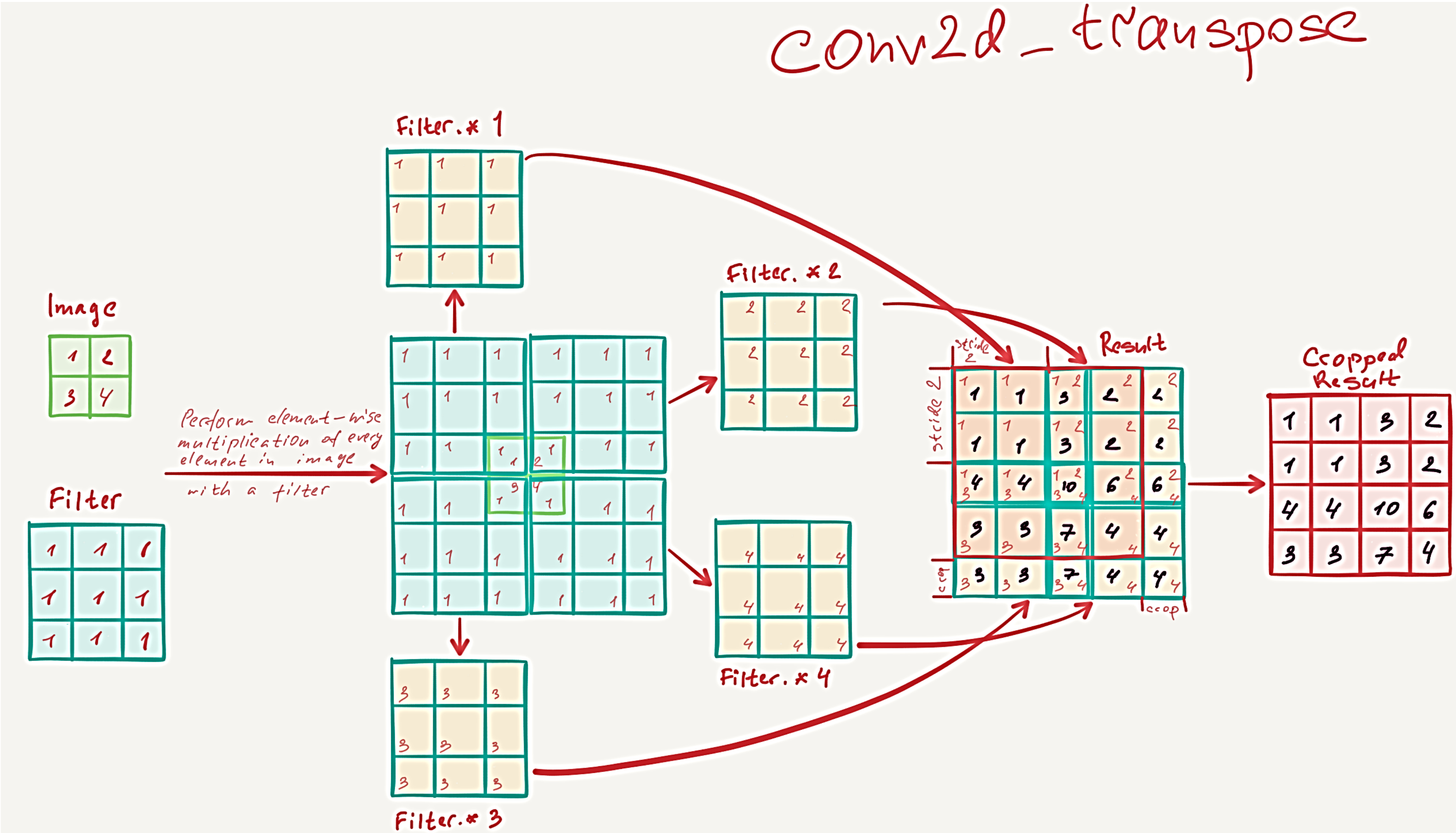


Image Source:

Semantic Segmentation

UNET [1]

Objective:

- **There is large consent that successful training of deep networks requires many thousand annotated training samples.**

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**.

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.

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]

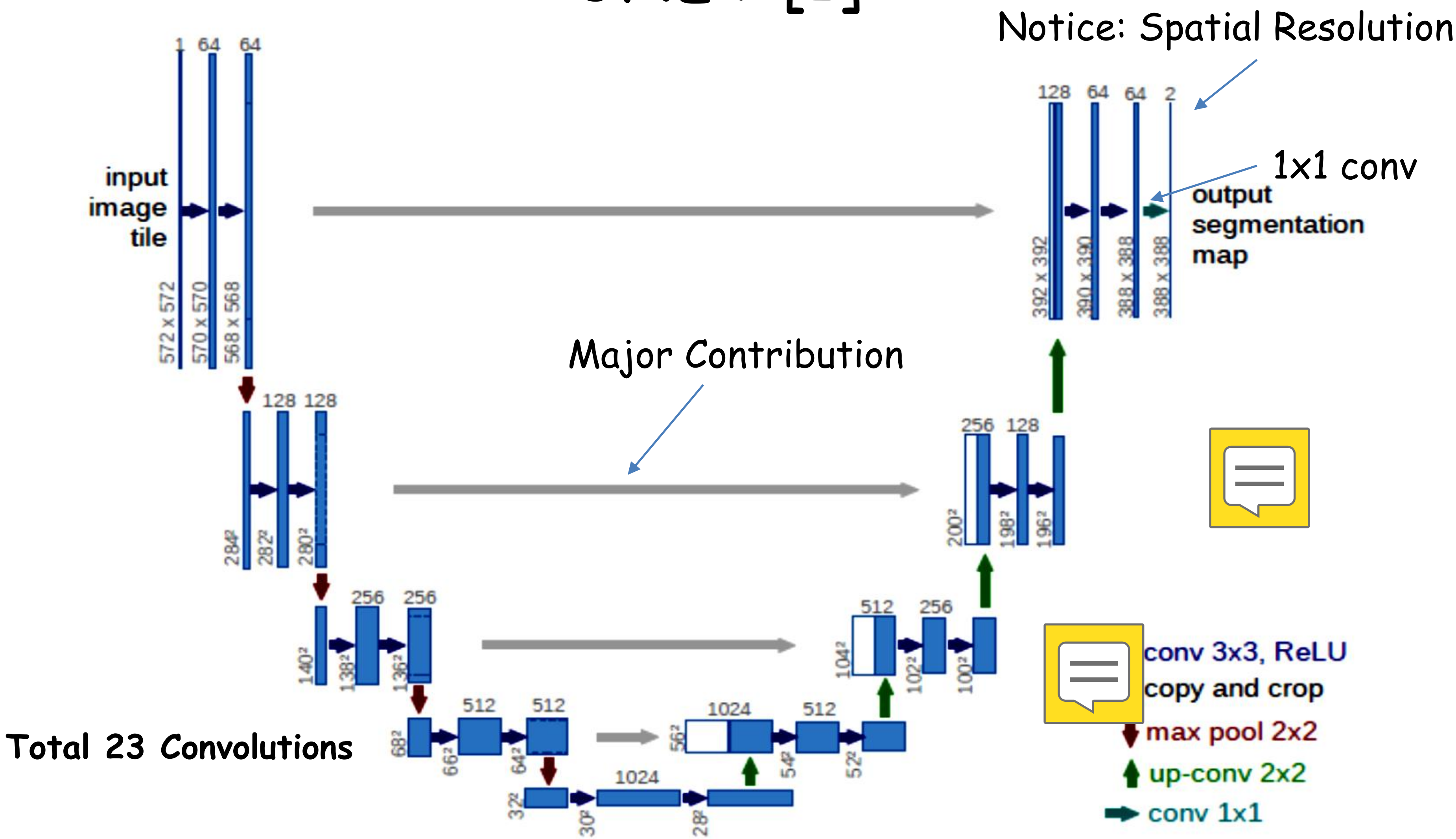


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows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

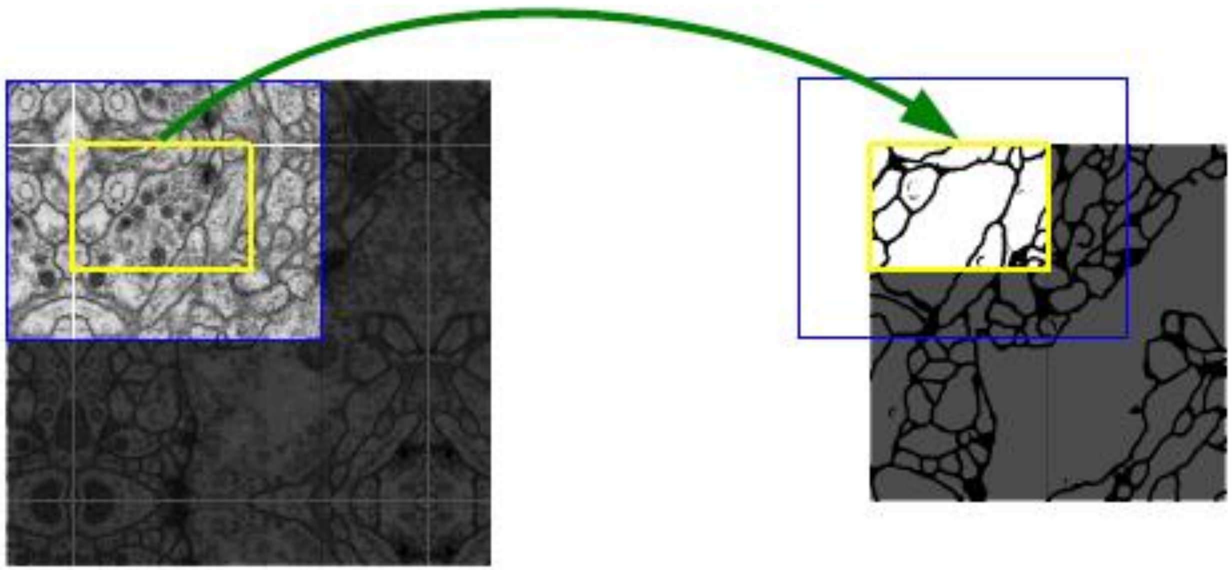


Fig. 2. Overlap-tile strategy for seamless segmentation of arbitrary large images (here segmentation of neuronal structures in EM stacks). Prediction of the segmentation in the yellow area, requires image data within the blue area as input. Missing input data is extrapolated by mirroring

UNET [1]

Major Contributions:
In order to localize, high resolution features from the contracting path are combined with the upsampled output. A successive convolution layer can then learn to assemble a more precise output based on this information.

UNET [1]

Major Contributions:

As for their tasks there is very little training data available, they used excessive data augmentation by applying elastic deformations to the available training images.

UNET [1]

Major Contributions:

As for their tasks there is very little training data available, they used excessive data augmentation by applying elastic deformations to the available training images.

This allowed the network to learn invariance to such deformations, without the need to see these transformations in the annotated image corpus.

UNET [1]

Major Contributions:

As for their tasks there is very little training data available, they used excessive data augmentation by applying elastic deformations to the available training images.

This allowed the network to learn invariance to such deformations, without the need to see these transformations in the annotated image corpus.

This is particularly important in biomedical segmentation, since deformation used to be the most common variation in tissue and realistic deformations can be simulated efficiently.

UNET – Training [1]

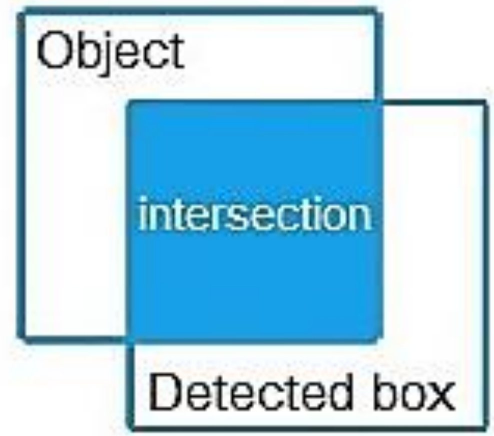
- The input images and their corresponding segmentation maps were used to train the network with the **stochastic gradient descent with momentum.**

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

- 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.

Intersection over Union

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



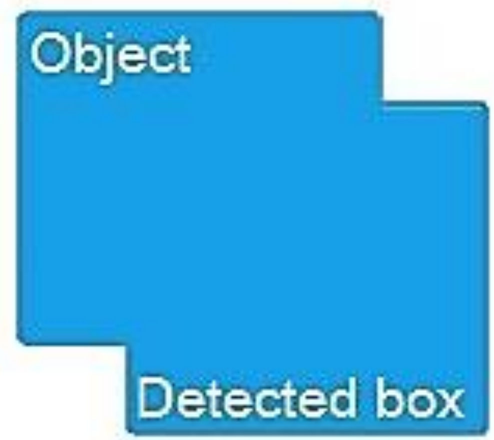


Image Source: https://i.stack.imgur.com/JlHnn.jpg

Intersection over Union

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

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

Skip Connection:

1

What are skip connections?:Skip connections are a technique that allows convolutional neural networks (CNNs) to bypass some layers and connect directly to deeper or shallower ones. They can improve the performance and efficiency of CNNs, but they also have some drawbacks and limitations. In this article, we will explore what skip connections are, how they work, and what are their benefits and drawbacks for CNNs.

Skip connections are a type of shortcut that connects the output of one layer to the input of another layer that is not adjacent to it. For example, in a CNN with four layers, A, B, C, and D, a skip connection could connect layer A to layer C, or layer B to layer D, or both. Skip connections can be implemented in different ways, such as adding, concatenating, or multiplying the outputs of the skipped layers.

2

How do skip connections work?:Skip connections work by allowing information and gradients to flow more easily through the network. Information is the input data and the features extracted by the layers, while gradients are the signals that adjust the weights of the layers during backpropagation. Skip connections can help to preserve information and gradients that might otherwise be lost or diluted by passing through multiple layers. They can also help to combine features from different levels of abstraction and resolution, which can enhance the representation power of the network.

3

What are the benefits of skip connections?

Skip connections can provide several benefits for CNNs, such as improving accuracy and generalization, solving the vanishing gradient problem, and enabling deeper networks. Skip connections can help the network to learn more complex and diverse patterns from the data and reduce the number of parameters and operations needed by the network. Additionally, skip connections can help to alleviate the problem of vanishing gradients by providing alternative paths for the gradients to flow. Furthermore, they can make it easier and faster to train deeper networks, which have more expressive power and can capture more features from the data.

4

What are the drawbacks of skip connections?:Skip connections are a popular and powerful technique for improving the performance and efficiency of CNNs, but they are not a panacea. They can help preserve information and gradients, combine features, solve the vanishing gradient problem, and enable deeper networks. However, they can also increase complexity and memory requirements, introduce redundancy and noise, and require careful design and tuning to match the network architecture and data domain. Different types and locations of skip connections can have different impacts on the network performance, with some being more beneficial or harmful than others. Thus, it is essential to understand how skip connections work and how to use them wisely and effectively for CNNs.