ULB

UNIVERSITÉ LIBRE DE BRUXELLES

Segmentation with DCNN and data augmentation

Common computer vision tasks

One object per class

Rabbit

Classification

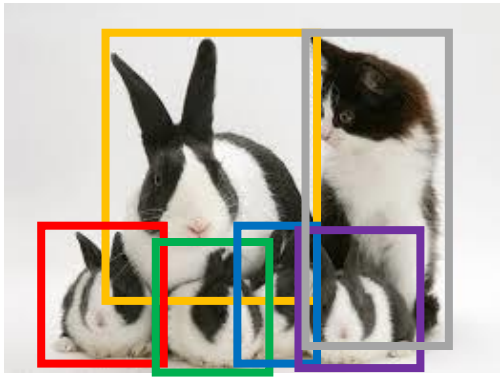
Rabbit

Localization

Sky, Rabbit, Grass

Semantic segmentation

Multiple objects per class



Rabbit, Rabbit, Rabbit, Rabbit, Rabbit, Cat

Detection

Rabbit, Rabbit, Rabbit, Rabbit, Rabbit, Cat

Instance segmentation

Today

One object per class

Rabbit

Classification

Rabbit

Localization

Sky, Rabbit, Grass

Semantic segmentation

Multiple objects per class

Rabbit, Rabbit, Rabbit, Rabbit, Rabbit, Cat

Detection

Rabbit, Rabbit, Rabbit, Rabbit, Rabbit, Cat

Instance segmentation

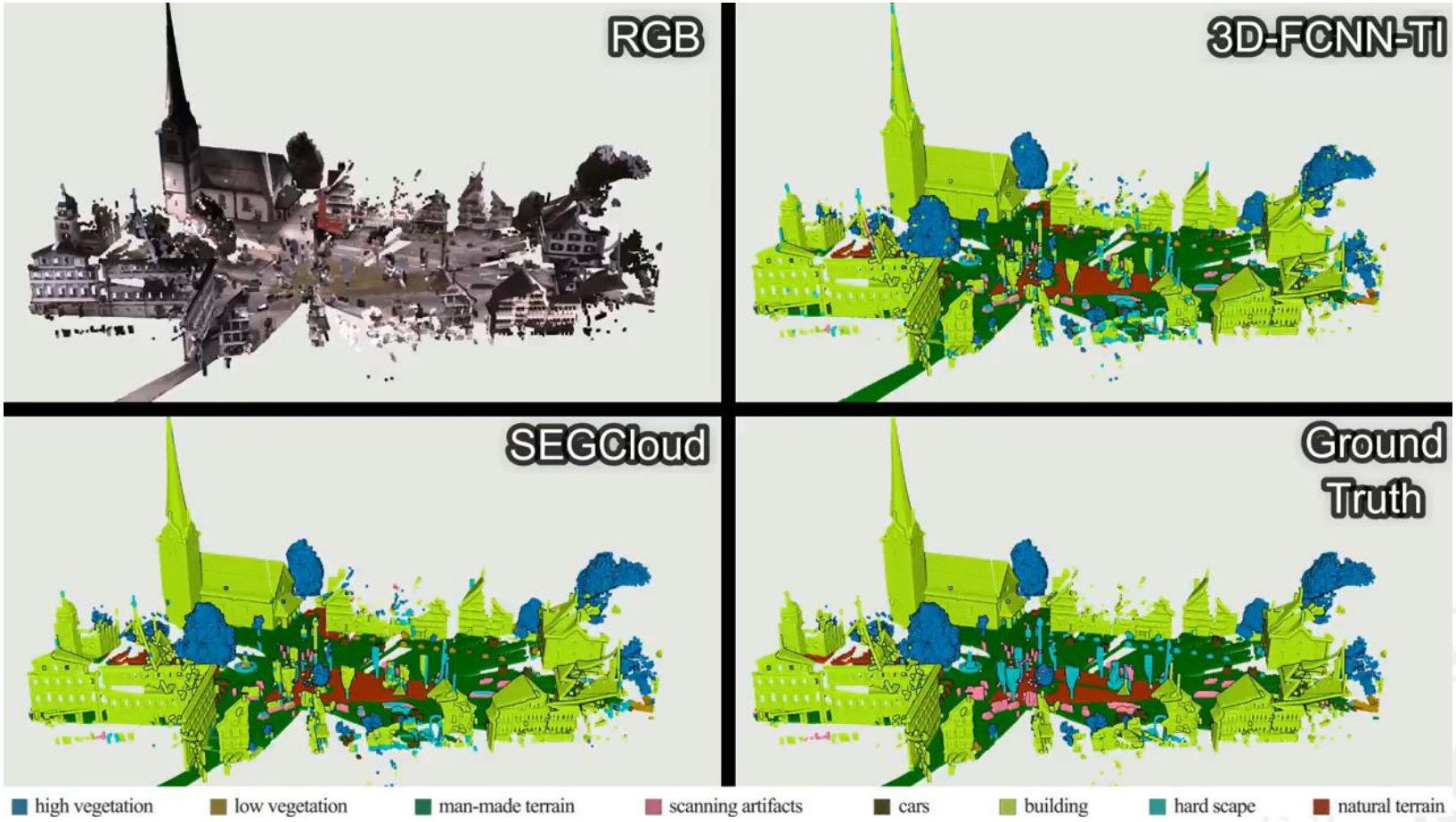
Semantic segmentation

Each pixel in the image is labelled with a class

Do not separate objects of the same class

Example: autonomous driving

Example: 3D point cloud

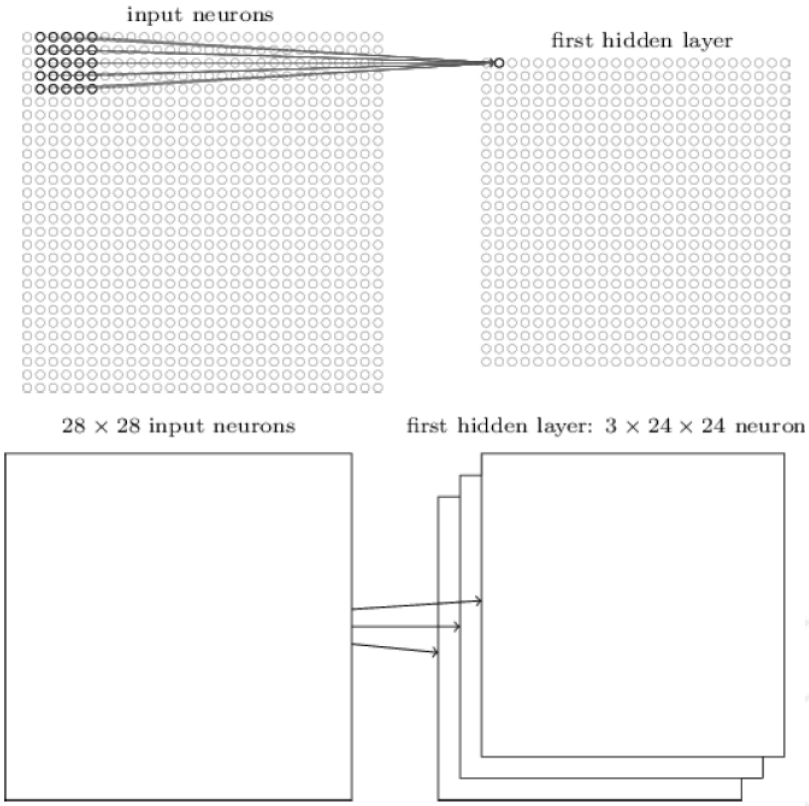


Reminder from last lecture: Convolution

In a **convolutional** layer, connections are only made in a **local receptive field**.

$$o_{l,m} = \phi\left(\sum_{ij}(w_{ij}x_{l+i,m+j} + b_{l,m})\right)$$

All neurons from the same **feature map** share the **same weights**. The feature maps are the result of the convolution of the input image by the weights of the connections.

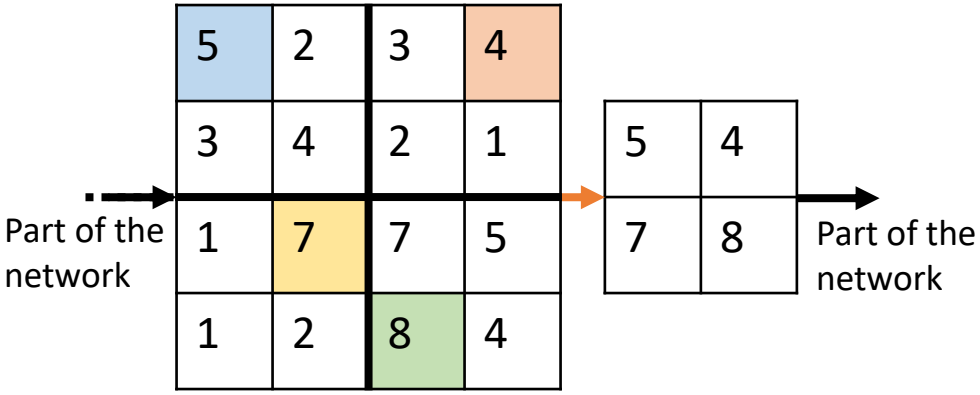


Michael Nielsen, http://neuralnetworksanddeeplearning.com/chap6.html

RELU is usually used after convolutions

Reminder from last lecture: pooling layers

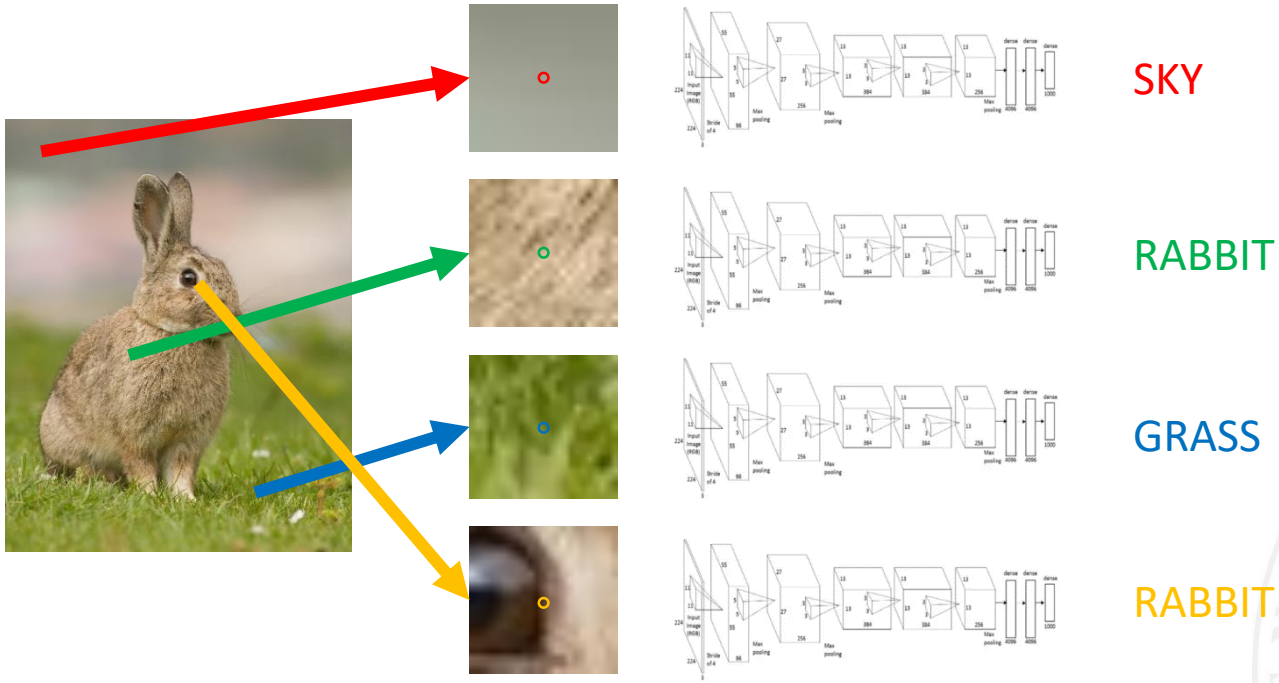
Convolutional layers are often used in conjunction with **pooling layers**.



Example: max-pooling 2x2

Semantic segmentation: How?

1. Sliding Window
    1. Extract Patch
    2. Classify pixel at the center of the patch
    3. Repeat for all the pixels

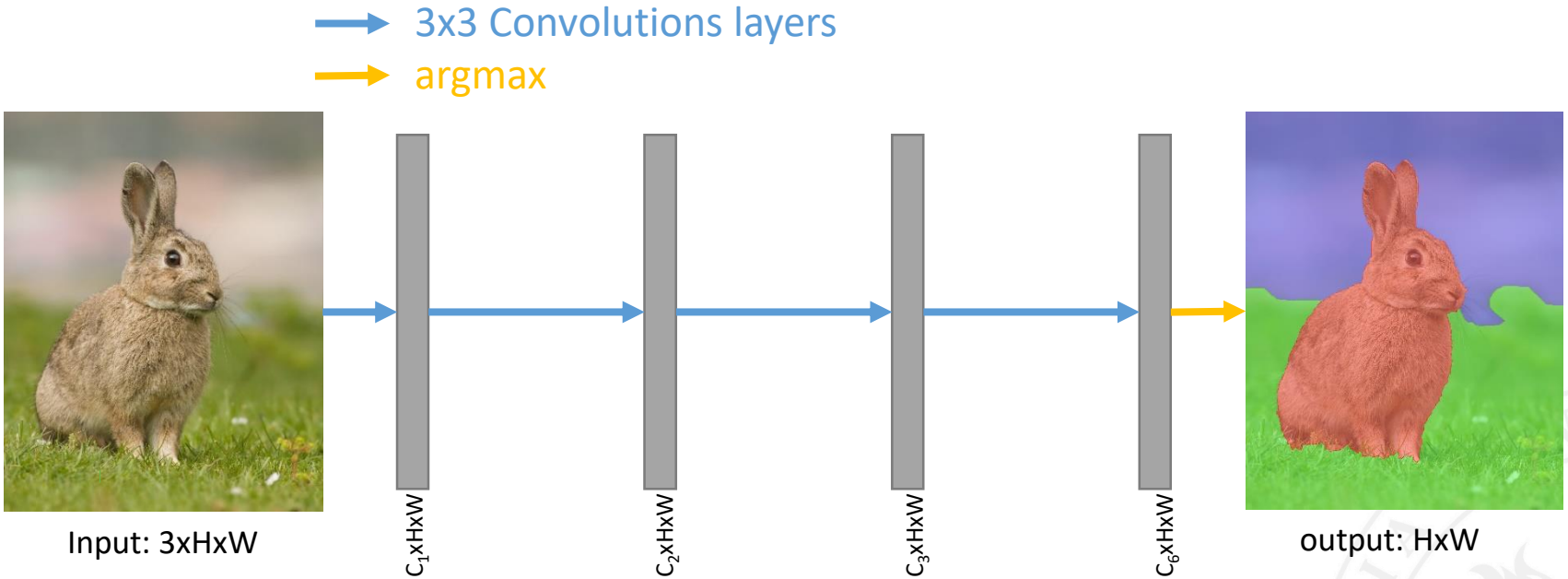


Pros and Cons

- Pros
  - Simple
- Cons
  - Inefficient/Slow
  - Shared features are recomputed for each pixel classification

Semantic segmentation: How?

2. Fully convolutional network
    1. Design a network that takes one image has input and do the segmentation for you!

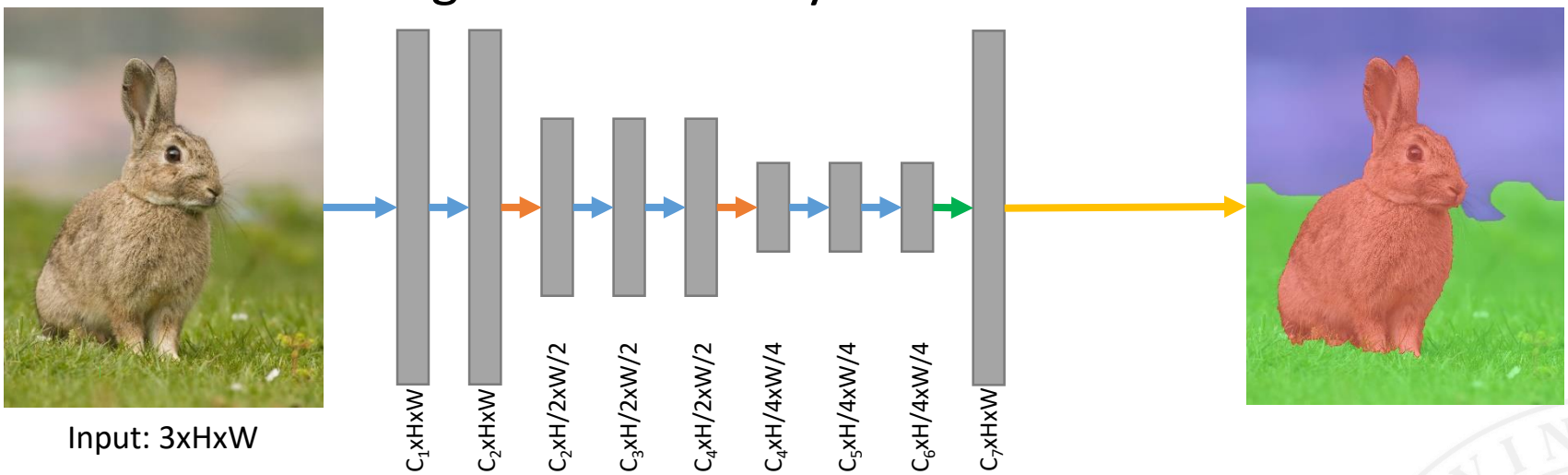


Problem: Full resolution convolution over the whole network are very expensive

➔Solution: Downsampling then upsampling

Semantic segmentation: How?

2. Fully convolutional network
    1. Design a network that takes one image has input and do the segmentation for you!



Downsampling and Upsamplig

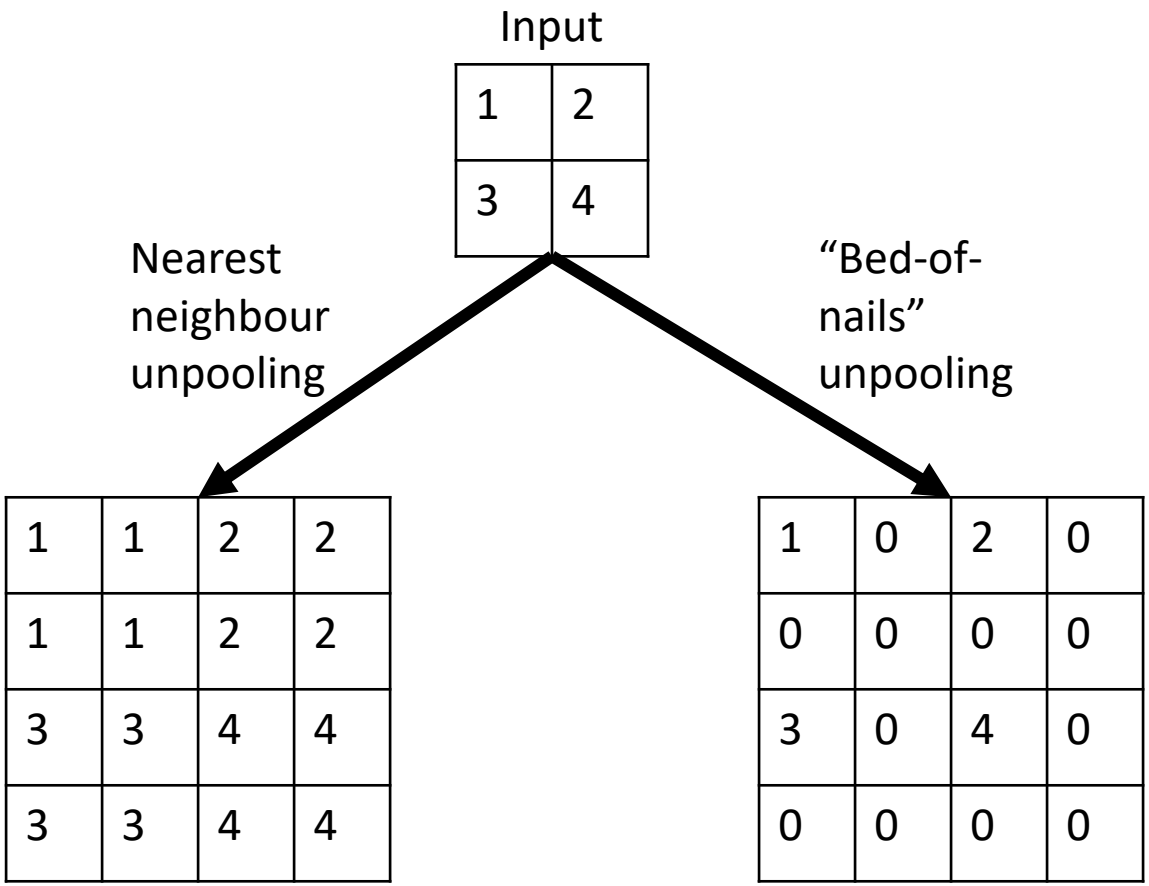
Downsampling

- Pooling (Max, Average,…)
- Strided convolutions

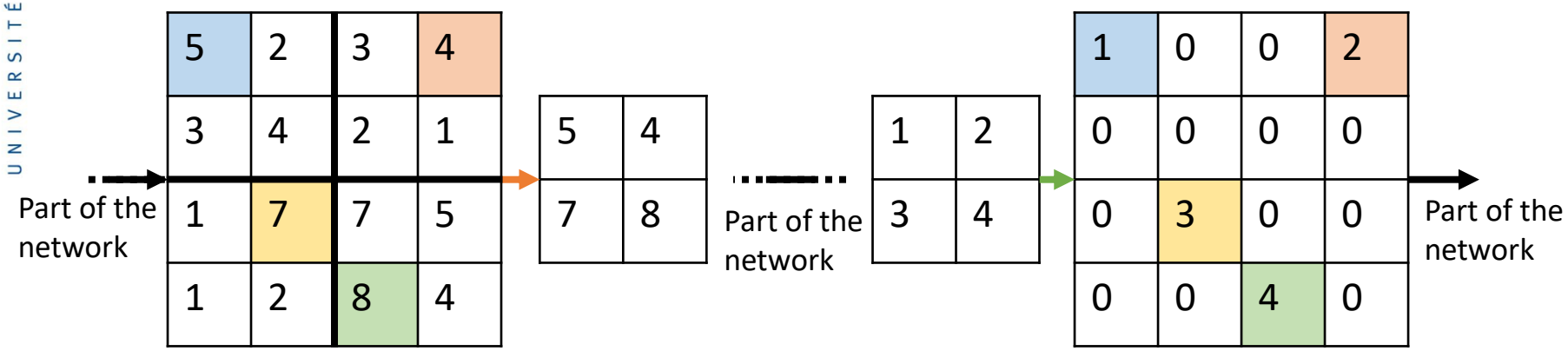
Upsampling

- “Unpooling”
- Transposed convolutions

Unpooling

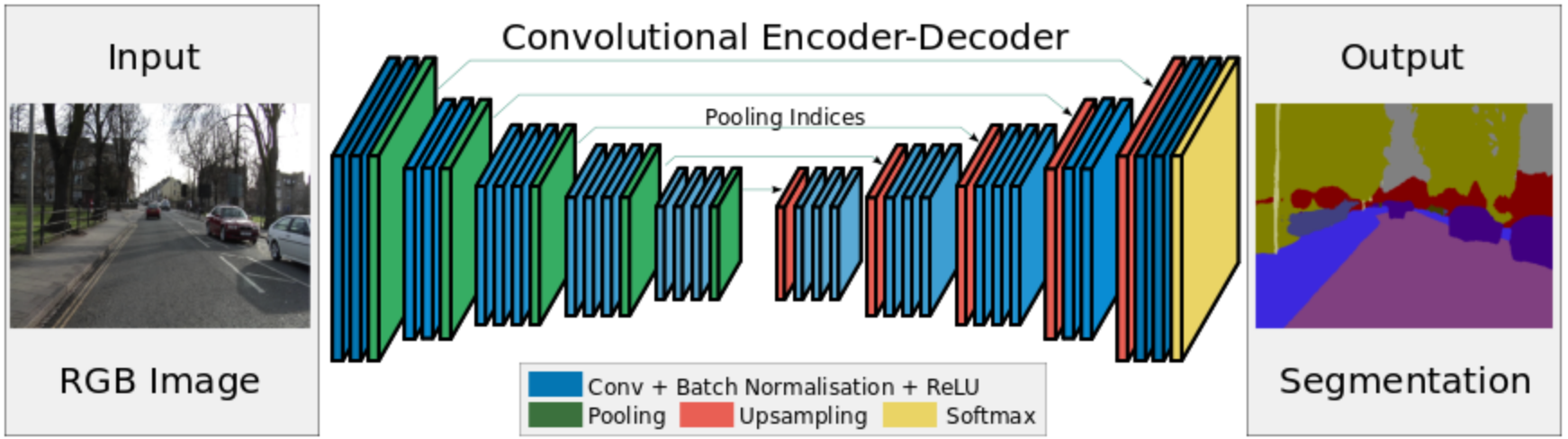

Input
1 2
3 4
Nearest neighbour unpooling
"Bed-of-nails" unpooling
1 1 2 2
1 1 2 2
3 3 4 4
3 3 4 4
1 0 2 0
0 0 0 0
3 0 4 0
0 0 0 0

Max-Unpooling



- There should be corresponding pairs of layers (1 max-pooling for each max-unpooling)
- ➔ You need to save the indices used in the pooling operation somewhere!

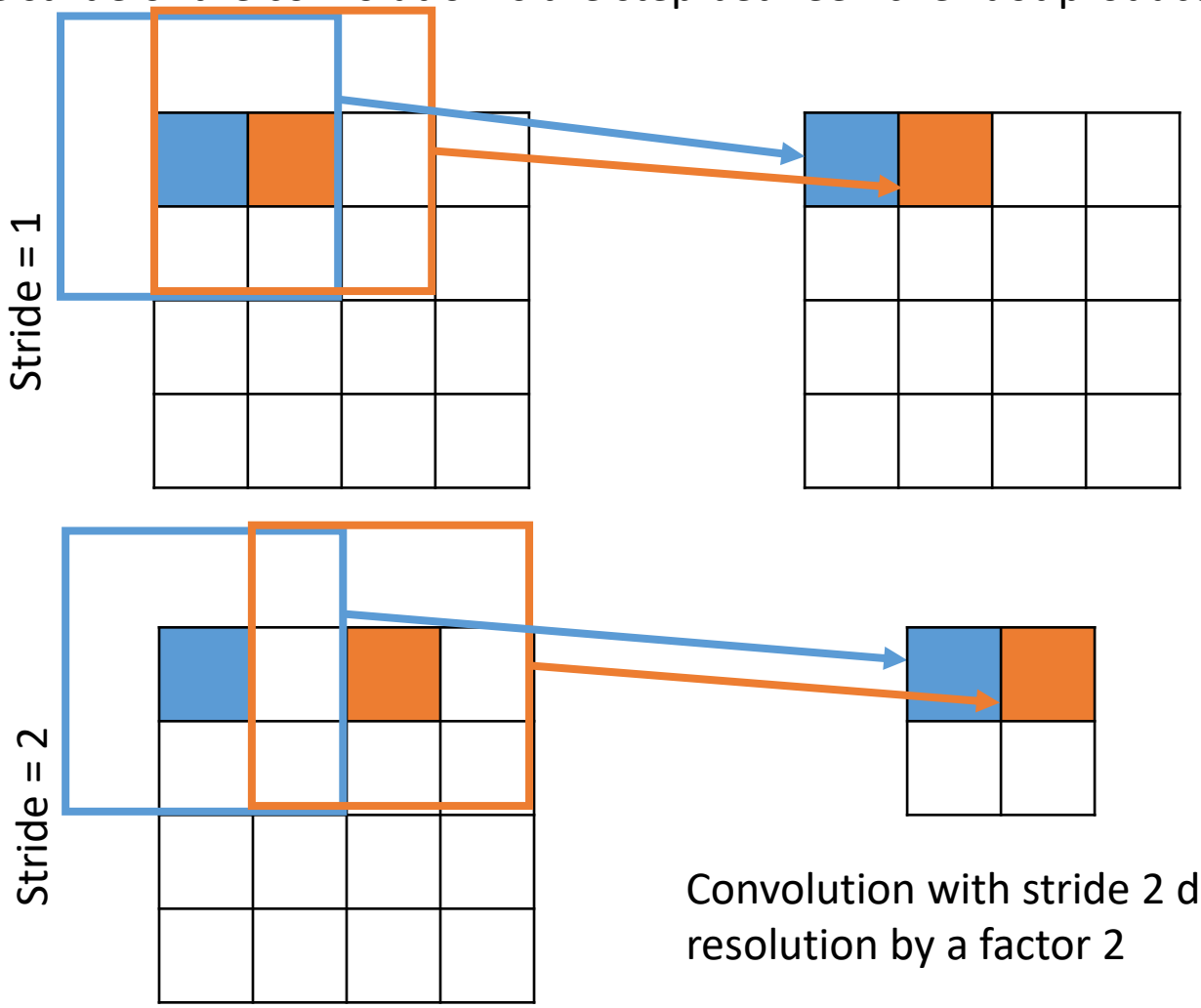
Example: Segnet



Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, *39*(12), 2481-2495.

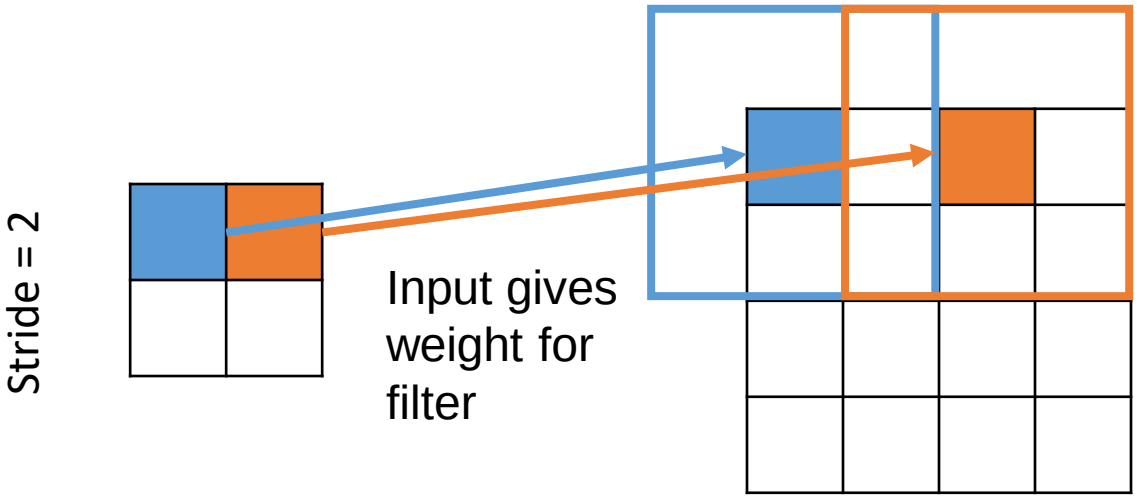
Transposed convolution: the stride parameter

The stride of the convolution is the step between chez dot product in the input:



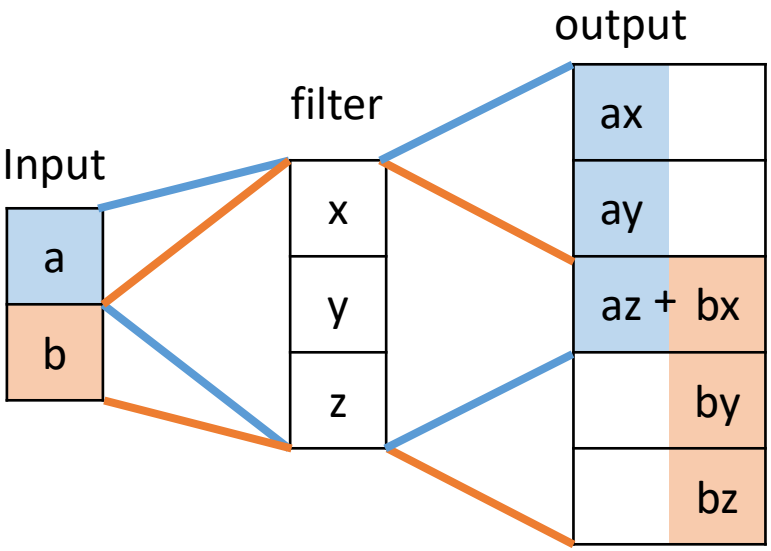
Convolution with stride 2 downsample the resolution by a factor 2

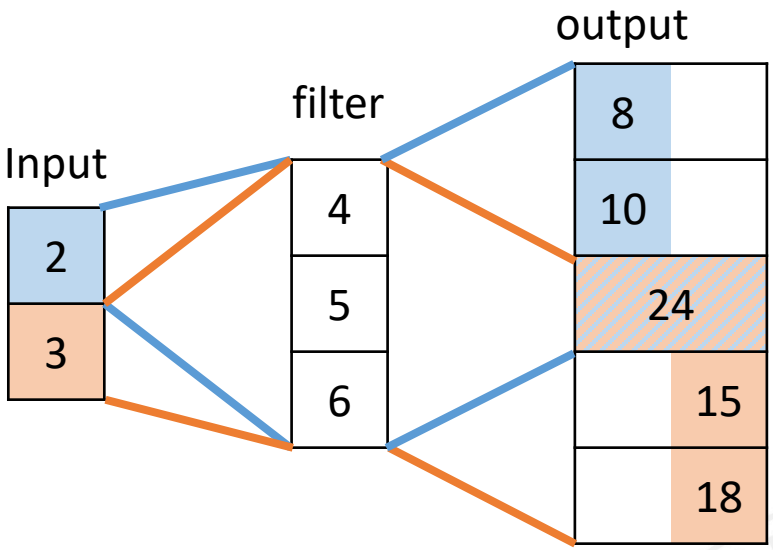
Transposed convolution



- Transposed convolution = inversed strided convolution
- Outputs that overlaps are summed
- Stride here is the ratio between step in input and output. Stride = 2 means 1 step in input is two step in output
- Transposed convolution is also called deconvolution (do not use it), upconvolution or backward strided convolution.

Transposed convolution: 1d example





Transposed convolution: 1d example using matrix multiplications, stride=1

Normal convolution

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

Transposed convolution

$$\begin{bmatrix} x & 0 & 0 & 0 \\ y & x & 0 & 0 \\ z & y & x & 0 \\ 0 & z & y & x \\ 0 & 0 & z & y \\ 0 & 0 & 0 & z \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} ax \\ ay + bx \\ az + by + cx \\ bz + cy + dx \\ cz + dy \\ dz \end{bmatrix}$$

When stride=1, transposed convolutions are just regular convolutions

Transposed convolution: 1d example using matrix multiplications, stride=2

Normal convolution

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

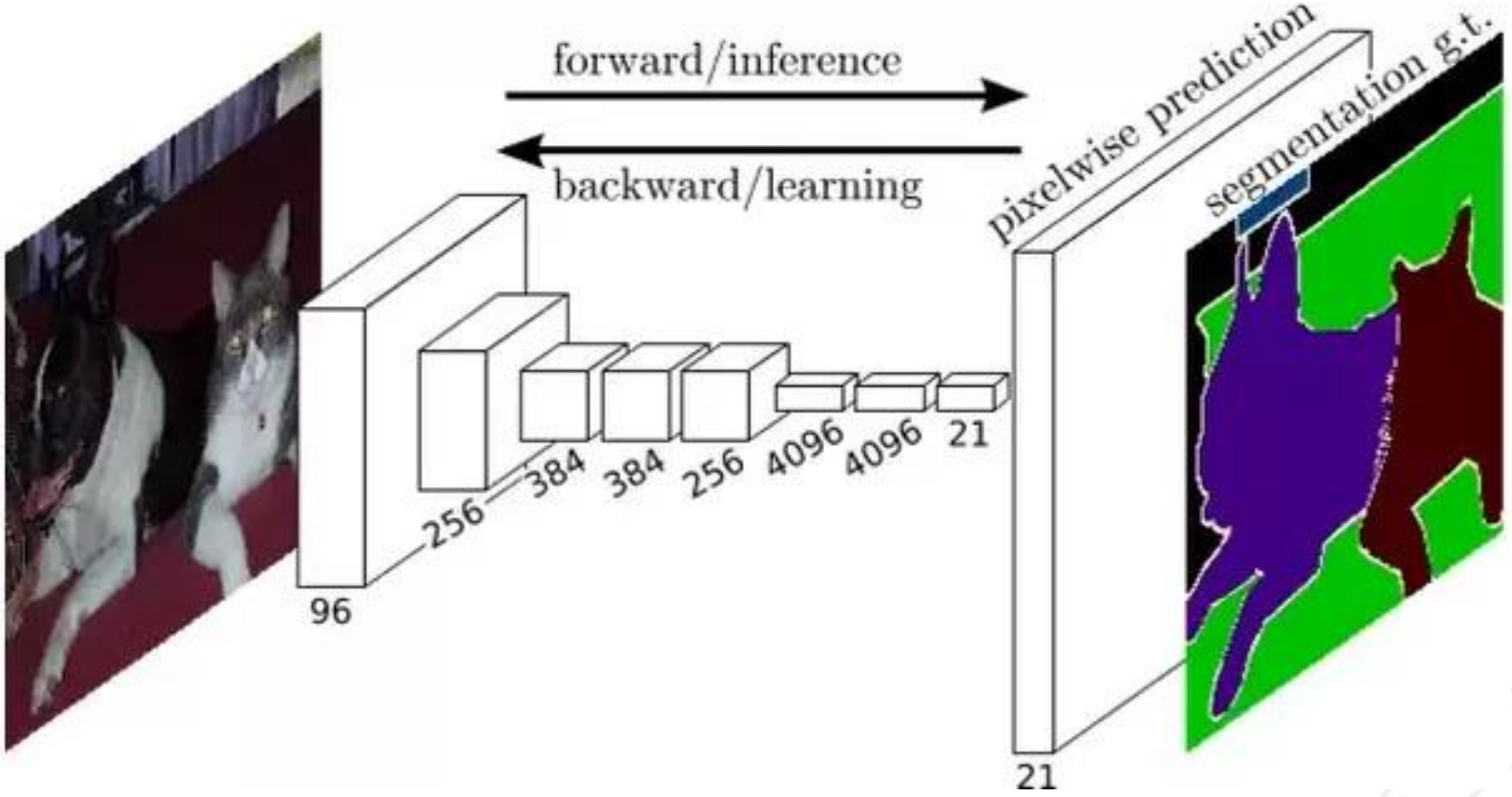
Transposed convolution

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

Normal convolutions with stride = 2 reduce the size of the output.

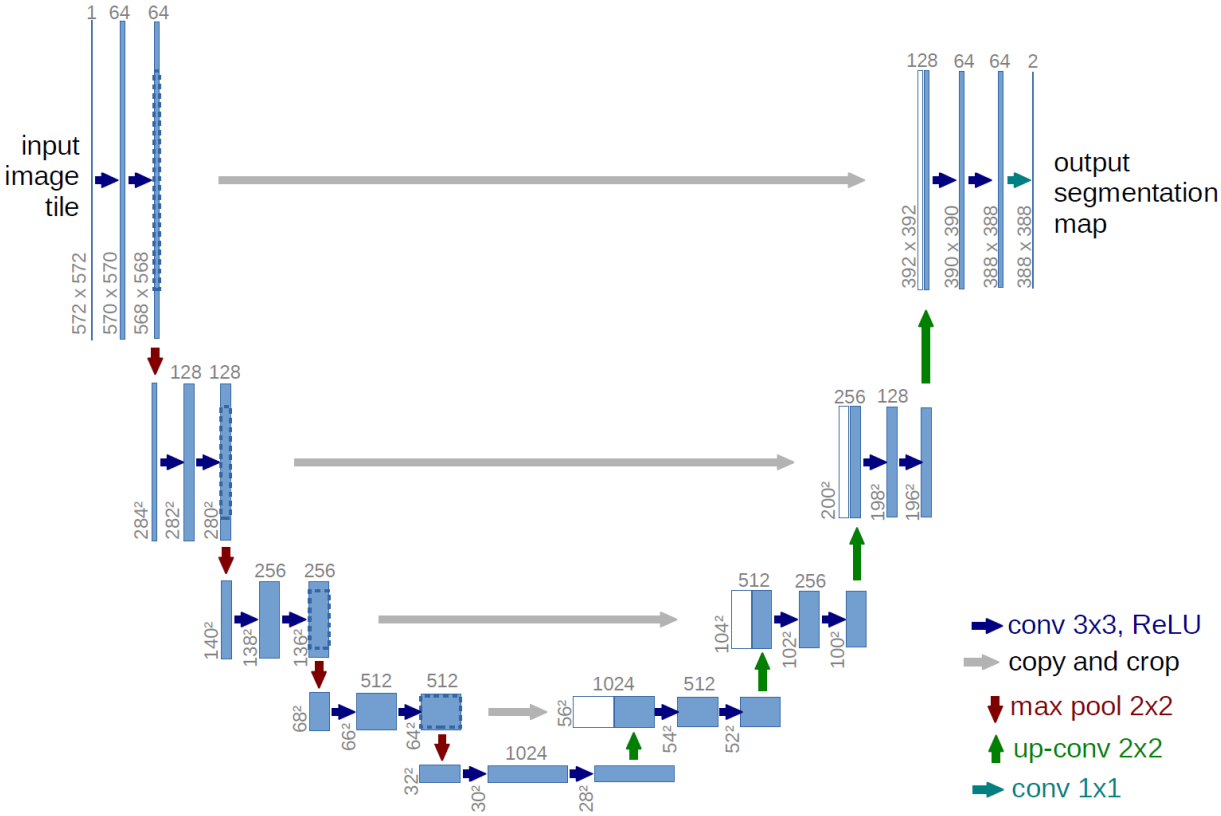
Transposed convolutions with stride = 2 increase the size of the output.

Example: fully convolutional



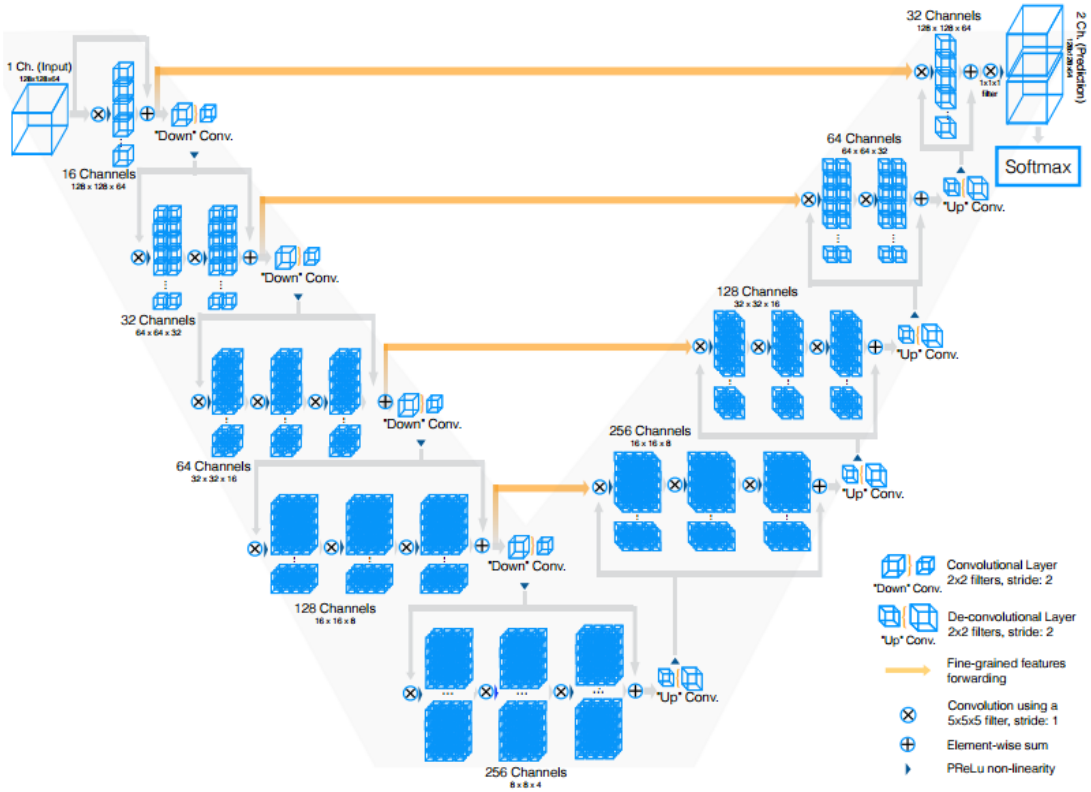
Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).

Example: U-net



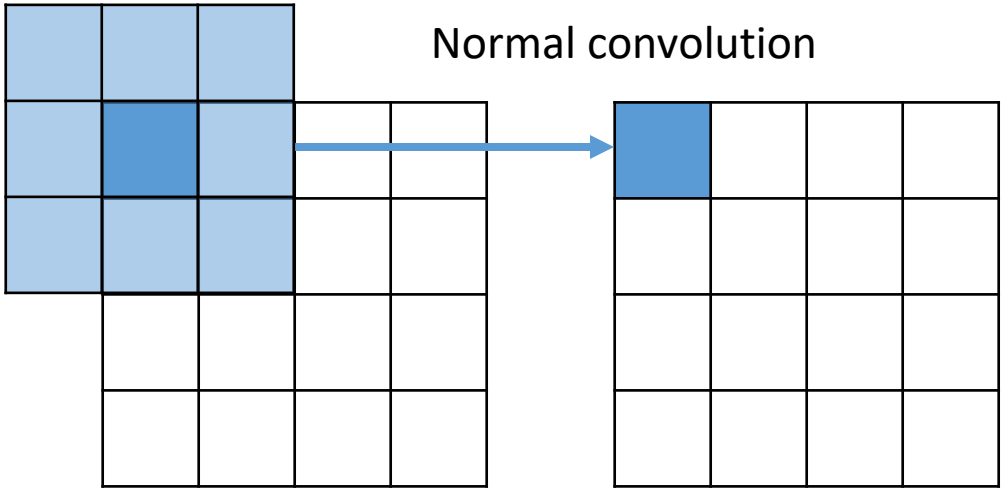
Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.

Example: V-net

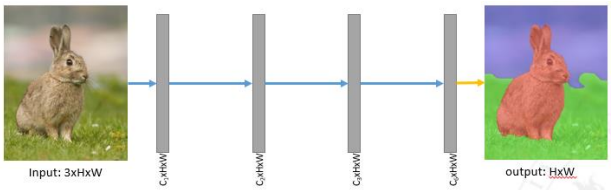


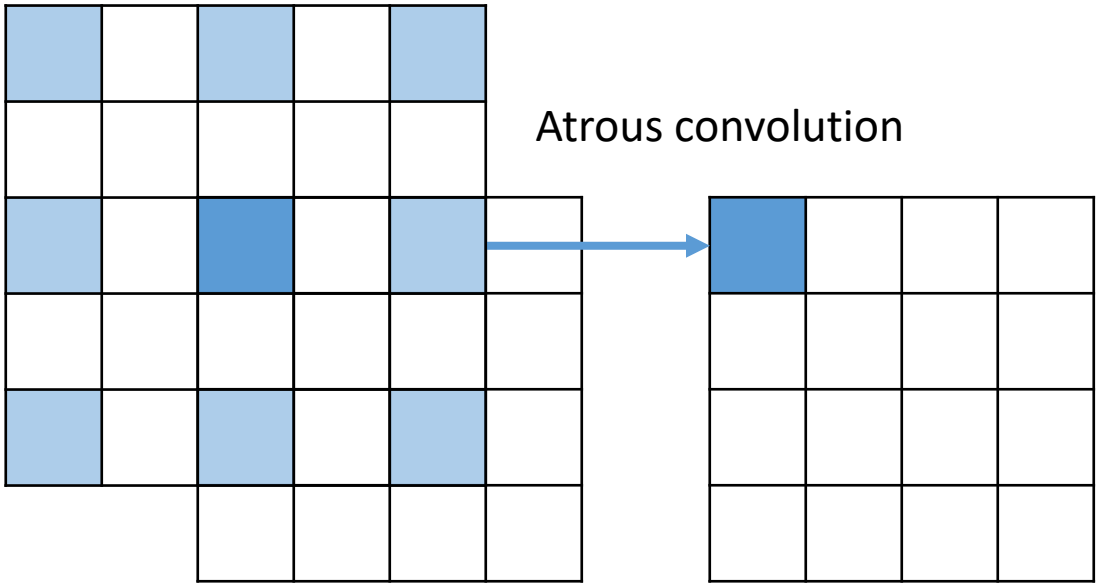
Milletari, F., Navab, N., & Ahmadi, S. A. (2016, October). V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on* (pp. 565-571). IEEE.

Atrous convolution

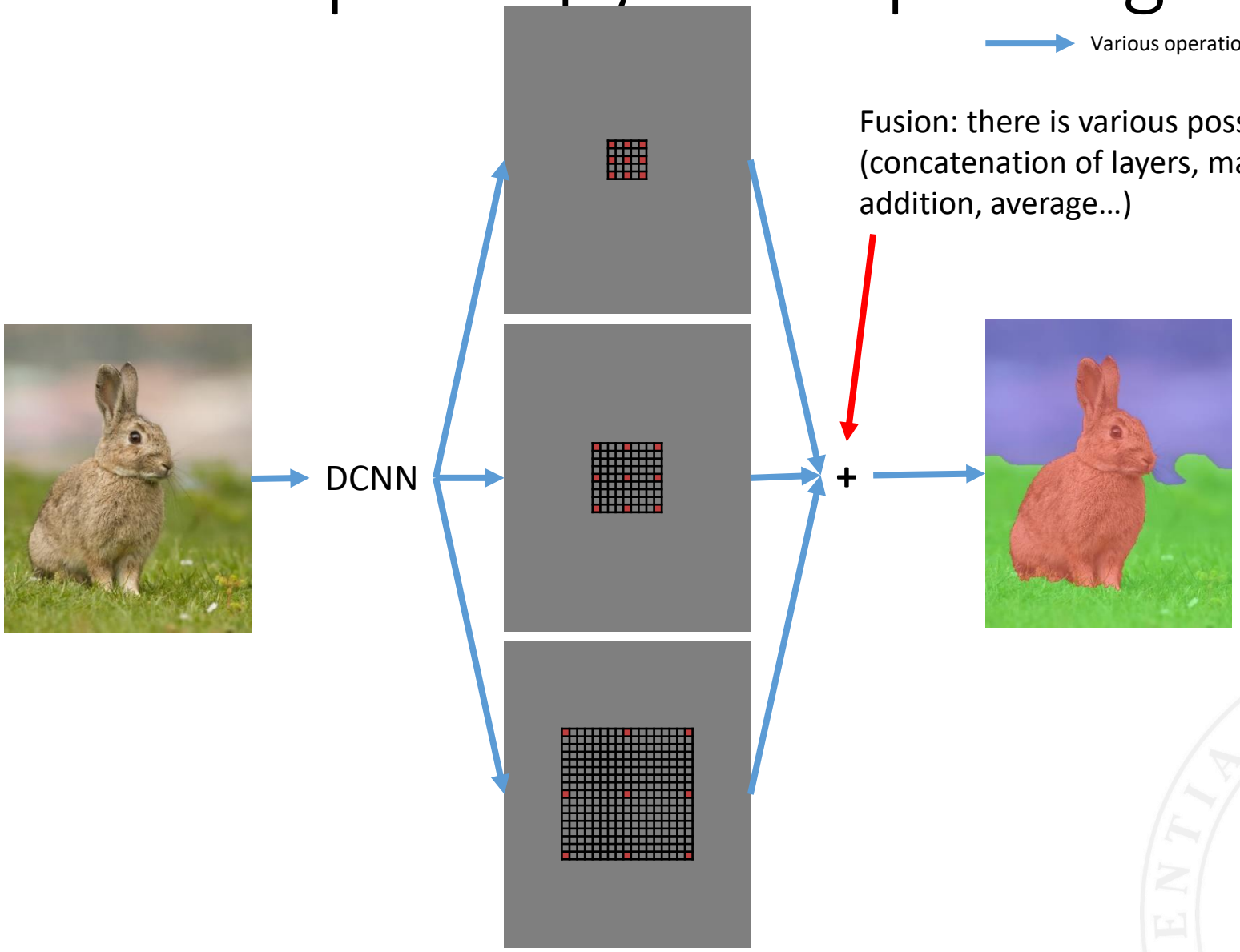


- Sampling region is larger
- Keep the number of parameters
- We do not need to reduce the resolution



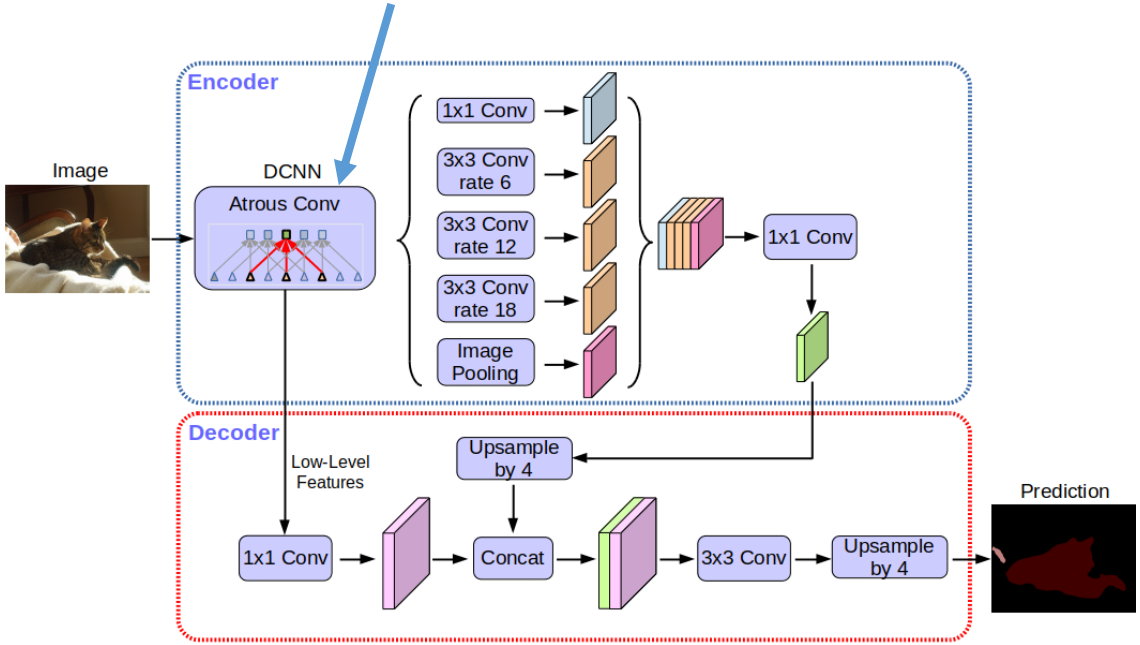


Atrous spatial pyramid pooling



Example: Deeplab

Convolutional neural network (e.g. vgg-16 or ResNet-101) where max-pooling operations have been replaced with atrous convolutions.



Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, *40*(4), 834-848.

Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *arXiv preprint arXiv:1802.02611*.

Guidelines

- Most used upsampling nowadays= atrous & transposed convolution
- Gathering information at different resolution is usually a good strategy (e.g. with spatial pyramid pooling, concatenation+1x1 convolution, shortcuts,…)
- Be aware of the size of your network!

Instance segmentation

Each instance of a class is detected separately
Labelling is done at pixel level

Rabbit, Rabbit, Rabbit, Rabbit, Rabbit,
Cat

Example: car detection

Instance segmentation: How?

- Use Post-processing to segment the objects
- Force the network to separate the object
- Detect the borders as a second output to help separate objects
- Mask R-CNN ➔ next lecture

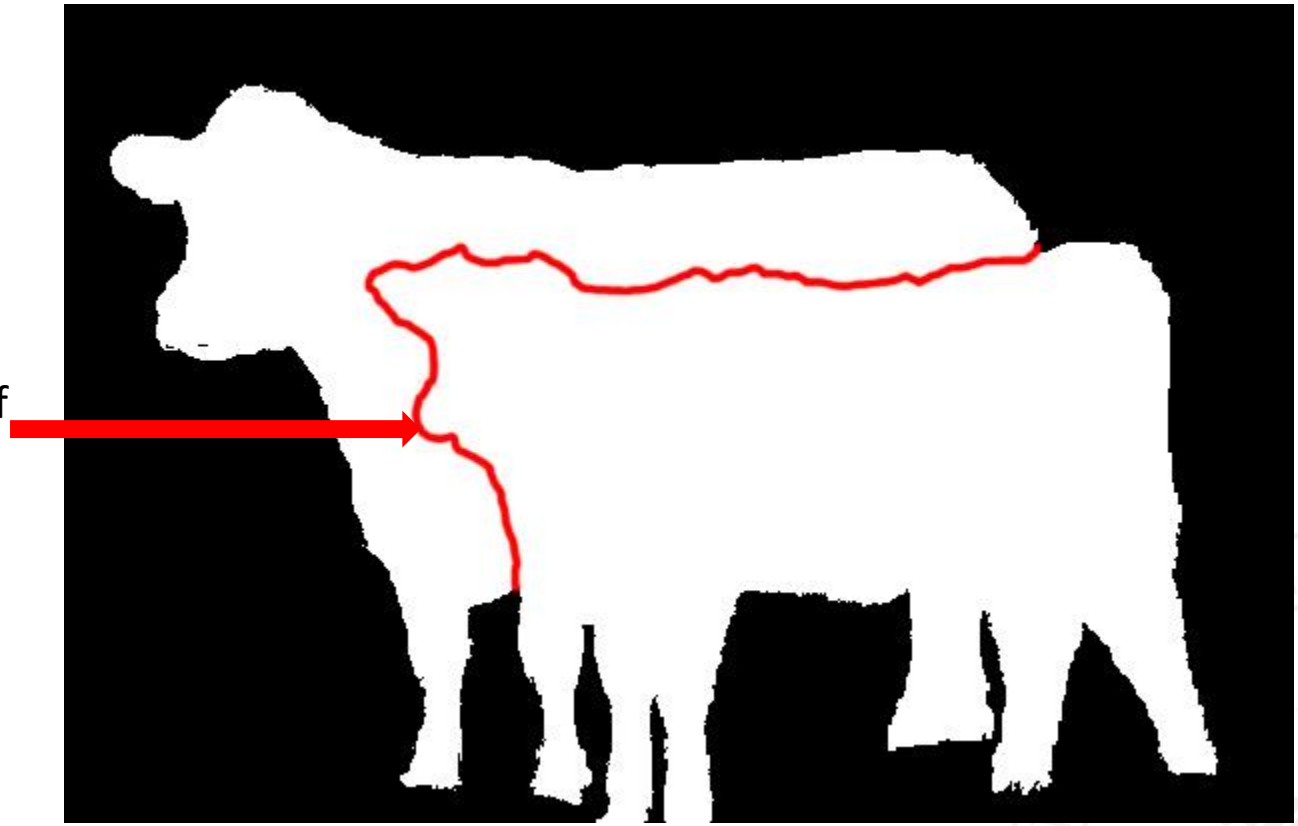
Solution1: Post-processing

Watershed, edge detection,…

- Problem dependent
- Limited

Solution2: Force the network

- Modify the cost function to add penalties to the boundaries

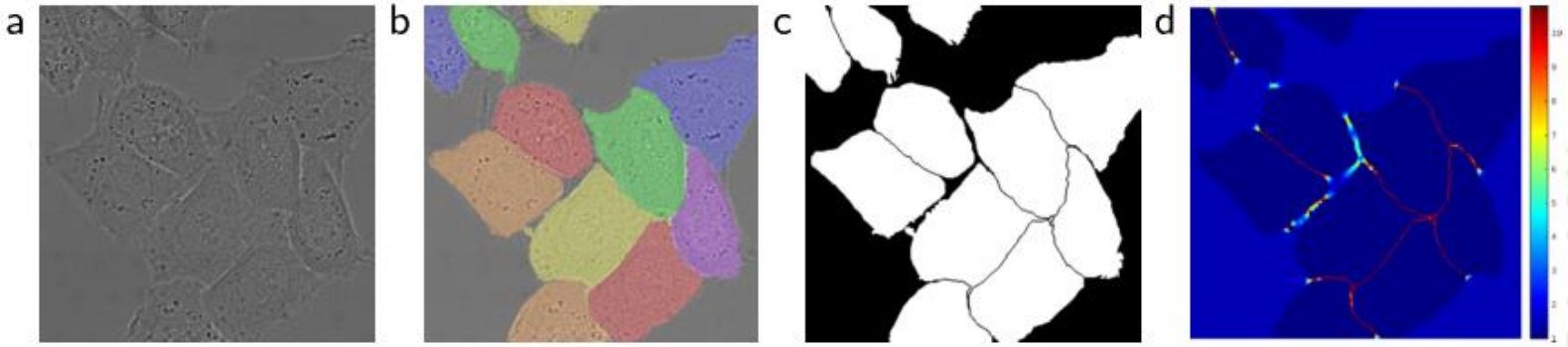


Example for cross-entropy

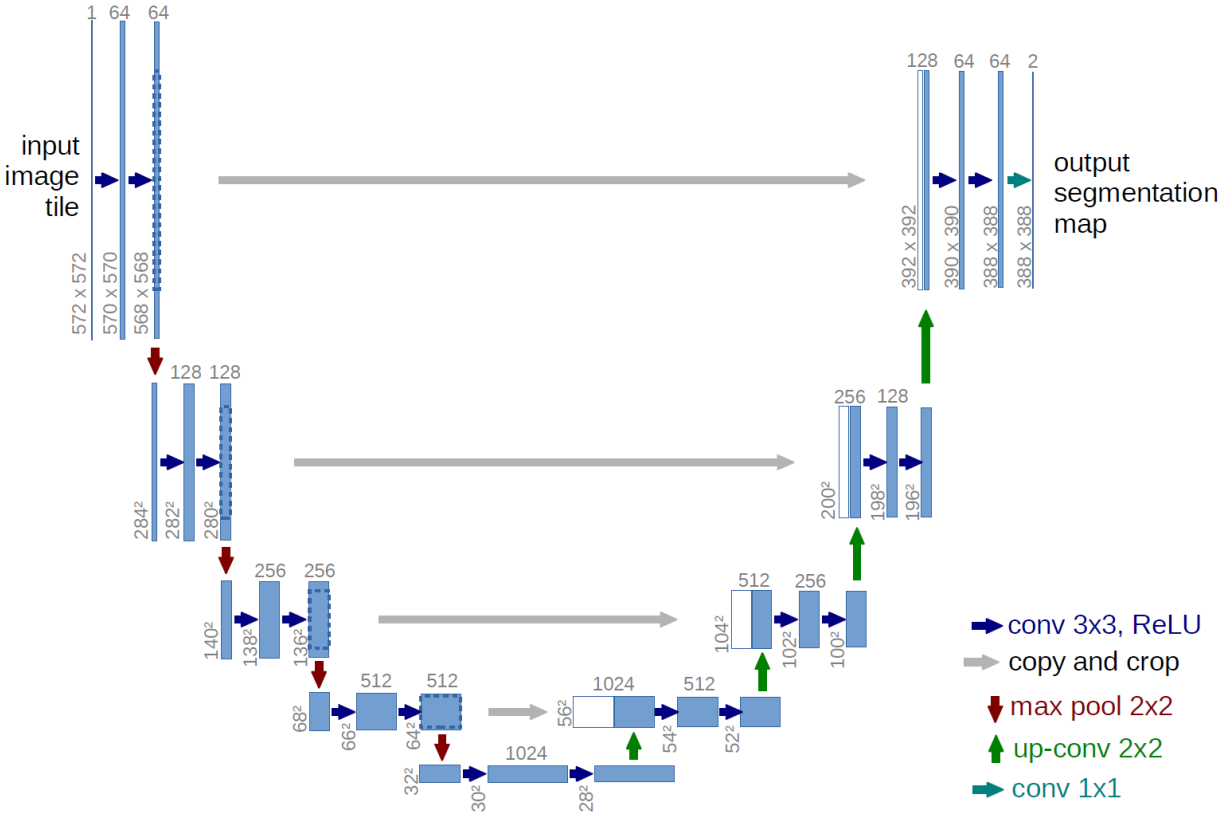
$$cost = -\sum_i Y_i \log(\hat{Y}_i)$$

$$cost = -\sum_i w\,(\mathrm{Y_i}) \log(\hat{Y}_i)$$

$$w(Y_i) = \begin{cases} kY_i & \text{If pixel } i \text{ is at less than } l \text{ pixels from a border} \\ Y_i & \text{Otherwise} \end{cases}$$

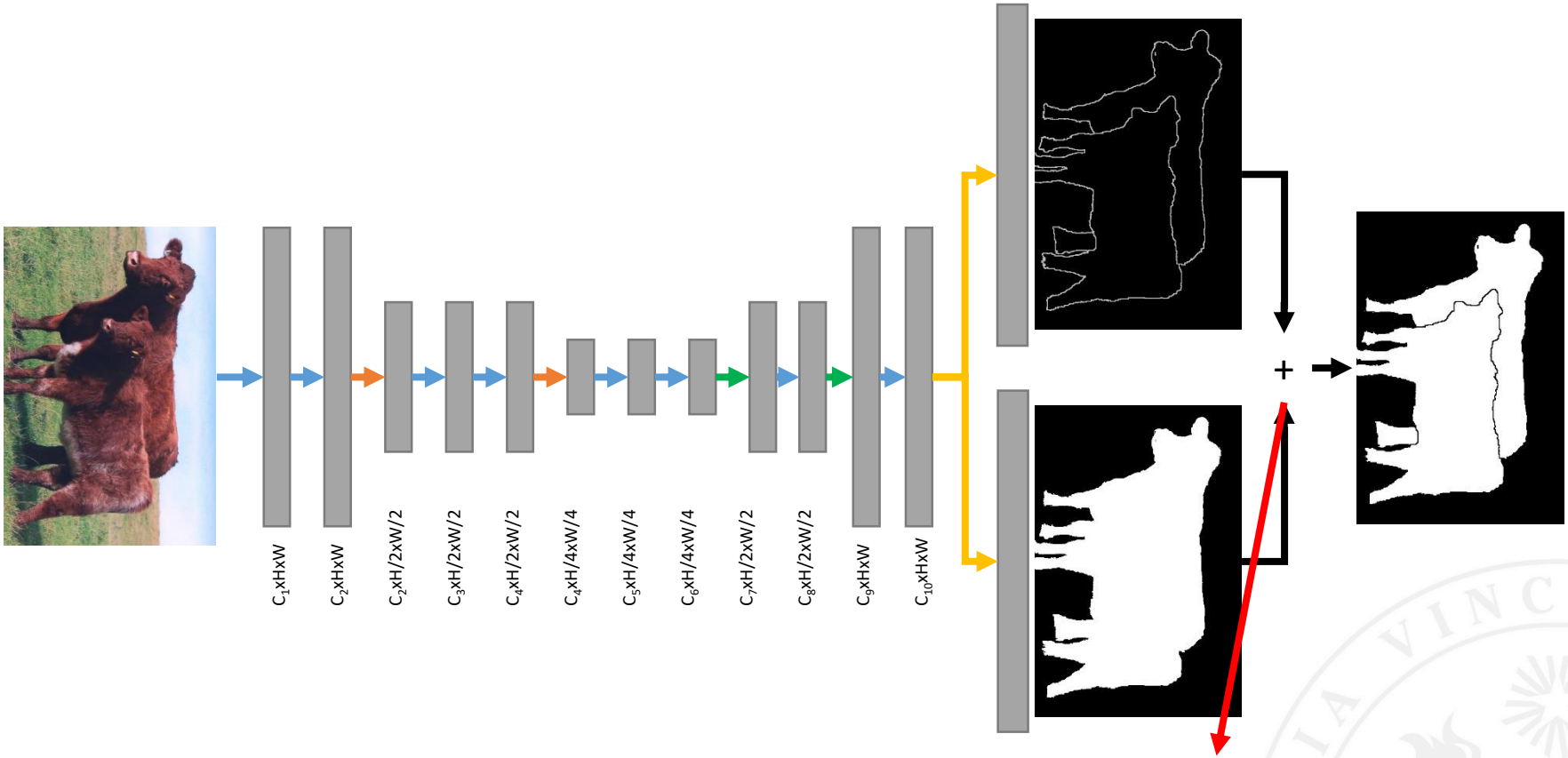


Example: U-net



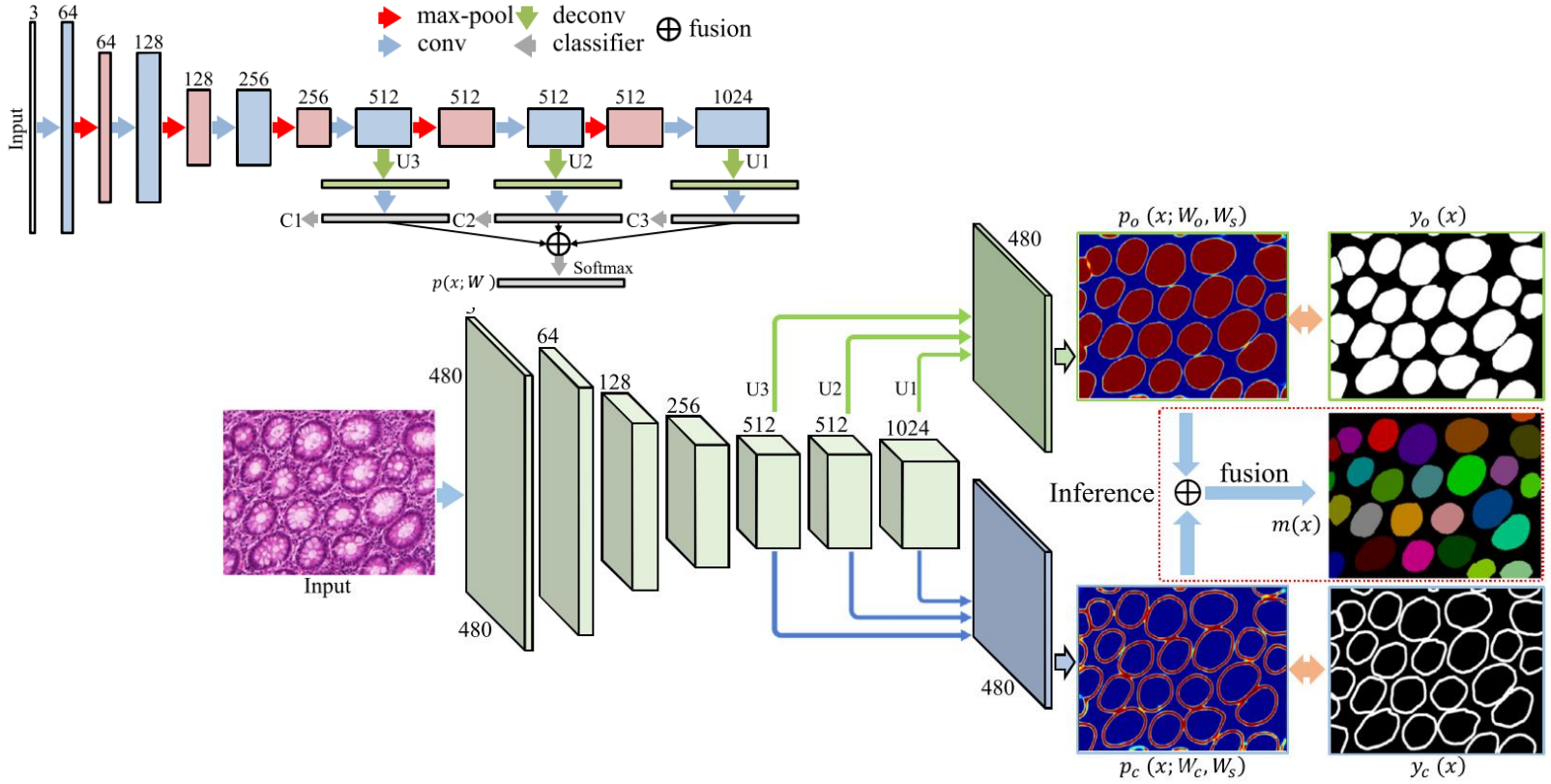
Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.

Solution 3: Detect the borders as a second output to help separate objects



Fusion = difference + postprocessing

Example: DCAN



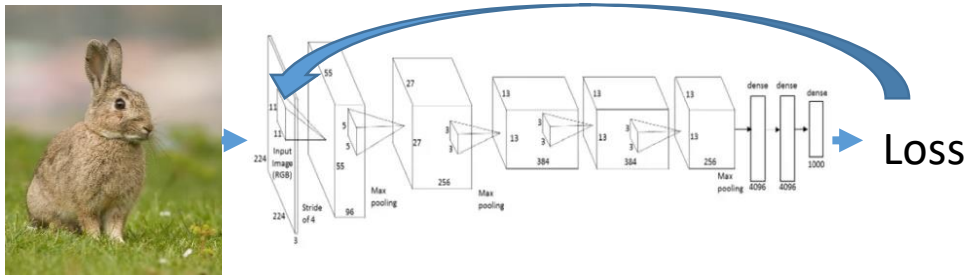
Chen, H., Qi, X., Yu, L., & Heng, P. A. (2016). Dcan: Deep contour-aware networks for accurate gland segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 2487-2496).

Guidelines

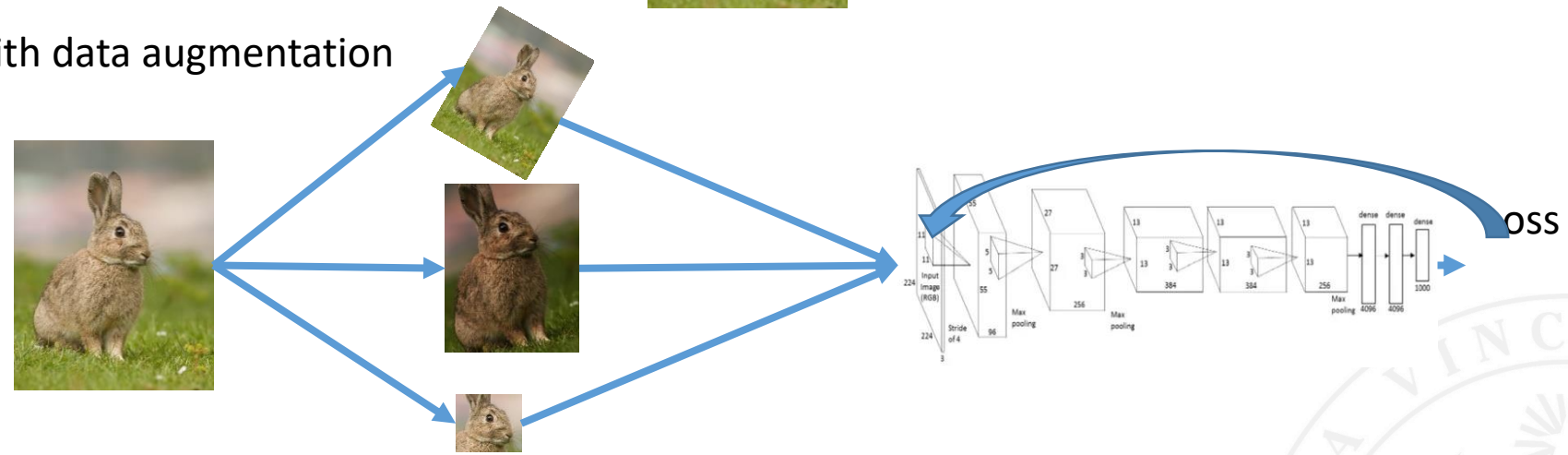
- You can mix the strategy! E.g. modifying the cost function + detecting borders + post-processing.
- Be aware of the metaparameters introduced (border width when detecting borders, additional constants in the cost function,…)

Data augmentation: training on transformed data

Without data augmentation



With data augmentation



Rotation, change in brightness, contrast,…

Why?

- Lack of data
- Desire to takes into account variations not present in the dataset used
- Help to avoid overfitting
- You can easily win 5 to 10% of accuracy

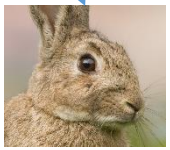
What kind of transformation?

Flip

Rotate

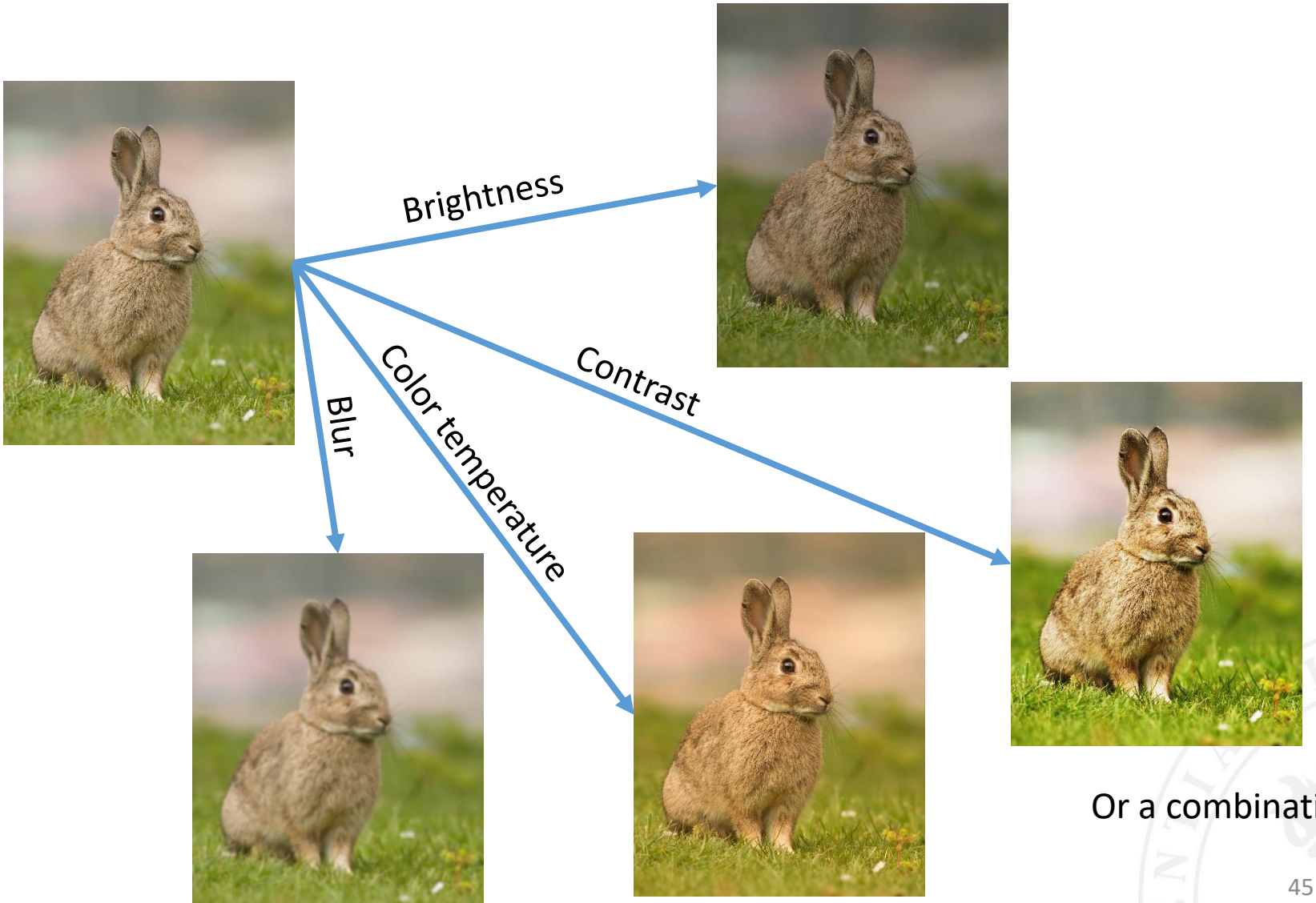
Scale

Crop



Or a combination

What kind of transformation?

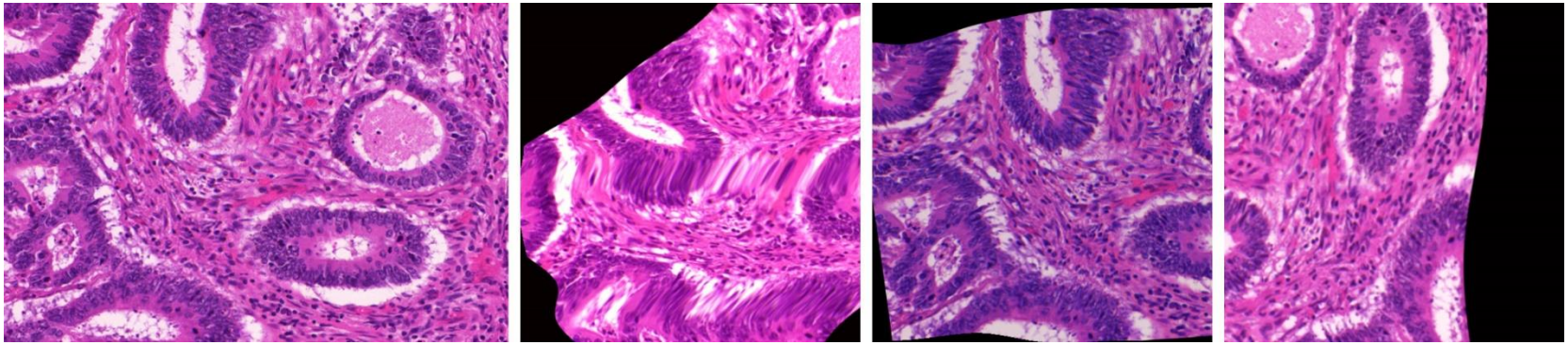


Or a combination

What kind of transformations?

- Lens distortions
- Translations, Stretch, Skew, Shear, perspective, Elastic deformations,…
- Noise addition
- Other changes in lighting or color conditions
- Use of a network that mimic learned deformations
- …

Example: biomedical images



Guidelines

- Mix the transforms!
- Only introduce transforms you will encounter after training.
- You can introduce specific transforms for specific purpose (biomedical images vs street images,…)