Image Segmentation

Dimitrios Papadopoulos
Associate Professor, DTU Compute
Edited by: J. Miguel Valverde
Postdoc, DTU Compute

Schedule for Part 3: Image segmentation

Part 3: Segmentation

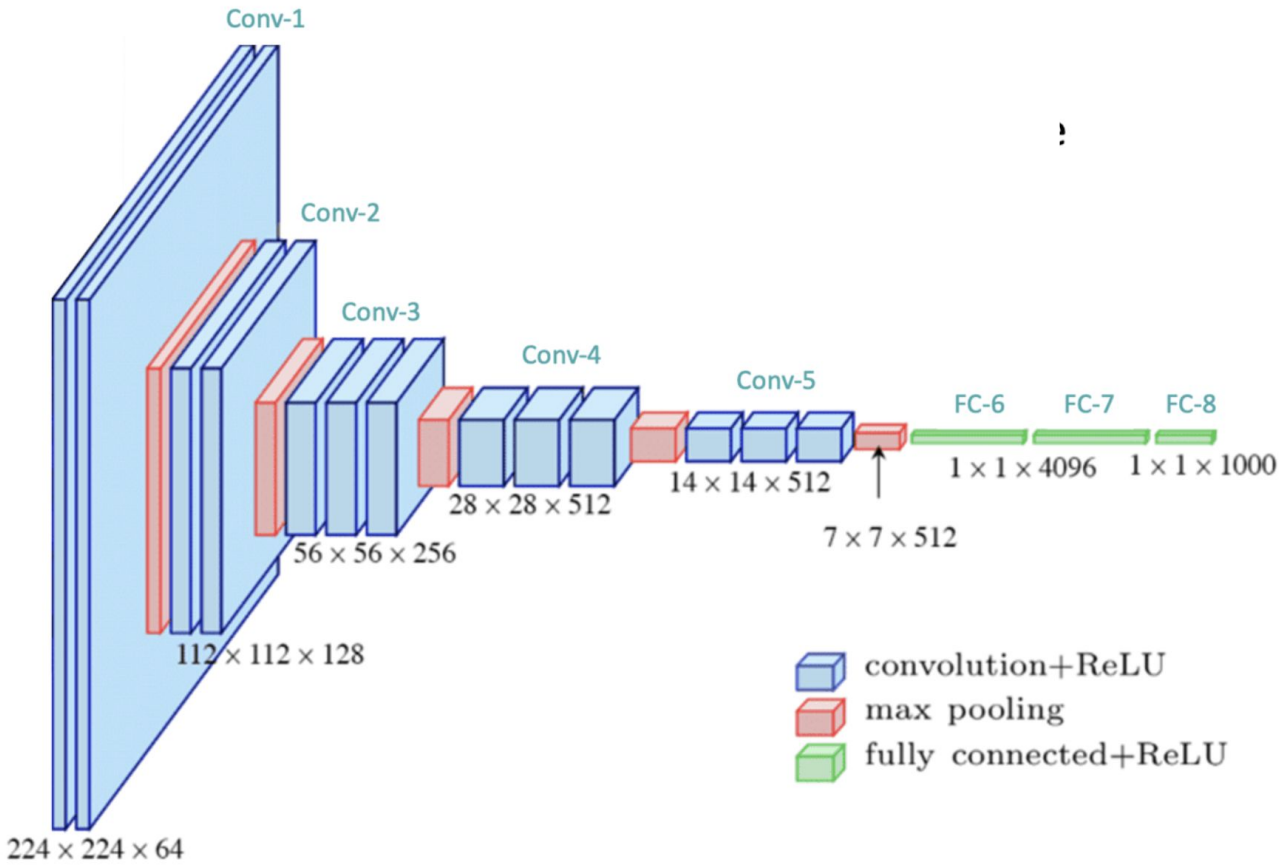
Wednesday 29.10	13:00-15:00 Lecture - Segmentation 15:00-17:00 Exercises	Miguel 4 TAs (Carl, Julius, Christian, Changlu)
Wednesday 5.11	13:00-14:00 Lecture - Semantic segmentation with limited labels 14:00-17:00 Exercises	Miguel 4 TAs (Carl, Julius, Christian, Changlu)
Wednesday 12.11	13:00-15:00 Lecture - Multi-class segmentation - Segmentation (nnUnet, Segment Anything) 15:00-17:00 Exercises	Miguel 4 TAs (Carl, Julius, Christian, Changlu)
Sunday 16.11	**Project report submission at 20:00**	Corrections: Carl, Julius, Christian, Changlu

Lecture's learning goals

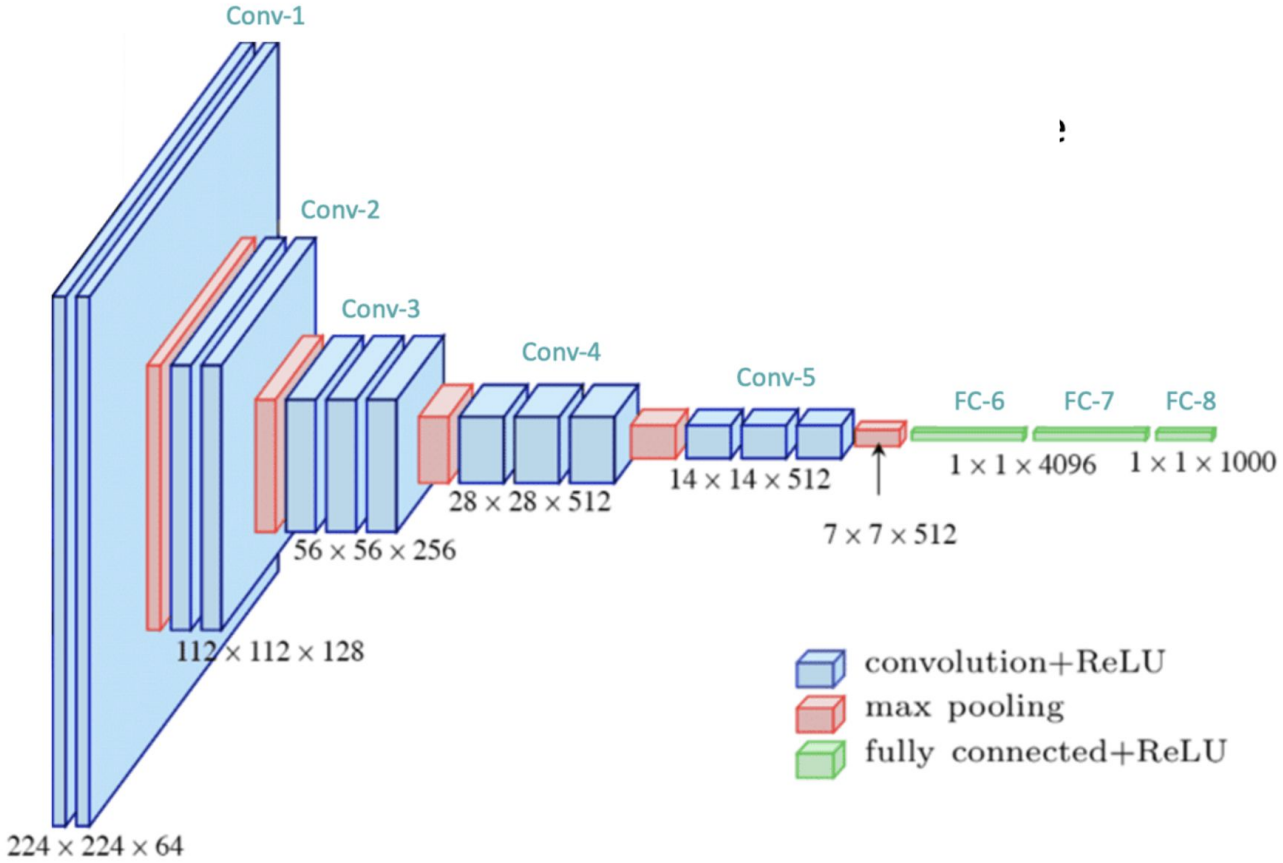
After this lecture you should

- Know what image segmentation is, along with some of its applications
- Be familiar with challenges associated with image segmentation
- Be able to implement and validate a CNN for image segmentation
- Be able to implement your own U-net architecture
- Be able to reason about the U-net's construction and the roles played by its different parts

So far, you have learned…

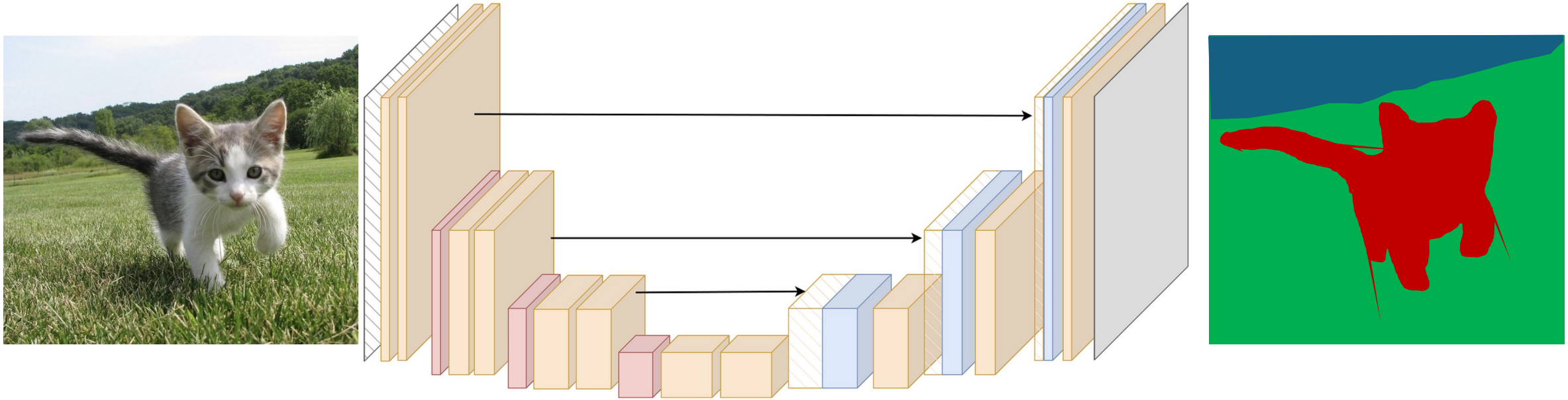


So far, you have learned…

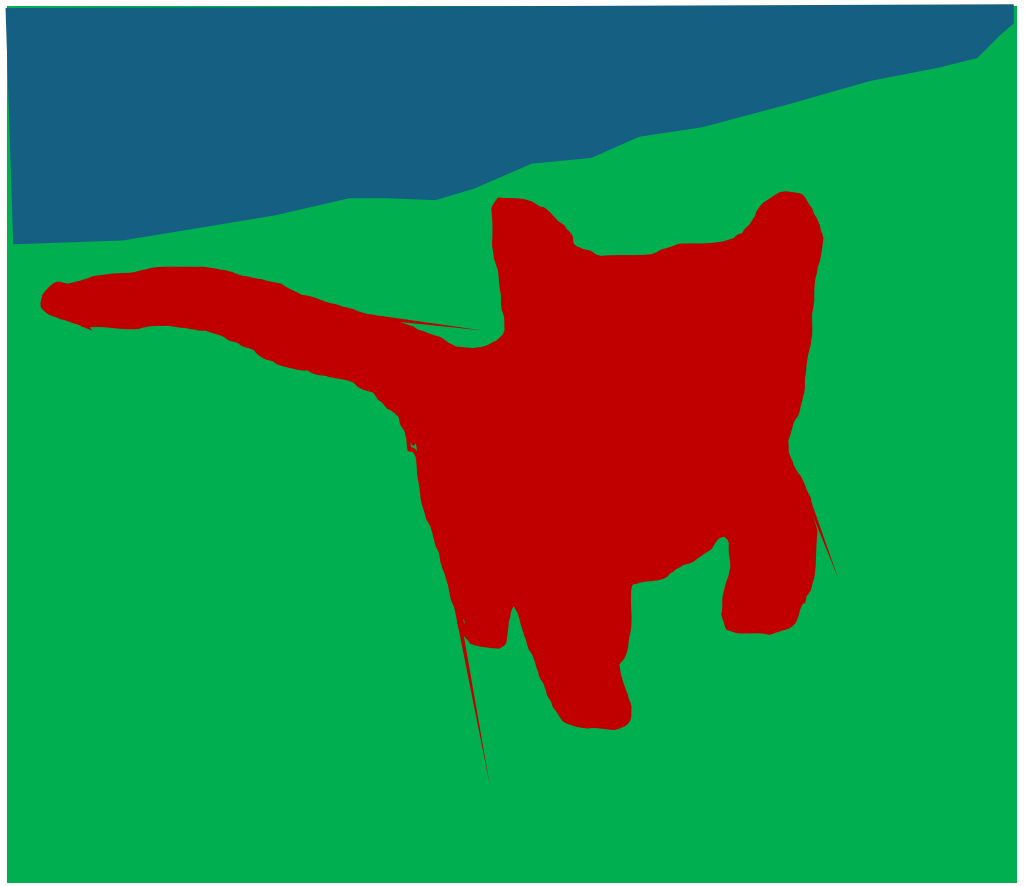


cat

Image segmentation



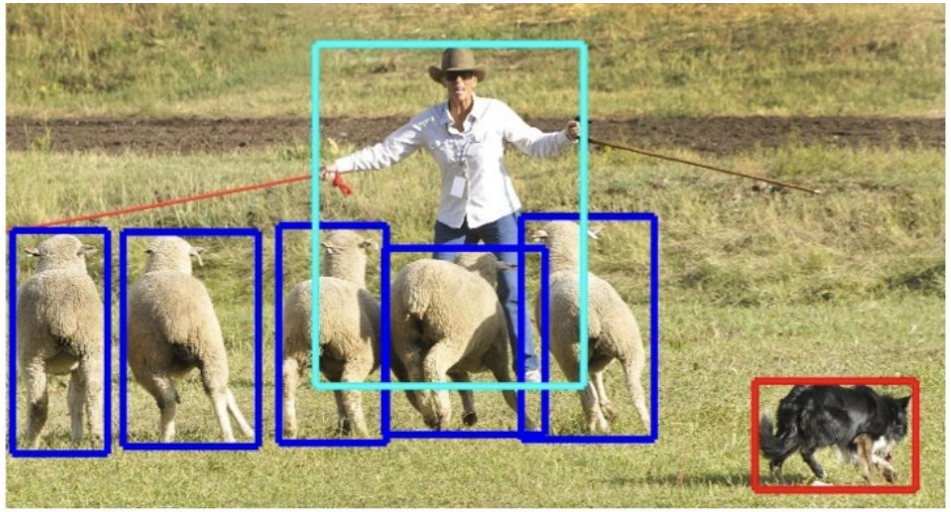
What is image segmentation?



Label each pixel in the image with a category label

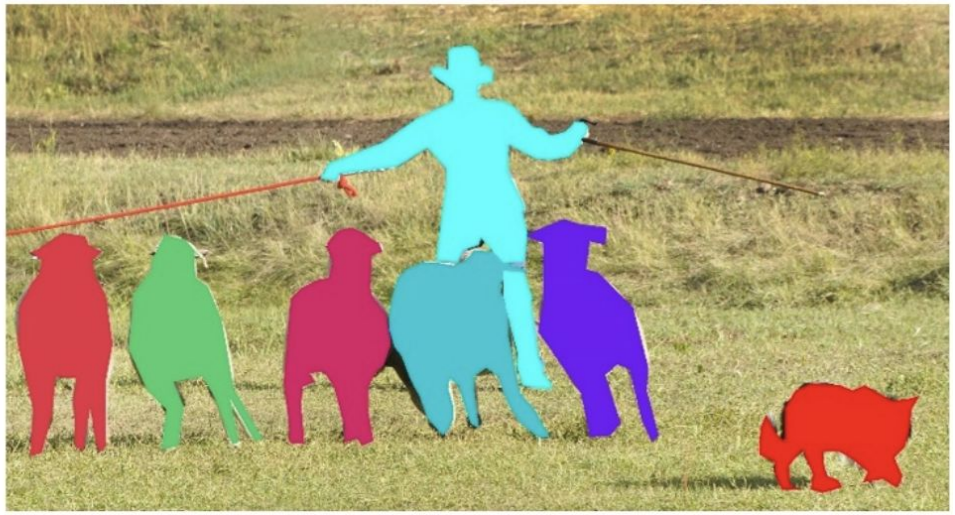
What is image segmentation?

Image classification



Object detection

Semantic segmentation



Instance segmentation

What is image segmentation?

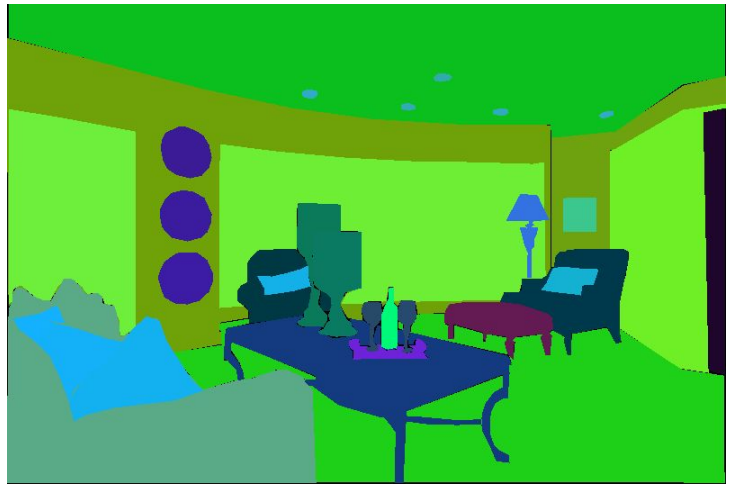
We will focus on the semantic segmentation task

Semantic segmentation

What is image segmentation useful for?

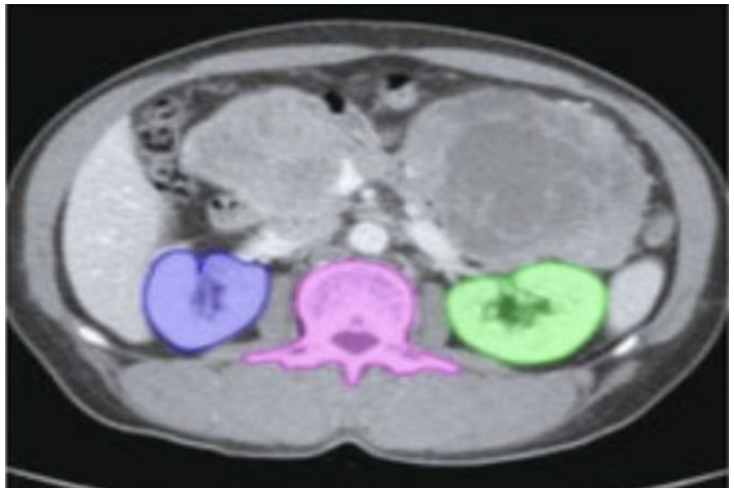
What is image segmentation useful for?

Scene understanding

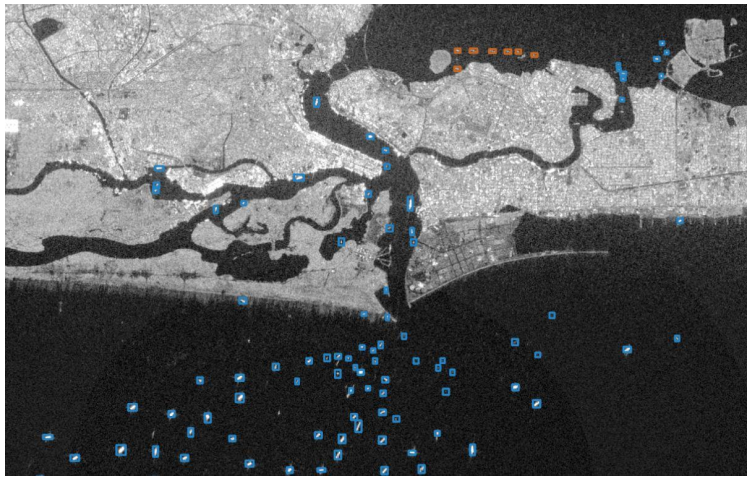


Autonomous driving

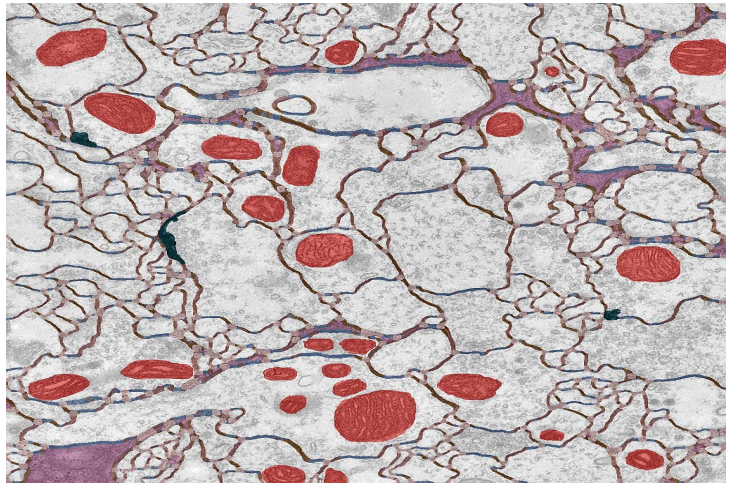
Medical imaging



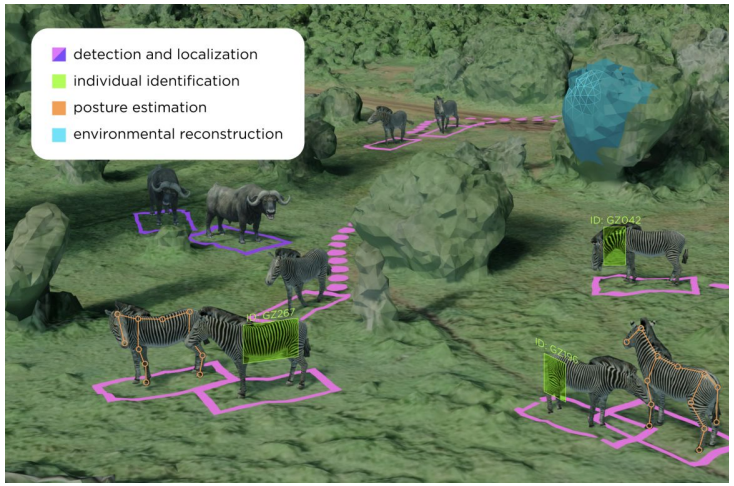
Remote sensing



Microscopy



Biodiversity/ Ecology



Application: Preprocessing

Brain tissue segmentation: Constraining analysis to relevant regions

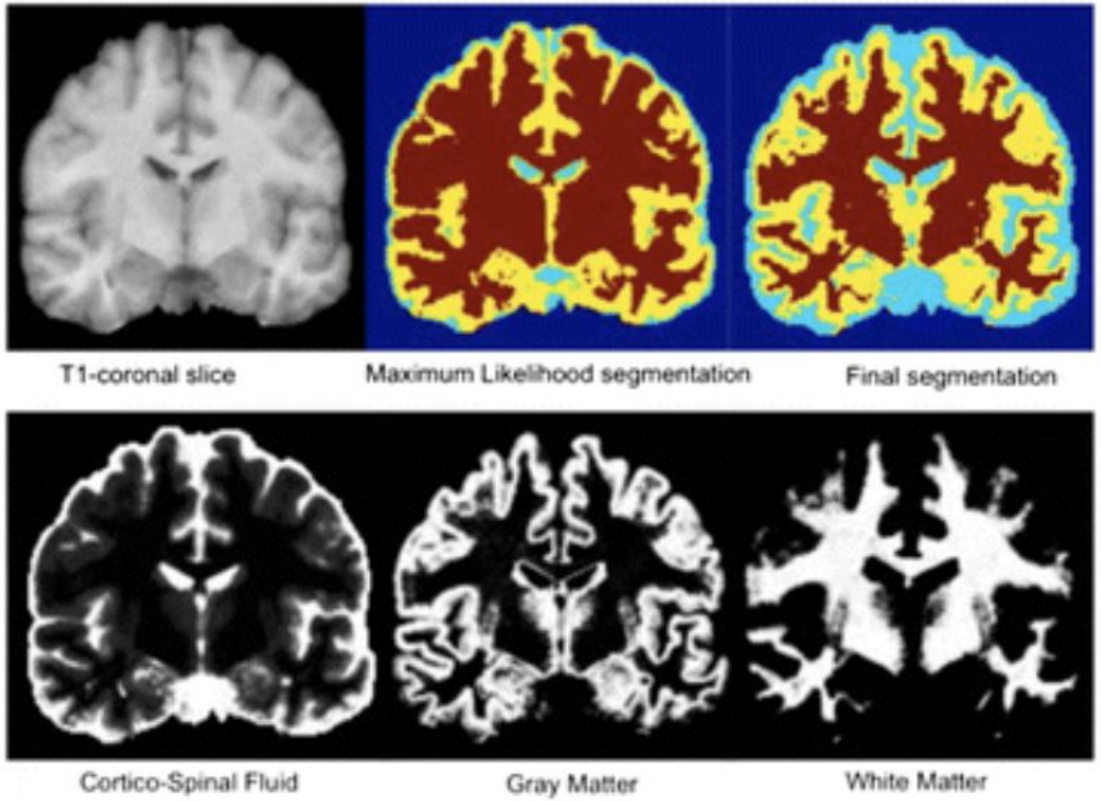


Figure: Figure from Villalon-Reina et al, 2016

Application: Preprocessing

Brain region segmentation: Building a connectivity network

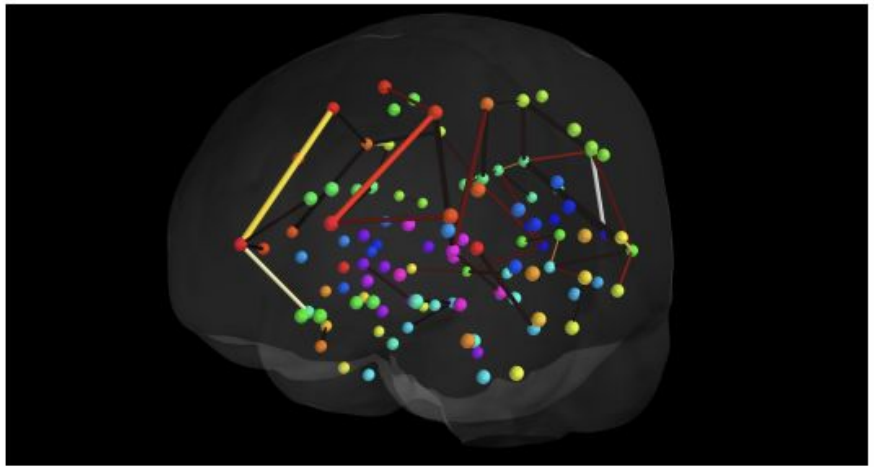


Figure: Left figure from Fan et al, 2016

Application: Predicting disease or focusing treatment

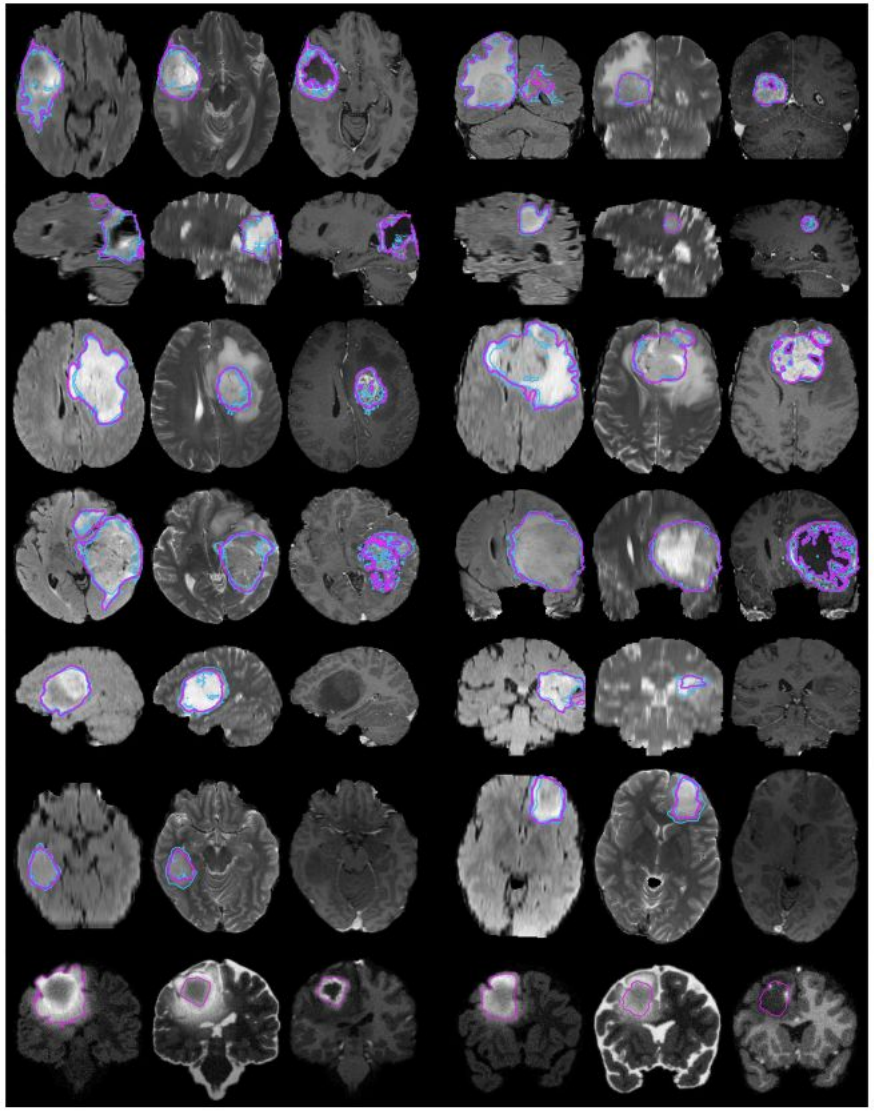
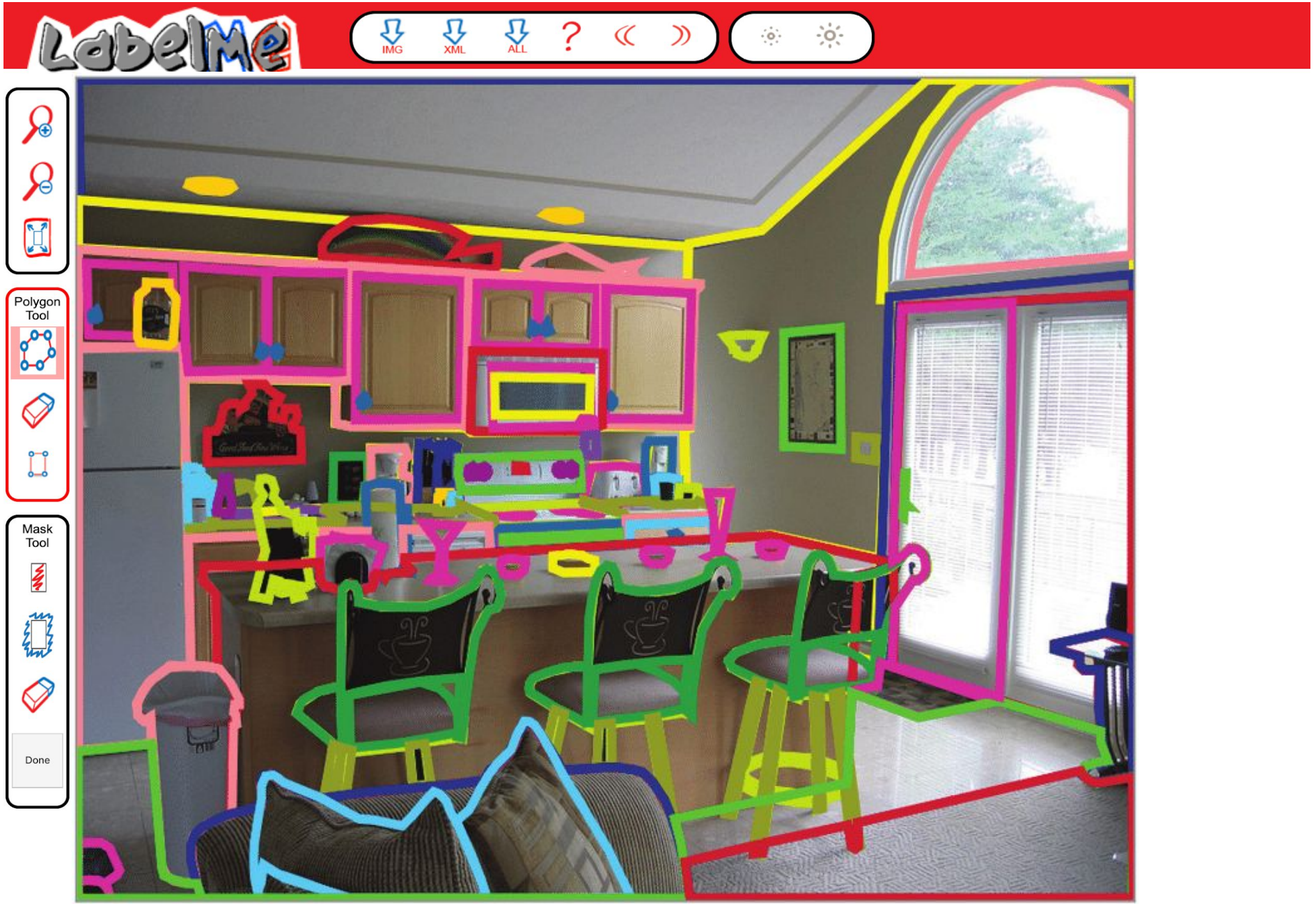


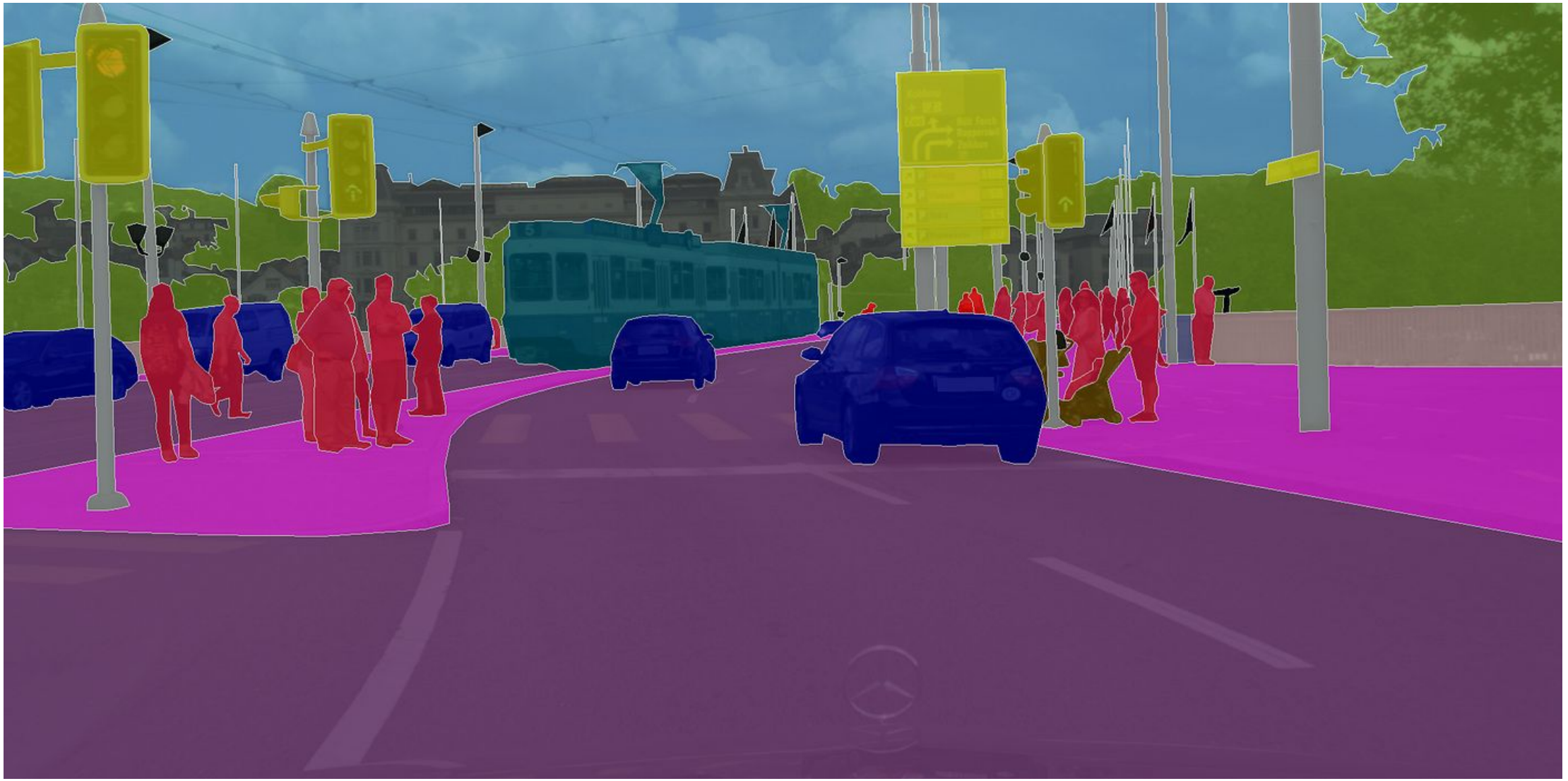
Figure: From the BRATS challenge dataset.

Challenges in image segmentation

Challenges: Annotation (expensive: few labels)

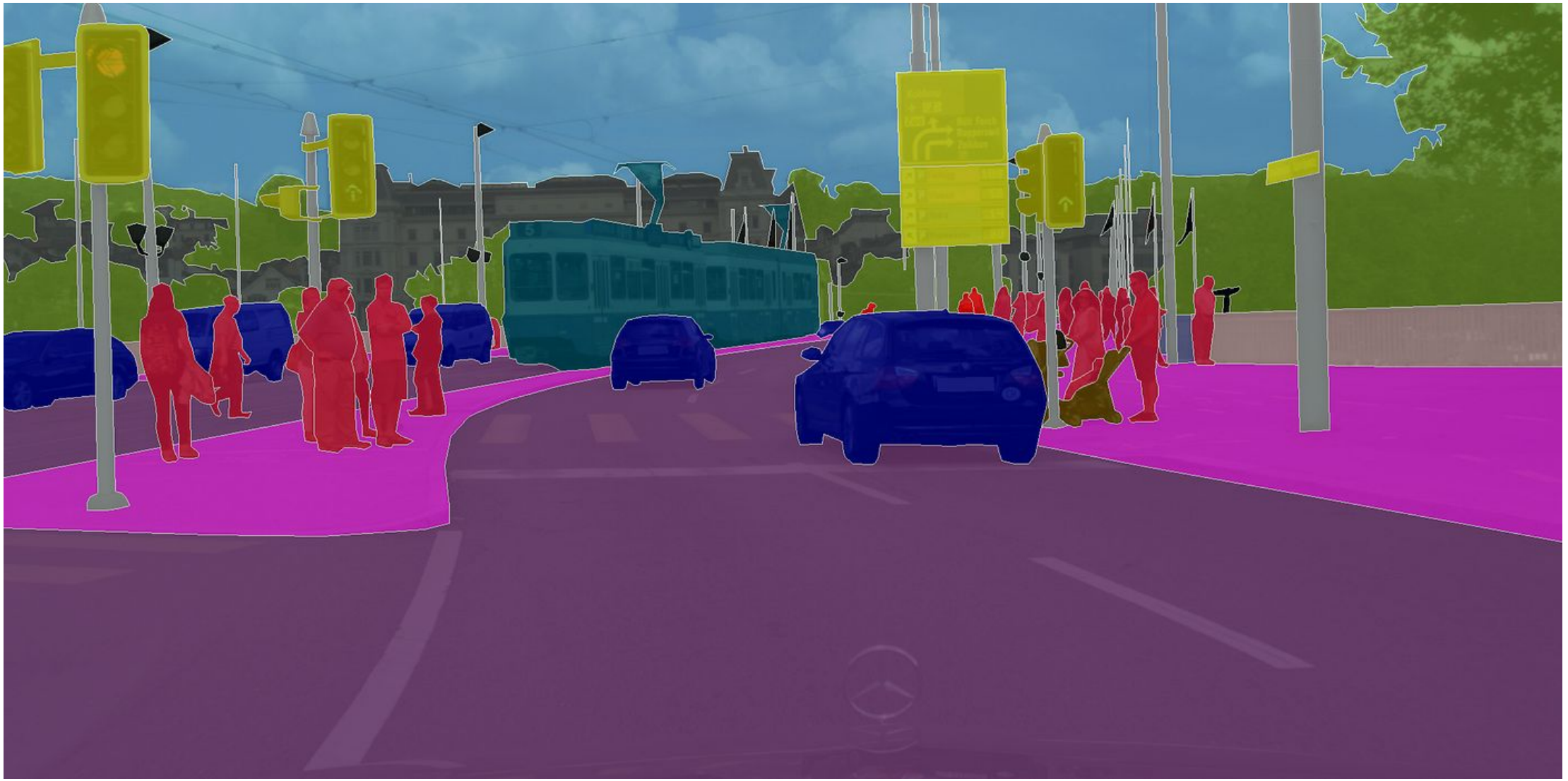


Challenges: Annotation (expensive: few labels)



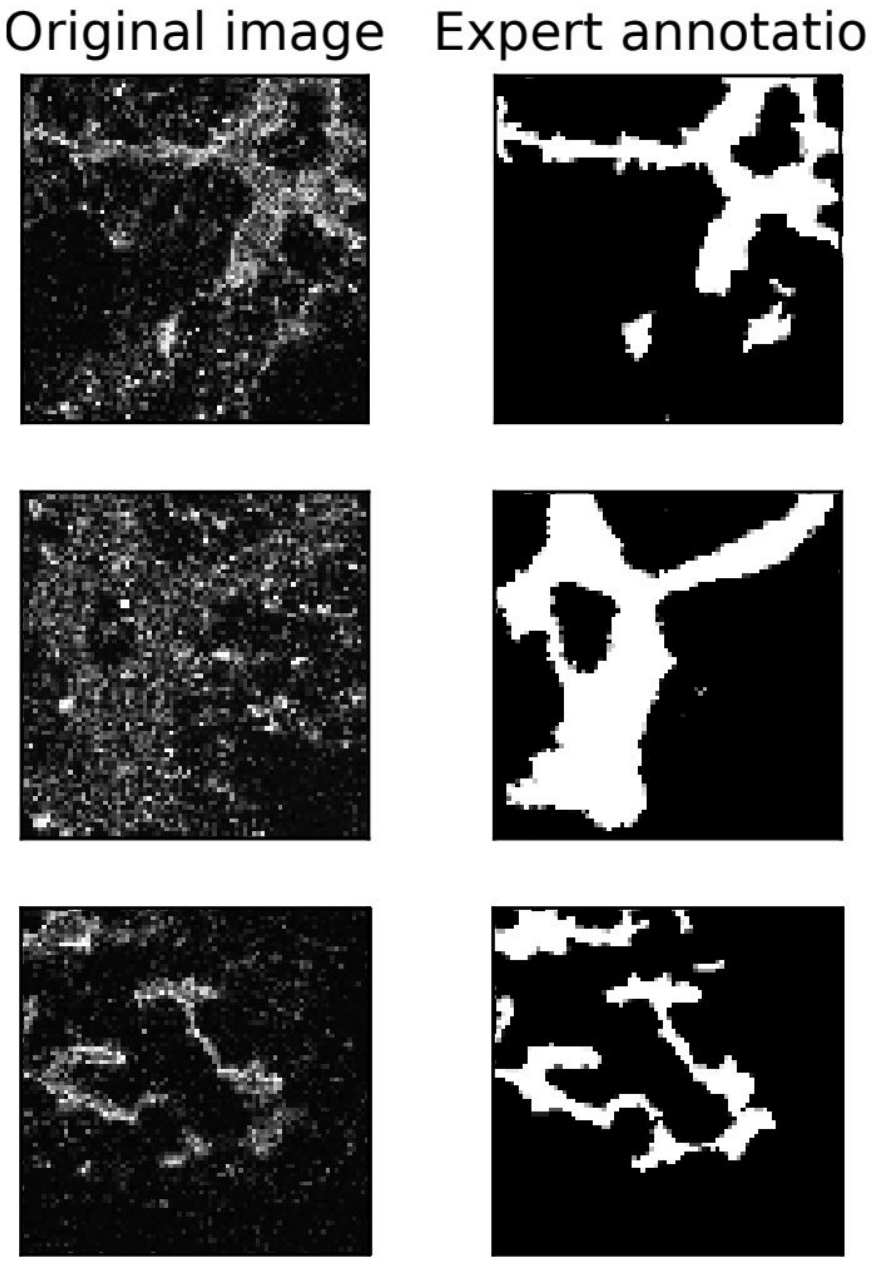
[Image from the Cityscapes dataset]

Challenges: Annotation (expensive: few labels)



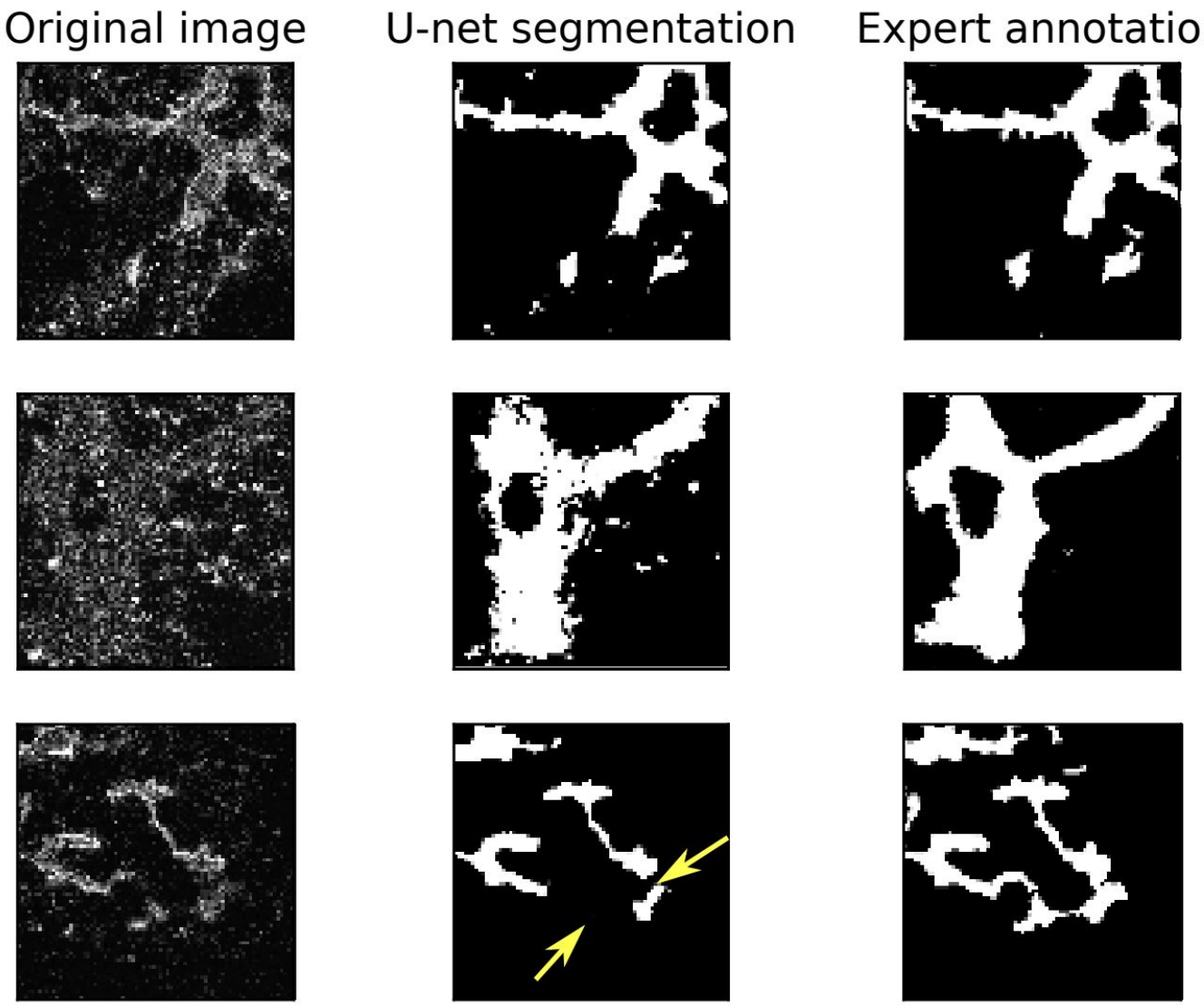
1.5 hours per image

Challenges: Annotation (noisy and few labels)



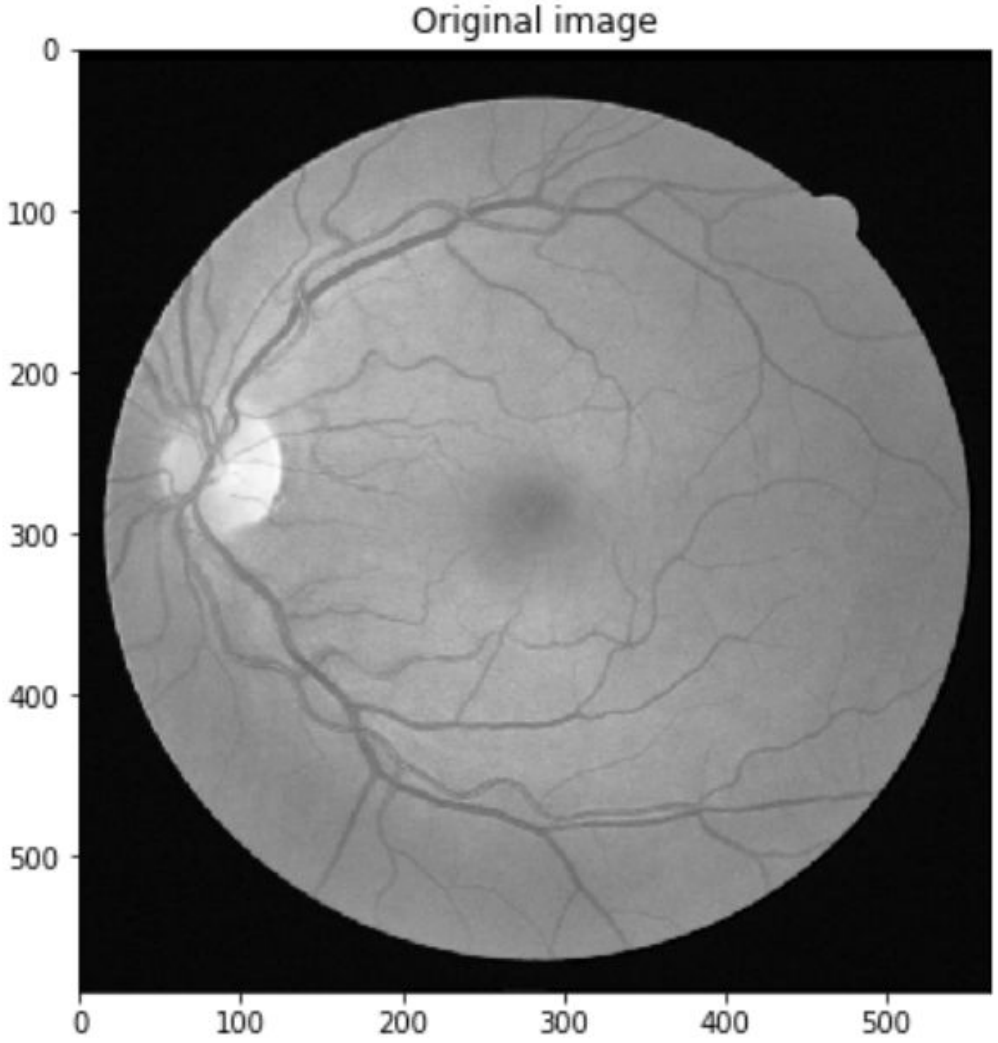
- Require expertise
- Ambigous fuzzy boundaries
- Error-prone annotation Noisy labels

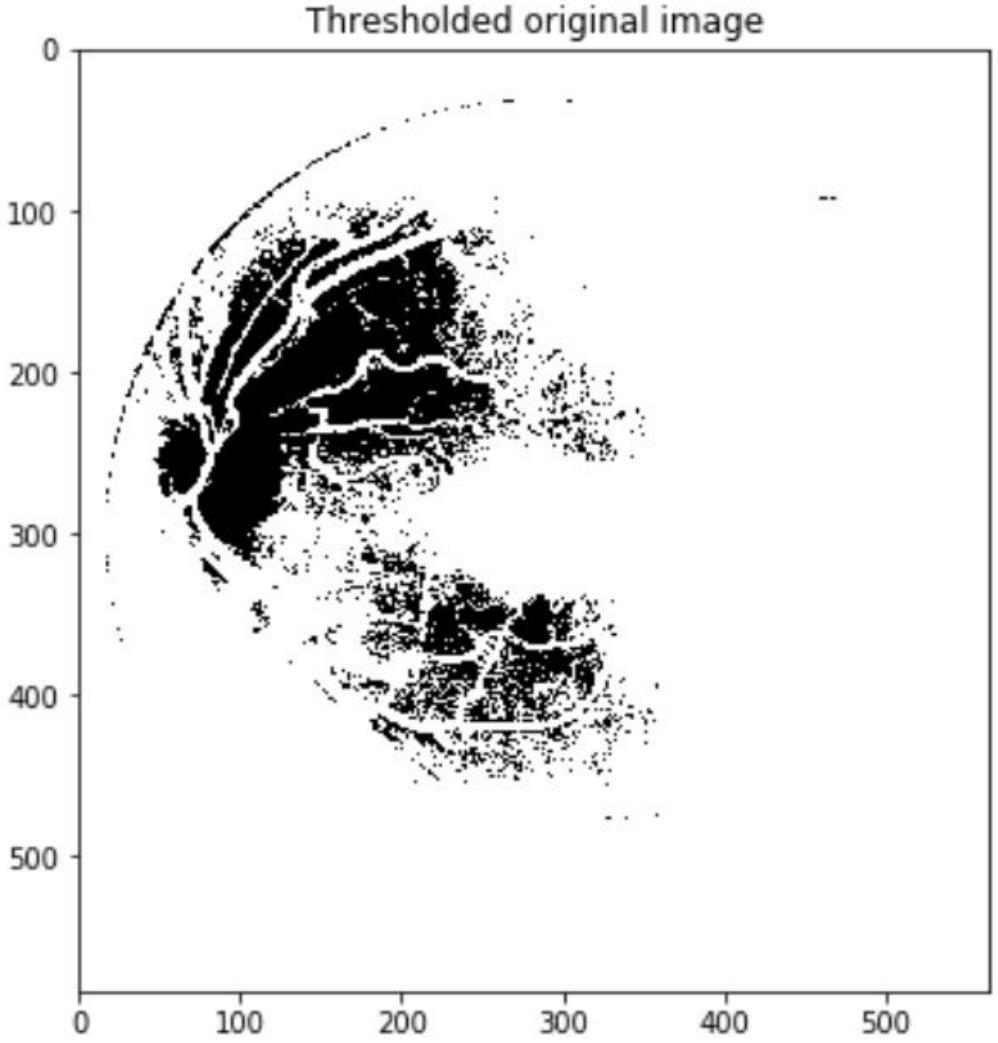
Challenges: Prediction errors



Small pixel error can be large object error

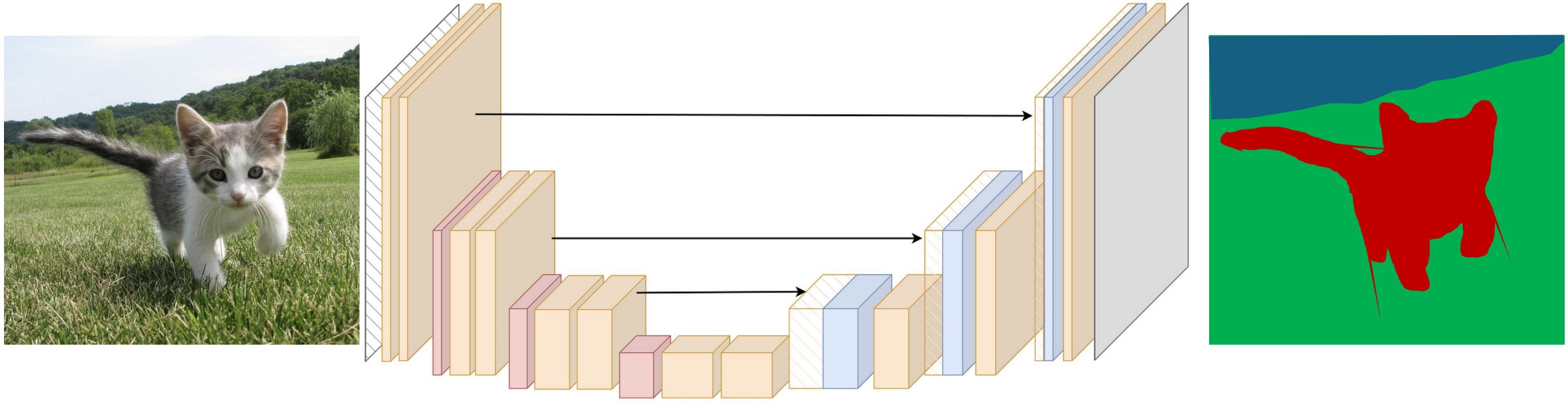
Challenges: Background inhomogeneity





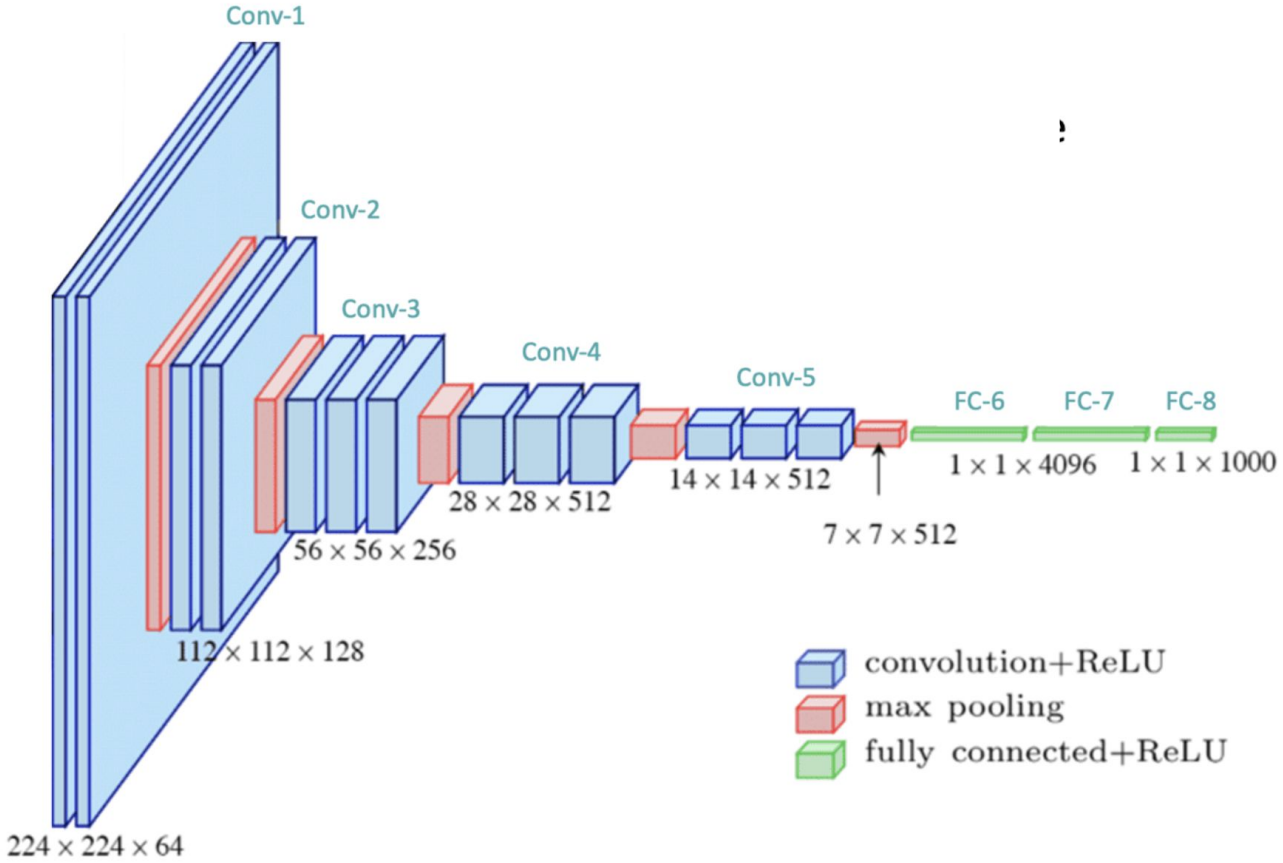
Questions???

CNNs for segmentation



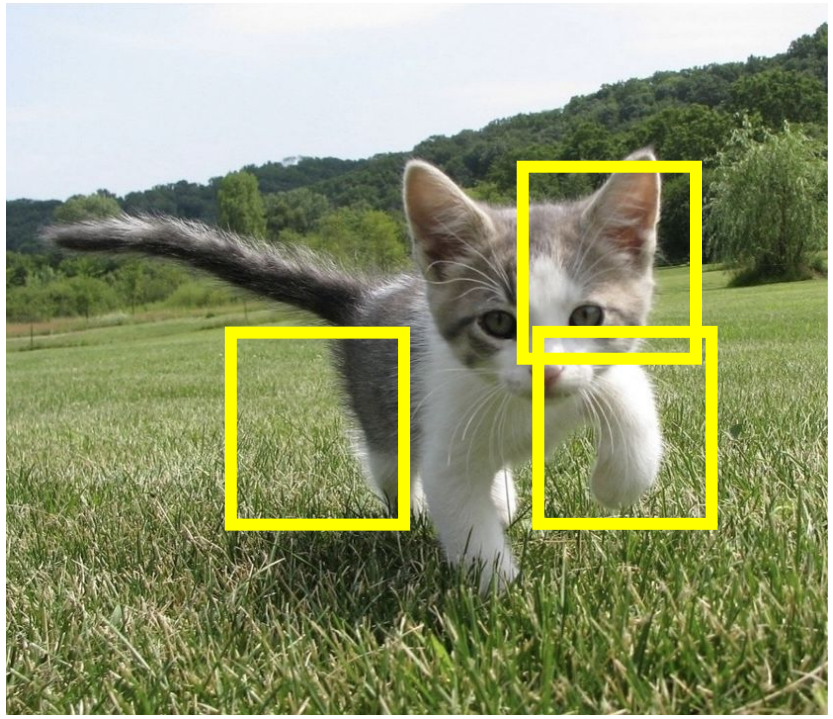
- We want a pixel-wise classification
- **In principle:** A CNN with output same size as your image, with a softmax at the end, will give you a segmentation network
- **However**, good networks require a little more modelling

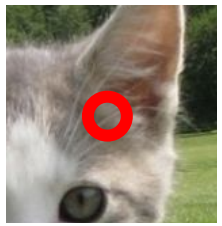
Can I use the classification net from Project 1?

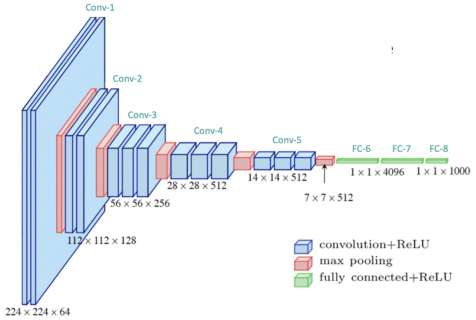


cat

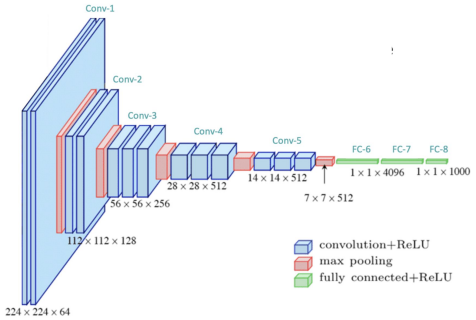
Can I use the classification net from Project 1?



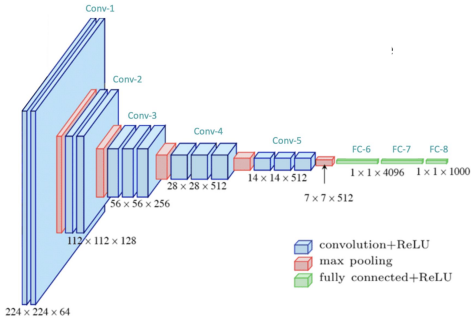




cat



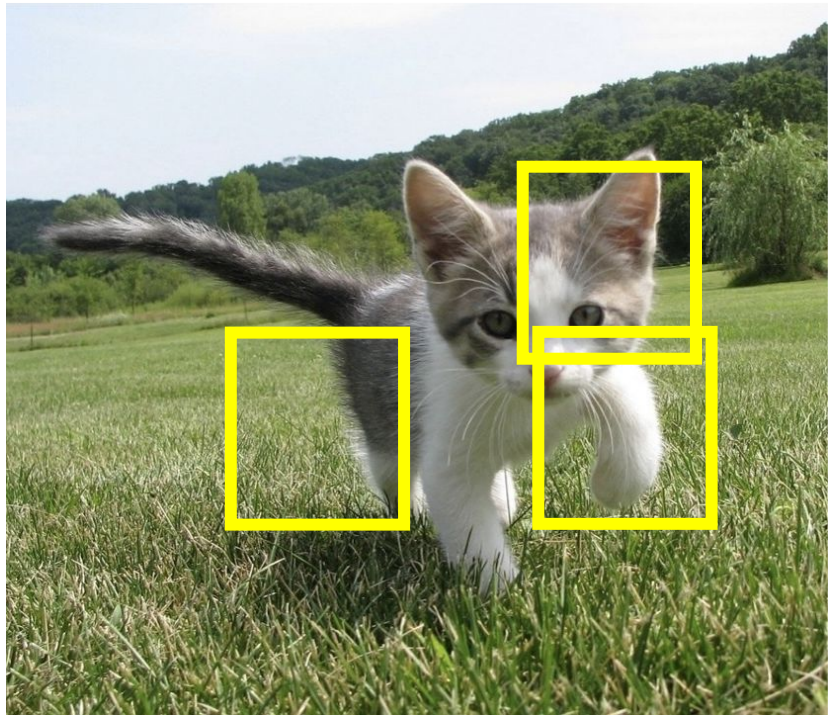
grass

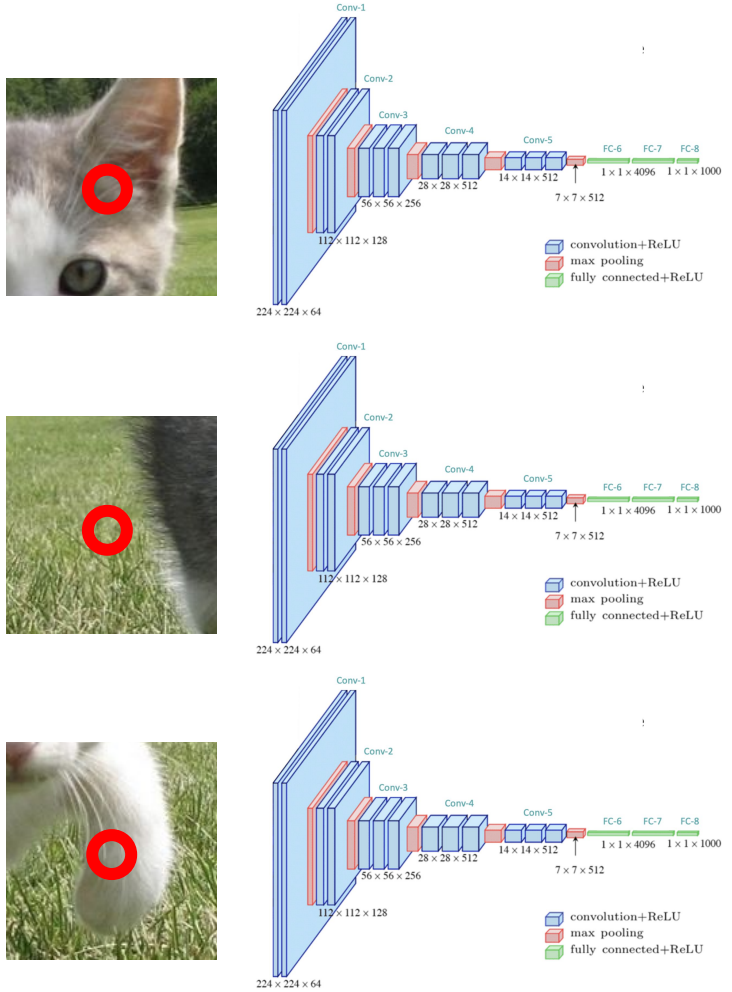


cat

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

Can I use the classification net from Project 1?





cat

grass

cat

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]

Semantic segmentation: Fully convolutional networks

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

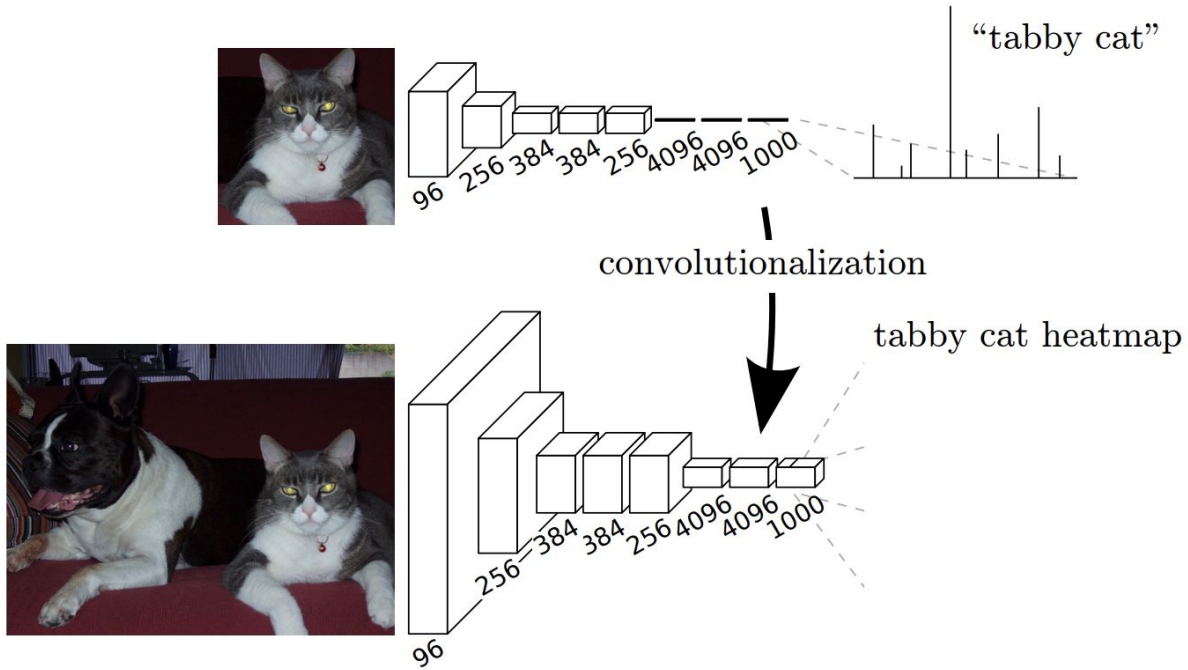


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

Loss function: Per-Pixel cross-entropy

- **Problem #1: Effective receptive field size is linear in number of conv layers: With L 3x3 conv layers, receptive field is 1+2L**
- **Problem #2: Convolution on high res images is expensive! Recall ResNet stem aggressively downsamples**

Long et al, "Fully convolutional networks for semantic segmentation", CVPR 2015

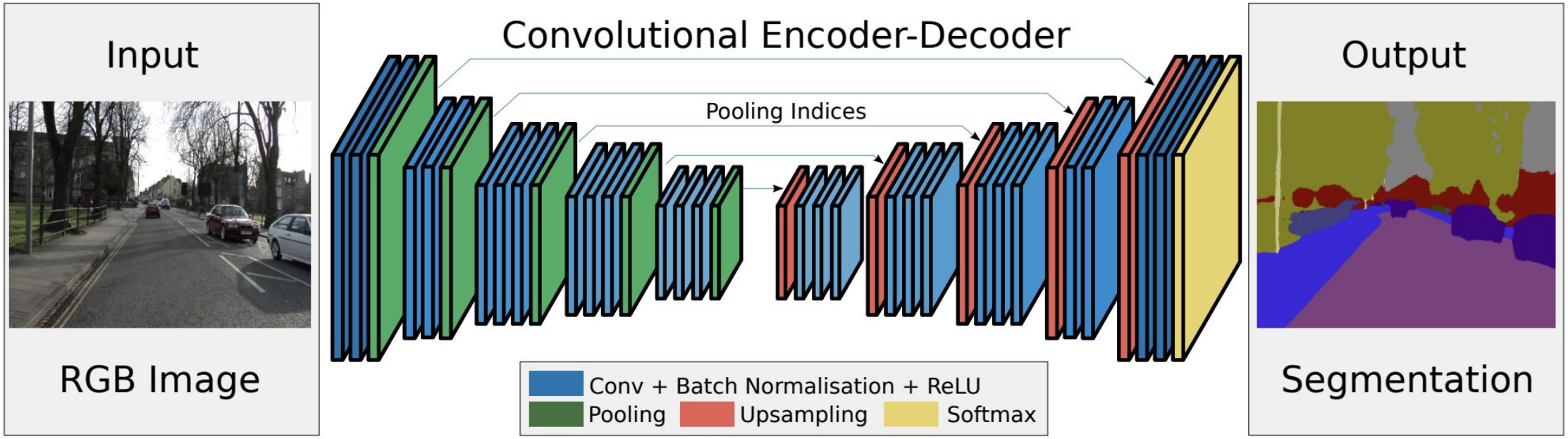
Semantic segmentation: Fully convolutional networks

Design network as a bunch of convolutional layers,
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
Badrinarayanan et al, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation” 2016

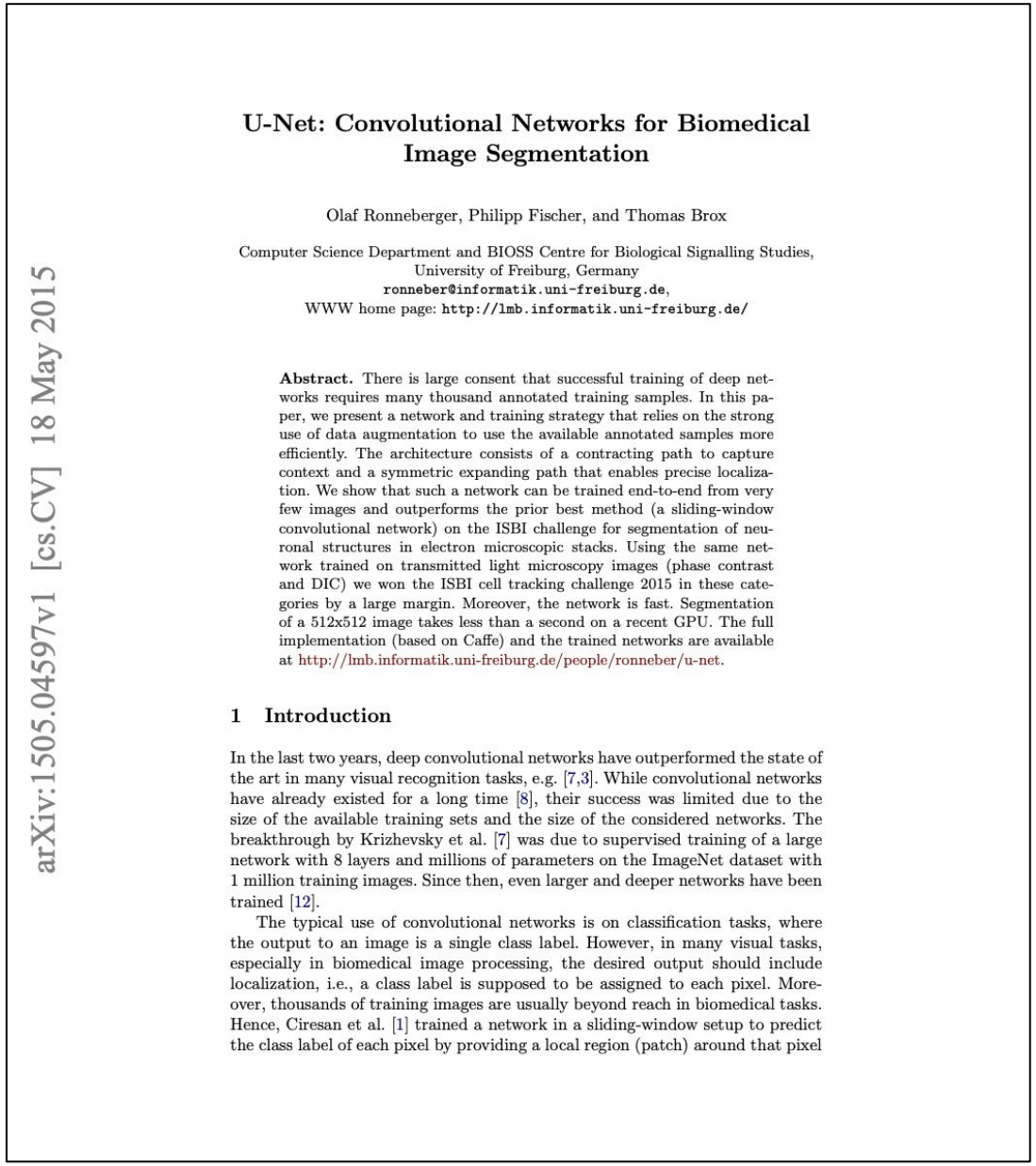
Semantic segmentation: Fully convolutional networks

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



- An encoder-decoder network
- **Downsampling**: Pooling, strided convolution
- **Upsampling**: Iinterpolation, transposed conv
- Transports pooling indices to the decoder to obtain better upsampling

The U-Net (Rönneberger et al, 2015)



U-Net: Convolutional Networks for Biomedical Image Segmentation

Olaf Ronneberger, Philipp Fischer, and Thomas Brox

Computer Science Department and BIOSS Centre for Biological Signalling Studies,
University of Freiburg, Germany
ronneber@informatik.uni-freiburg.de,
WWW home page: http://lmb.informatik.uni-freiburg.de/

Abstract. There is large consent that successful training of deep networks requires many thousand annotated training samples. In this paper, we present a network and training strategy that relies on the strong use of data augmentation to use the available annotated samples more efficiently. The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. We show that such a network can be trained end-to-end from very few images and outperforms the prior best method (a sliding-window convolutional network) on the ISBI challenge for segmentation of neuronal structures in electron microscopic stacks. Using the same network trained on transmitted light microscopy images (phase contrast and DIC) we won the ISBI cell tracking challenge 2015 in these categories by a large margin. Moreover, the network is fast. Segmentation of a 512x512 image takes less than a second on a recent GPU. The full implementation (based on Caffe) and the trained networks are available at http://lmb.informatik.uni-freiburg.de/people/ronneber/u-net.

1 Introduction

In the last two years, deep convolutional networks have outperformed the state of the art in many visual recognition tasks, e.g. [7,3]. While convolutional networks have already existed for a long time [8], their success was limited due to the size of the available training sets and the size of the considered networks. The breakthrough by Krizhevsky et al. [7] was due to supervised training of a large network with 8 layers and millions of parameters on the ImageNet dataset with 1 million training images. Since then, even larger and deeper networks have been trained [12].

The typical use of convolutional networks is on classification tasks, where the output to an image is a single class label. However, in many visual tasks, especially in biomedical image processing, the desired output should include localization, i.e., a class label is supposed to be assigned to each pixel. Moreover, thousands of training images are usually beyond reach in biomedical tasks. Hence, Ciresan et al. [1] trained a network in a sliding-window setup to predict the class label of each pixel by providing a local region (patch) around that pixel

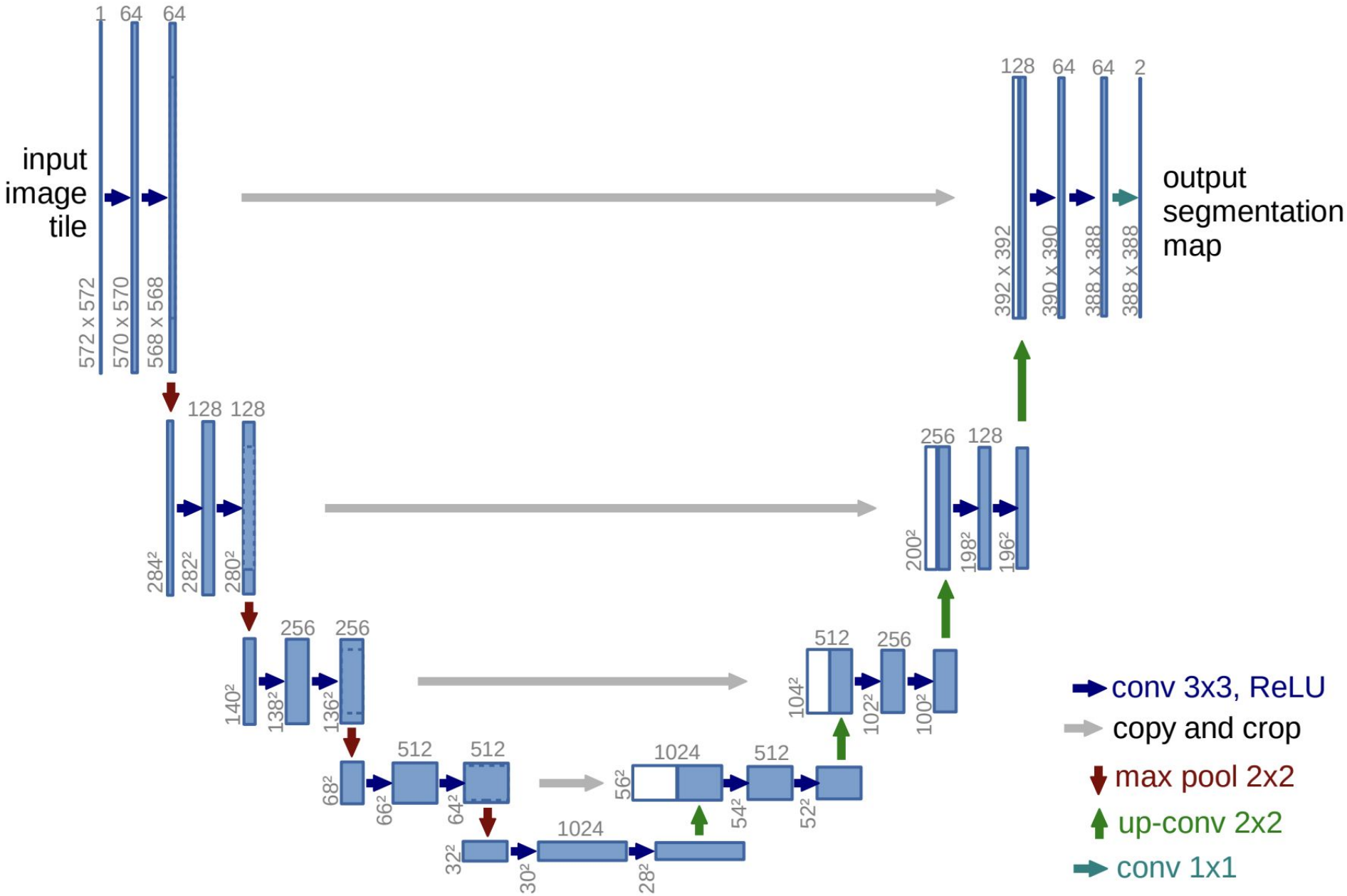
U-net: Convolutional networks for biomedical image segmentation
O Ronneberger, P Fischer, T Brox - … **image** computing and computer …, 2015 - Springer
… We demonstrate the application of the **u-net** to three different **segmentation** tasks. The first task is the **segmentation** of neuronal structures in electron microscopic recordings. An …
☆ Save Cite Cited by 93675 Related articles All 32 versions

As of Oct 2025:

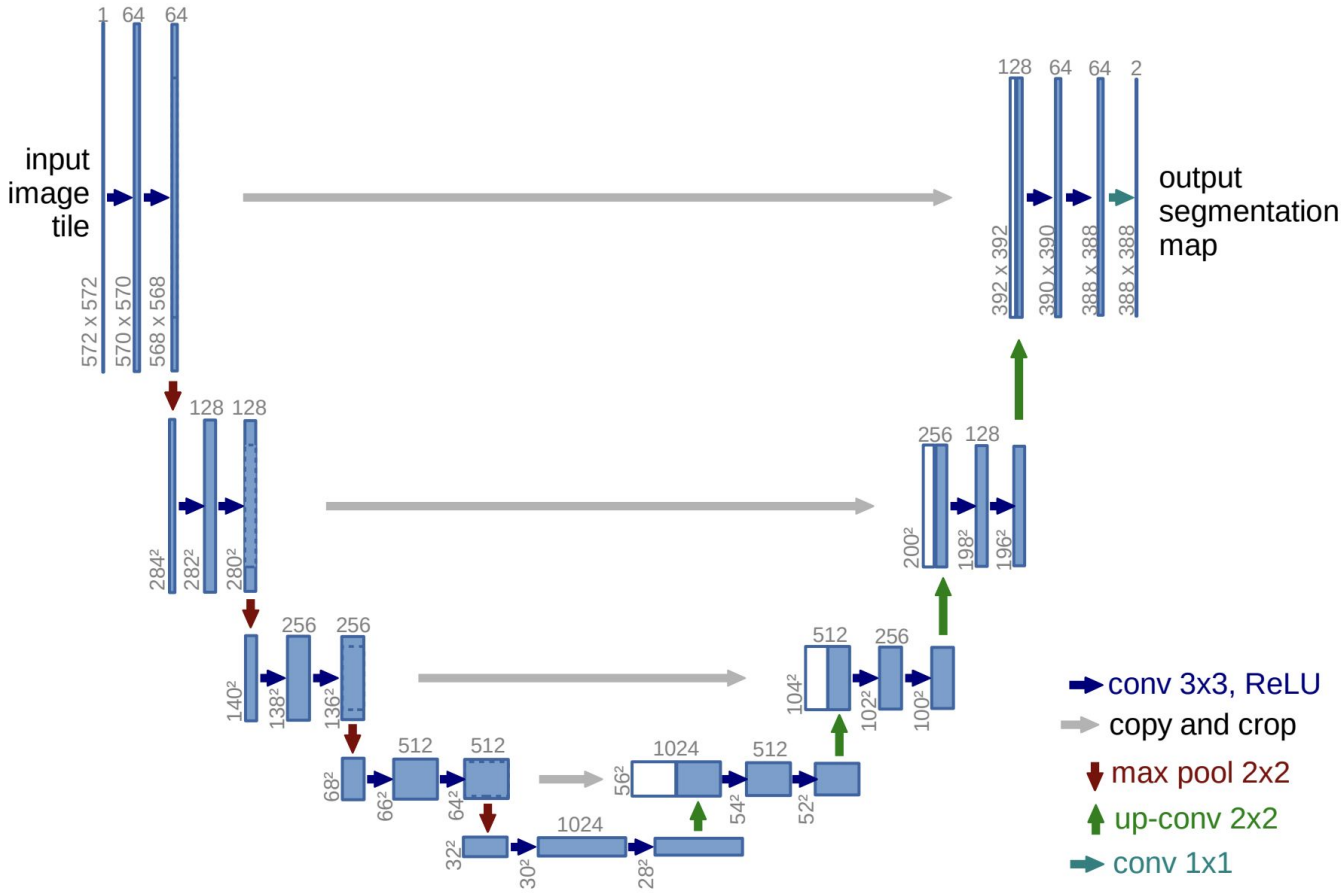
U-net: Convolutional networks for biomedical image segmentation
O Ronneberger, P Fischer, T Brox - International Conference on Medical …, 2015 - Springer
… We demonstrate the application of the **u-net** to three different segmentation tasks. The first
… The **u-net** (averaged over 7 rotated versions of the input data) achieves without any further …
☆ Save Cite Cited by 121580 Related articles All 28 versions

The U-net and variants thereof still remain state-of-the-art for biomedical image segmentation

The U-Net


input image tile
572 x 572
570 x 570
568 x 568
1 64 64
128 128
284²
282²
280²
256 256
140²
138²
136²
512 512
68²
66²
64²
1024
32²
30²
28²
1024 512
56²
54²
52²
512 256
104²
102²
100²
256 128
200²
198²
196²
128 64 64 2
392 x 392
390 x 390
388 x 388
388 x 388
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

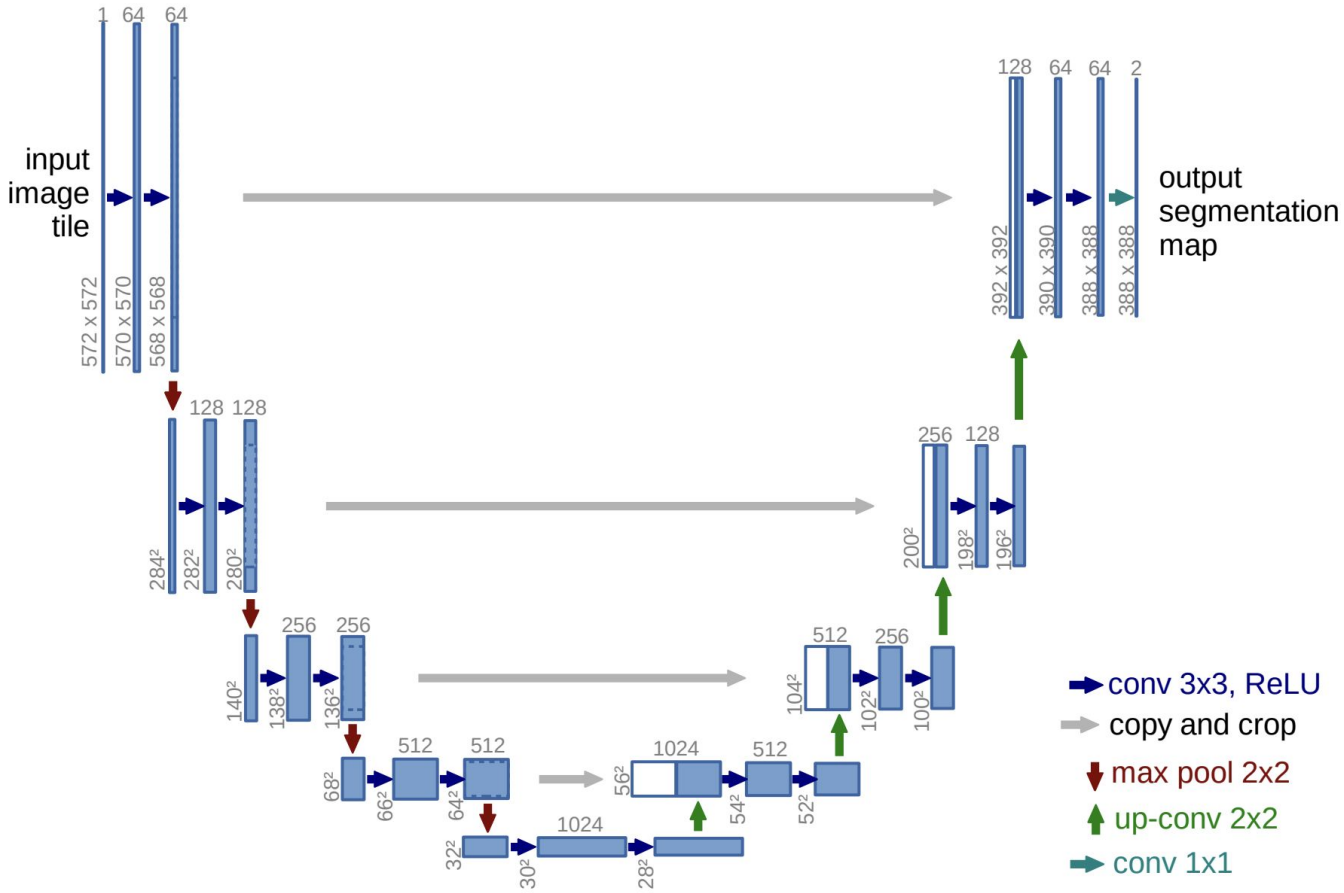
The U-Net



Discuss with your neighbor:

- Input and output have different sizes. Why?
- How do you map an output segmentation pixel to its corresponding input pixel?

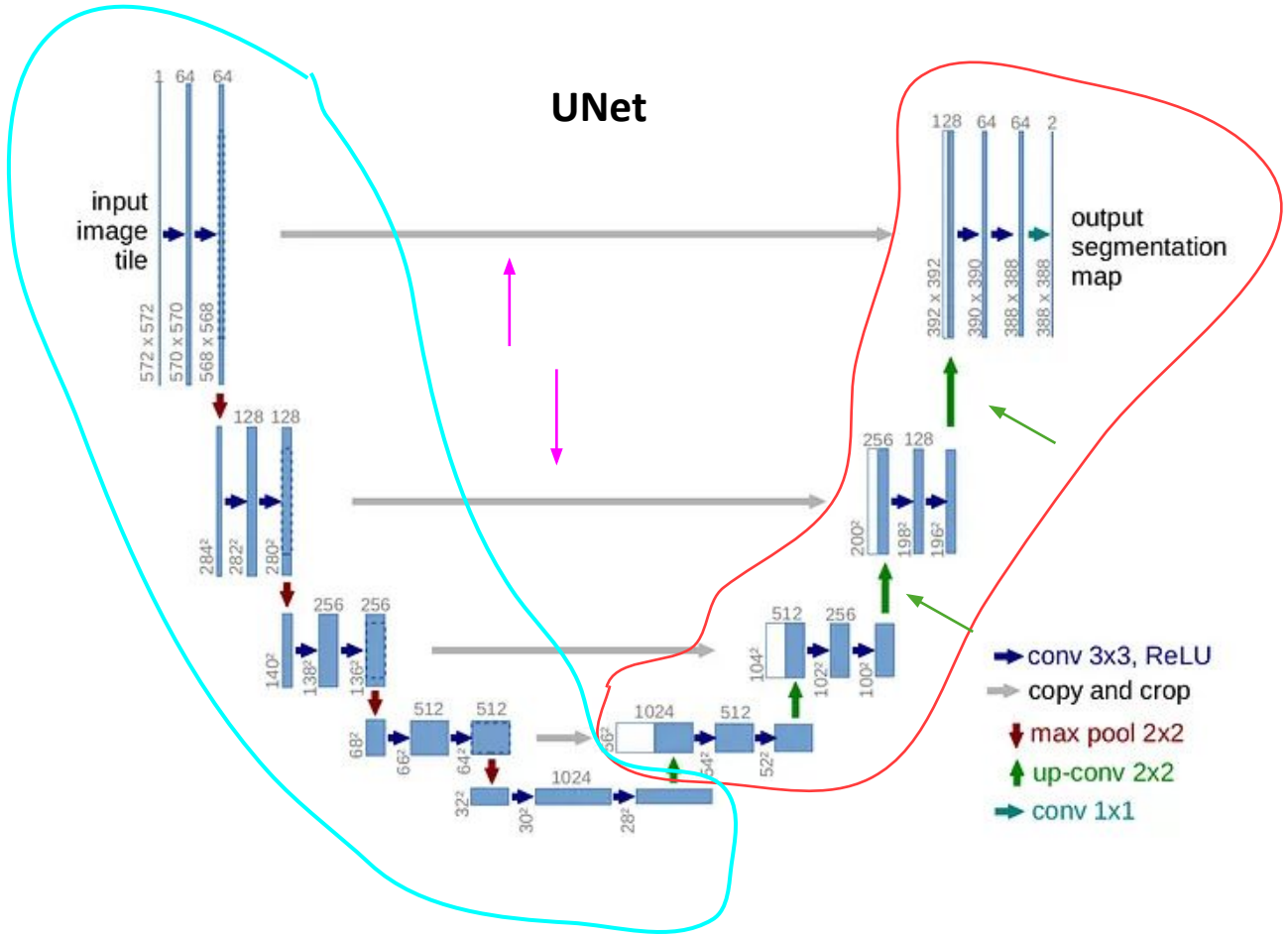
The U-Net



Discuss with your neighbor:

- The architecture – why does it look the way that it looks?
- What is the role of the max pooling and up-convolutions?
- What is the role of the skip connections?

The U-Net



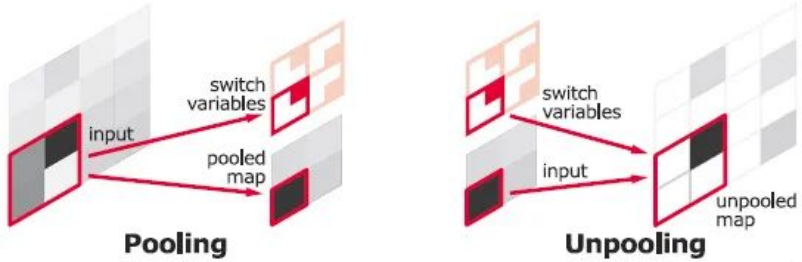
- Encoder
- Decoder
- Skip connections
- “Upsampling”

The U-Net (Upsampling)

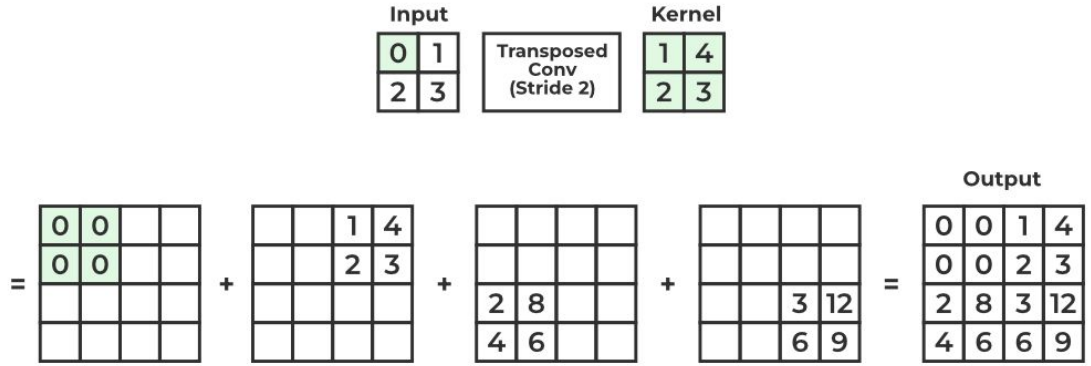
Going from feature maps of size, e.g., 128x128 to 256x256.

Ways to do it:

- **Interpolation** of any kind, e.g., bilinear, nearest neighbor. Think of when you are resizing an image.
- **Unpooling** (from “Learning Deconvolution Network for Semantic Segmentation” Noh et al. ICCV (2015)



- **Transposed convolutions**



Questions???

A few hints and observations

- The implementation in the Exercise of the U-net does not follow the figures precisely: Differences in depth, nr of convolutions, and padding. For your own simplicity, you may want to use a padding that matches your convolutional kernels to ensure that the output image has the same size as the input.

- Loss functions: Implementing your own is great for learning, but you may run into numerical issues. Try to figure out what causes them – but otherwise, a safe rescue is usually to use a built-in

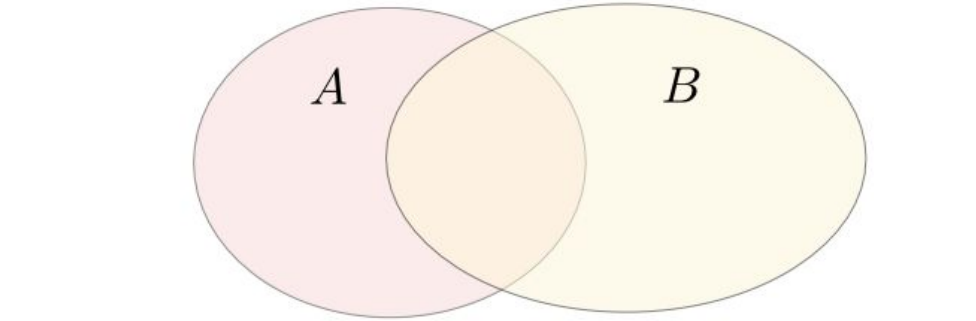
Evaluation: How well did your segmentation do?

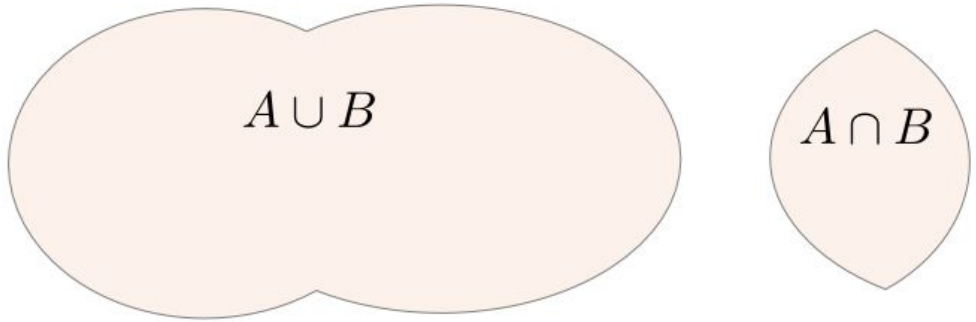
Accuracy:

$$\frac{\sharp \text{ correctly classified pixels}}{\sharp \text{ pixels in total}}$$

Evaluation: How well did your segmentation do?

Intersection over Union (IoU)/Jaccard

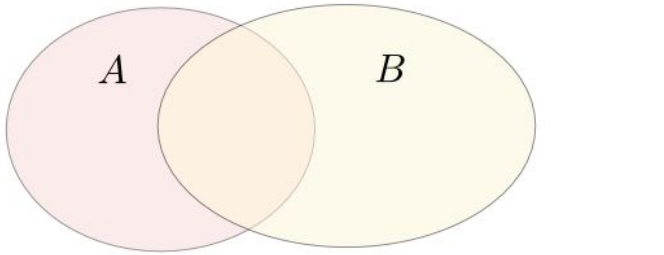


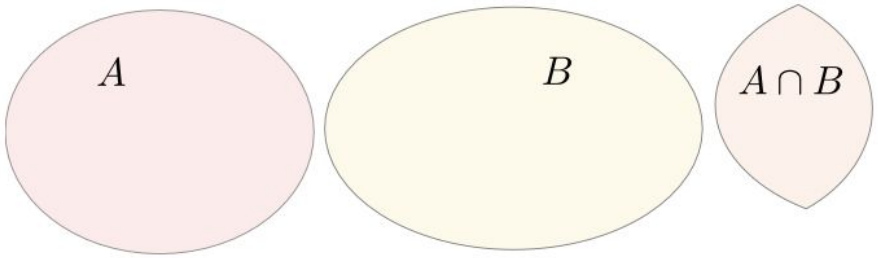


$$IoU = \frac{A \cap B}{A \cup B}$$

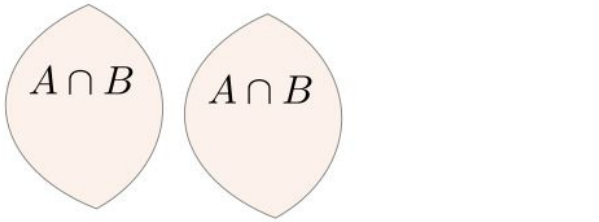
Evaluation: How well did your segmentation do?

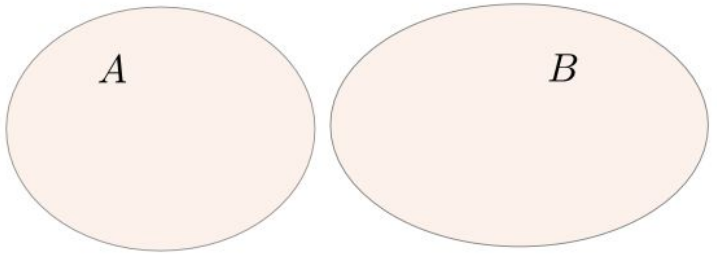
Dice





$$Dice = \frac{2|A \cap B|}{|A| + |B|} =$$

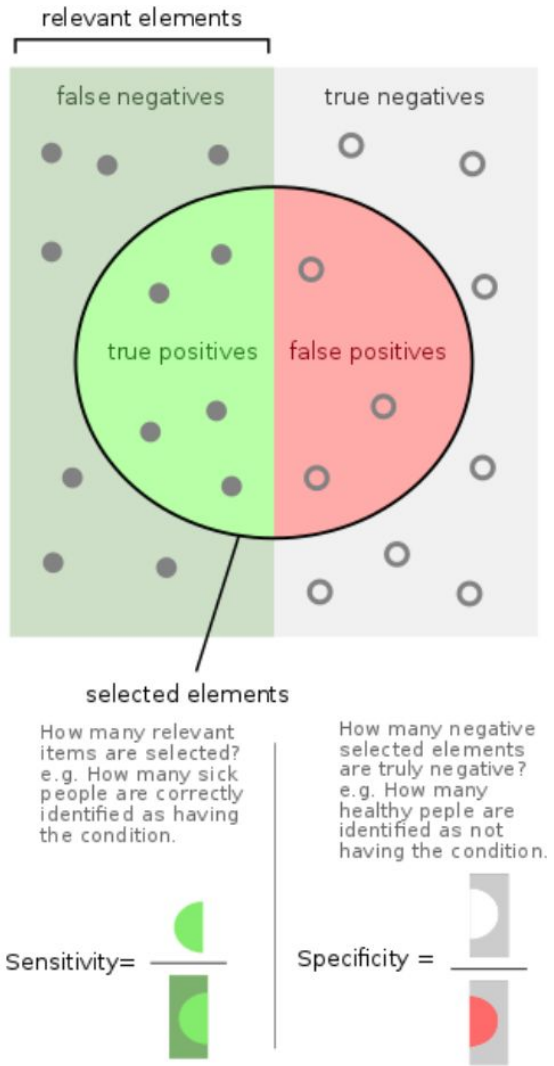




In today's exercise, you will use a version of Dice as a loss function.

Evaluation: How well did your segmentation do?

Sensitivity/Specificity



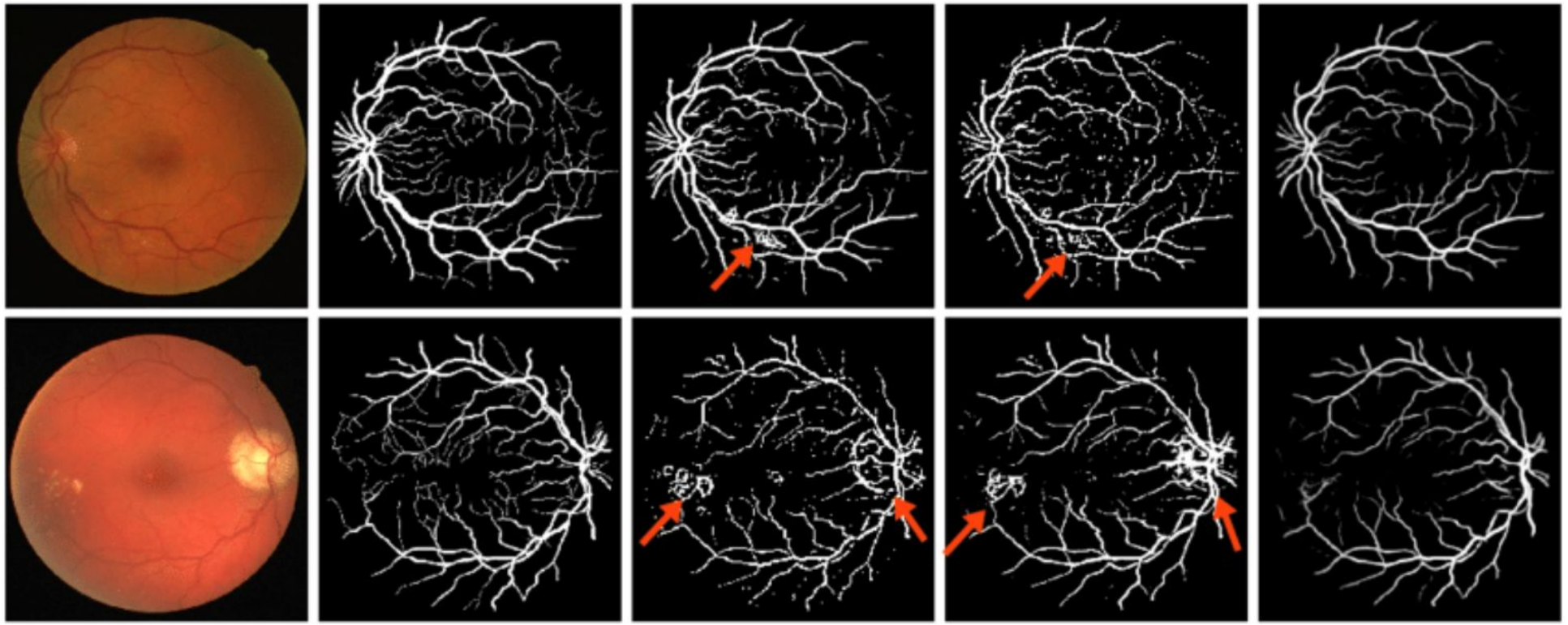
[Figure from https://commons.wikimedia.org/wiki/File:Sensitivity_and_specificity.svg]

Evaluation: How well did your segmentation do?

Can you imagine a situation where these are poor measures?

Evaluation: How well did your segmentation do?

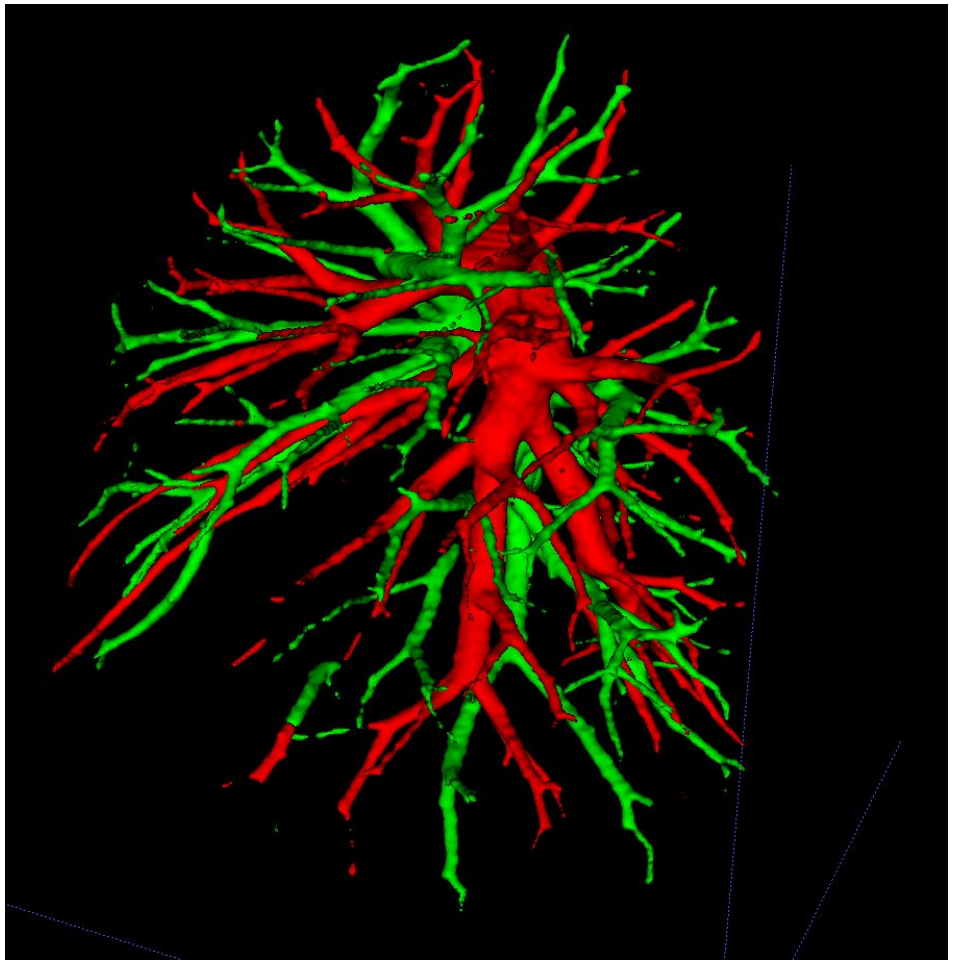
Can you imagine a situation where these are poor measures?



(A) Fundus image (B) Ground truth (C) Nguyen *et al.* 2013 (D) Orlando *et al.* 2014 (E) Our DeepVessel

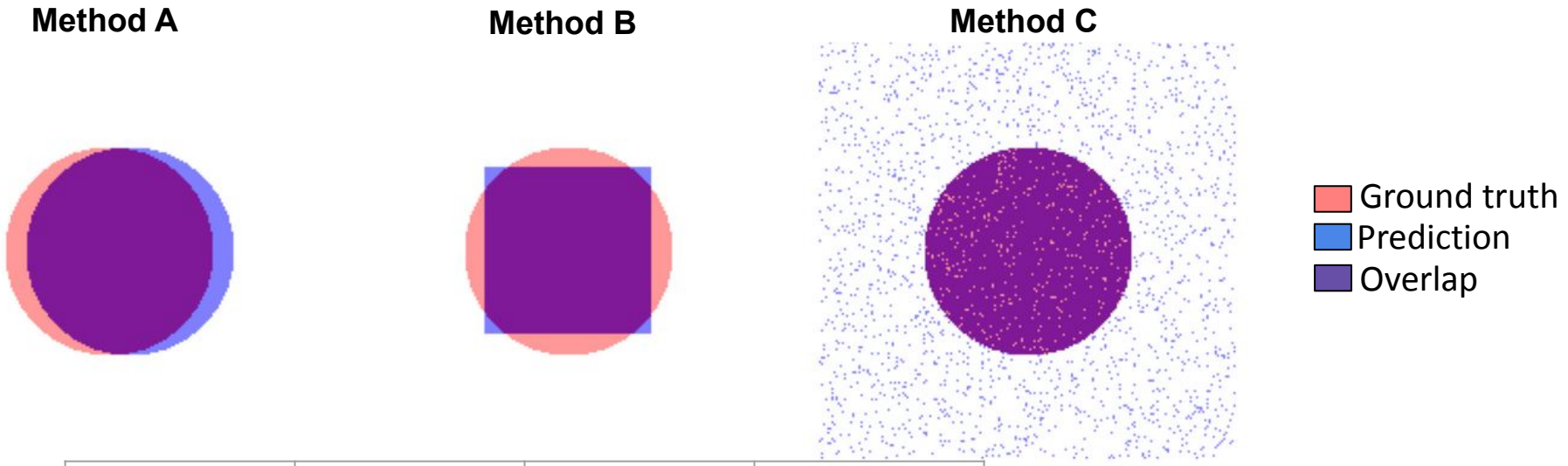
Evaluation: How well did your segmentation do?

Can you imagine a situation where these are poor measures?



119 connected components!

Topology, Shape and Segmentation Quality



Metric ↓	Method A	Method B	Method C
Dice coefficient	0.8734	0.8738	0.8767
Accuracy	0.9505	0.9551	0.9476
Jaccard index	0.7753	0.7759	0.7805
Connected components	1	1	1347
Holes	0	0	328

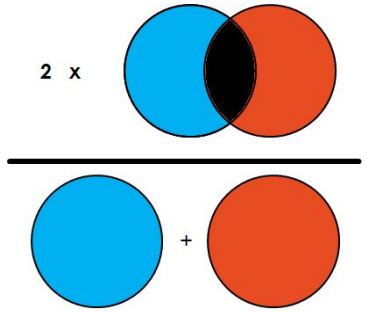
Evaluation: How well did your segmentation do?

Pixel/voxel-level metrics

Dice coefficient

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

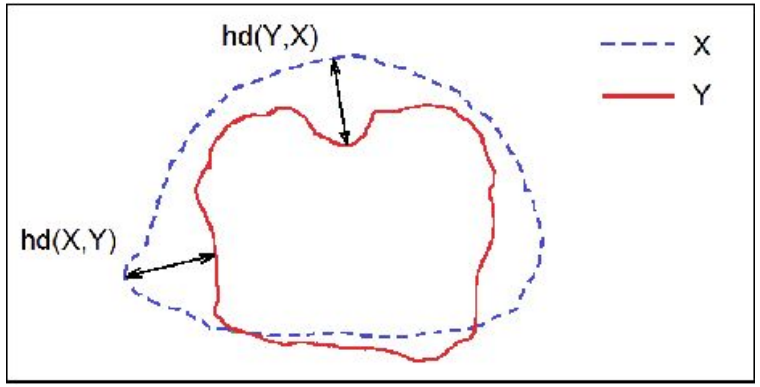
$$DSC = \frac{2TP}{2TP + FP + FN}$$



Accuracy, Recall, Precision

Distance-based metrics

Hausdorff distance / (HD95)



Other type of metrics (depending on the problem)

Betti numbers (Topology)

Compactness

A couple of useful techniques

Dilated convolution

- Alternative to the alternating convolutional / pooling layers
- Exponentially dilate the convolutional kernel
- Exponentially enlargened receptive field as you move through the layers

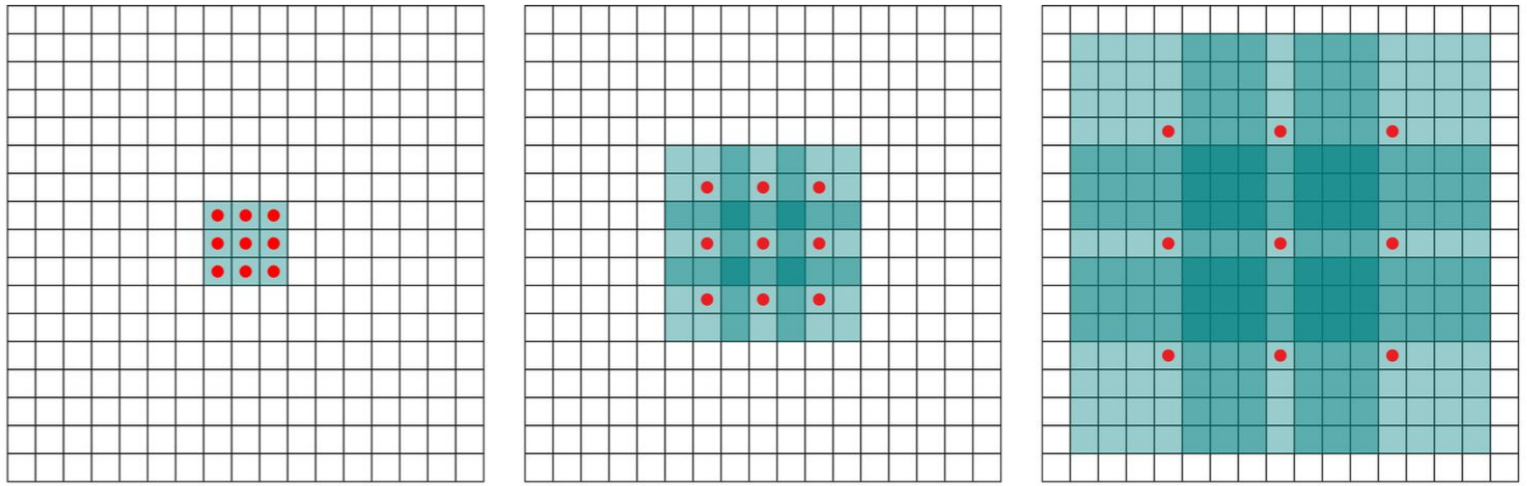
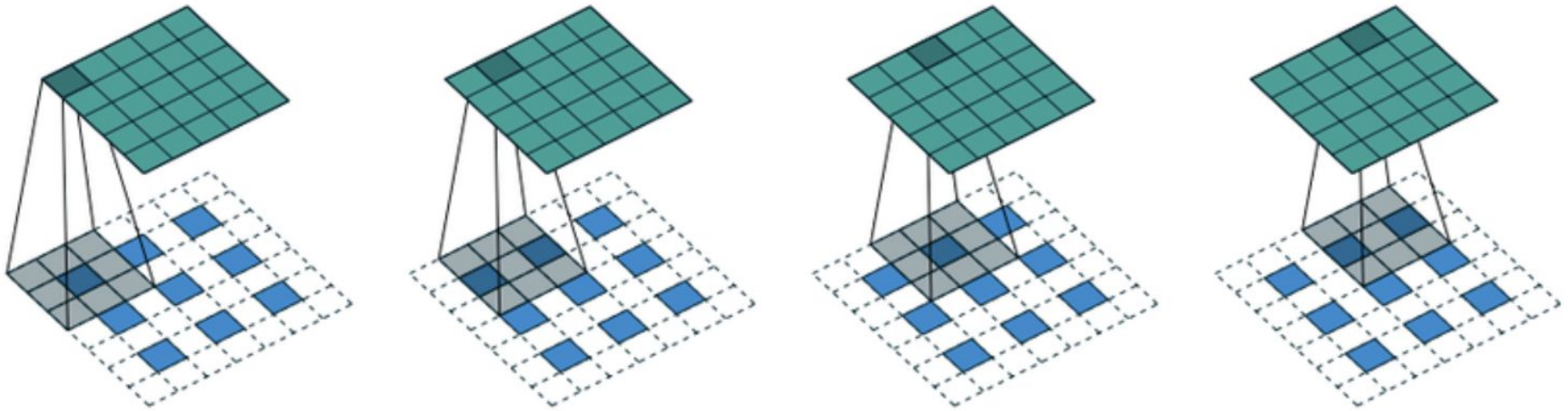


Figure 1: Systematic dilation supports exponential expansion of the receptive field without loss of resolution or coverage. (a) F_1 is produced from F_0 by a 1-dilated convolution; each element in F_1 has a receptive field of 3×3. (b) F_2 is produced from F_1 by a 2-dilated convolution; each element in F_2 has a receptive field of 7×7. (c) F_3 is produced from F_2 by a 4-dilated convolution; each element in F_3 has a receptive field of 15×15. The number of parameters associated with each layer is identical. The receptive field grows exponentially while the number of parameters grows linearly.

Transpose convolutions

An alternative upsampling strategy: Dilate the image prior to convolution.



Can lead to line-like artefacts due to uneven coverage of the image content.

Questions???

Comments on the Exercise