DLinCV Lecture 2.1

Image Segmentation
07.06.2023

Aasa Feragen
afhar@dtu.dk

Schedule for part 3: Segmentation

Part 2: Segmentation

Thursday 8.6	09:00-10:00 Lecture - Introduction to Segmentation 10:00-12:00 Exercises - Exercise 3 on Segmentation 13:00-17:00 Exercise 3 on Segmentation	Aasa 13 - 15 Thanos 14 - 16 Paraskevas 15 - 17 Manxi
Friday 9.6	09:00-10:00 Lecture: - Foundation models for segmentation *10:00-11:00 Project 1 poster session* - Give peer feedback to designated groups 11:00-17:00 Work on project	Aasa 10 - 12 Aasa 13 - 15 Manxi 14 - 16 Paraskevas 15 - 17 Thanos
Monday 12.6	09:00-9:30 Q&A, quick tips 09:30-17:00 Work on project **Project 2 deadline at midnight**	9 - 12 Aasa 13 - 15 Manxi 14 - 16 Paraskevas 15 - 17 Thanos

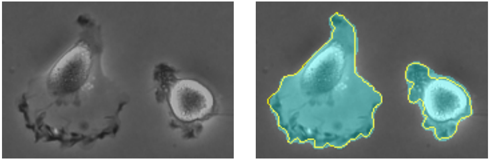
This lecture's learning goals

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

What is image segmentation?

What is image segmentation?

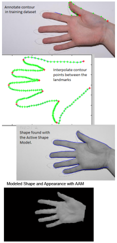


Rönneberger et a, 2015

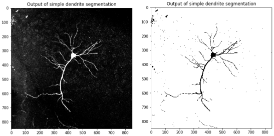
- Overall task: Partitioning an image into its objects and background – segmenting volumes and boundaries
- Difference from detection: Not just a bounding box

What is image segmentation?

"Old school" algorithms

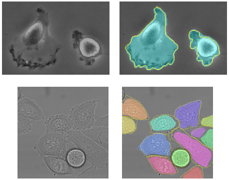


Active shape model



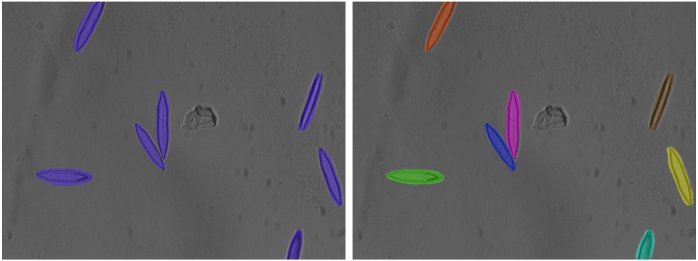
Segmentation by threshold

Voxel classification (today: deep learning)



What is image segmentation?

Semantic segmentation versus instance segmentation



(a) Semantic segmentation mask. (b) Instance segmentation mask.

We will focus on semantic segmentation.

What is image segmentation useful for?

Application: Preprocessing

Brain tissue segmentation: Constraining analysis to relevant regions

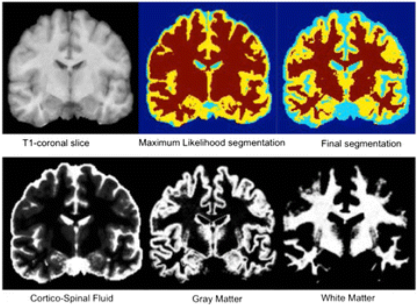
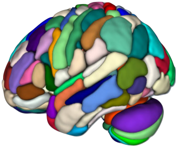


Figure: Figure from Villalon-Reina et al, 2016

Application: Preprocessing

Brain region segmentation: Building a connectivity network



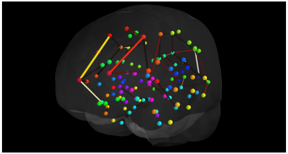
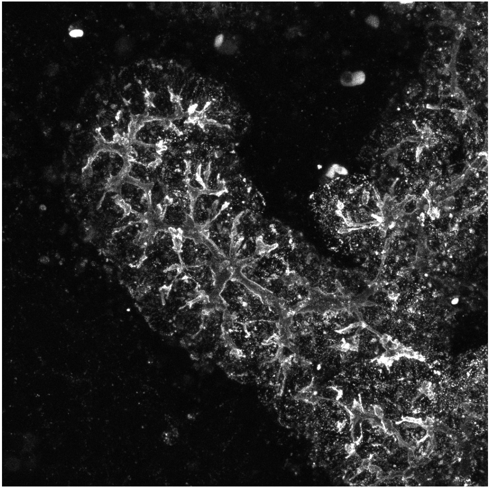


Figure: Left figure from Fan et al, 2016

Application: Extracting objects for scientific discovery



Predicting disease or focusing treatment

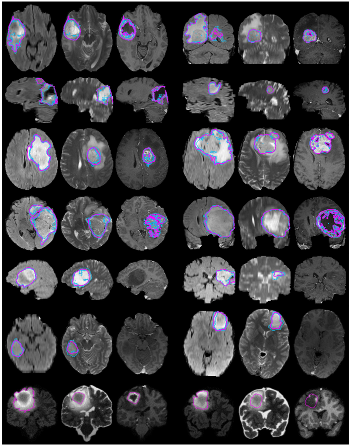
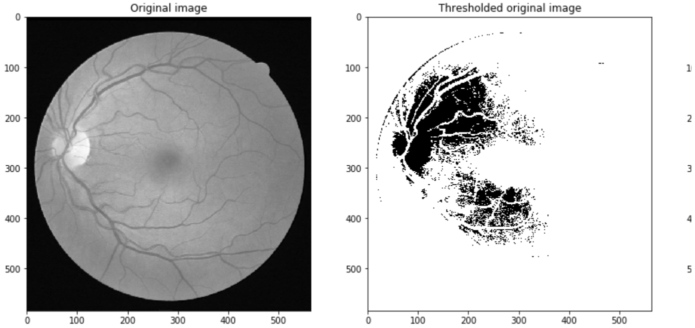


Figure: From the BRATS challenge dataset.

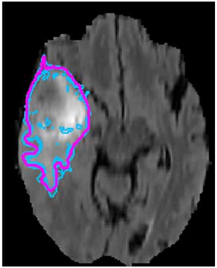
Challenges in image segmentation

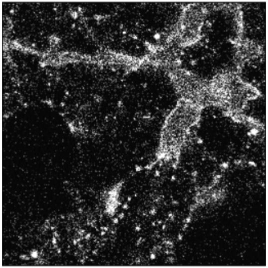
Background inhomogeneity



Challenges in image segmentation

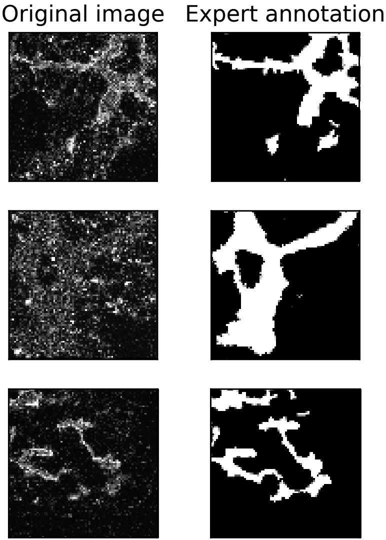
Fuzzy boundaries



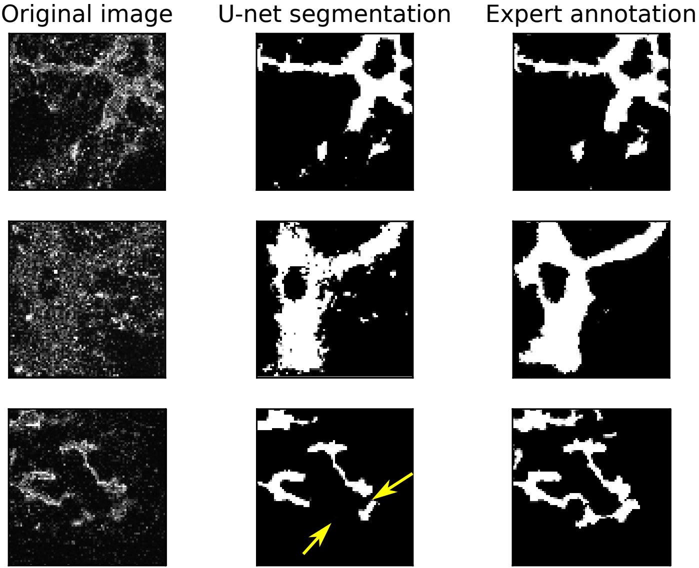


Challenges in image segmentation

Noisy labels, few labels



Challenges in image segmentation



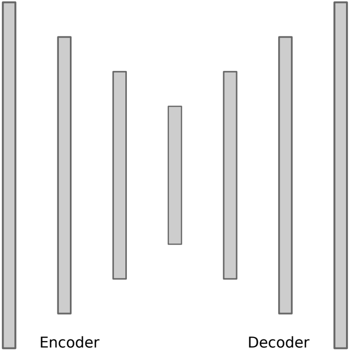
Small pixel error can be large object error

Questions?

CNNs for segmentation

It's "all" in the final layer

- You 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
- Good networks require a little more modelling



Example: SegNet

- An encoder-decoder network
- Transports pooling indices to the decoder to obtain better upsampling

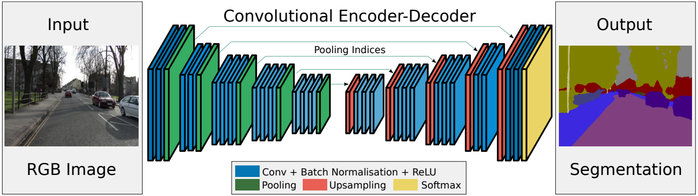
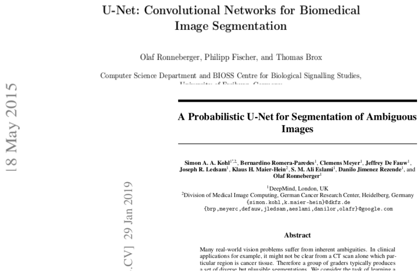


Figure: SegNet, Badrinarayanan et al, 2016

The U-net (Rönneberger et al, 2015)

- The U-net (https://arxiv.org/abs/1505.04597) and variants thereof still remain state-of-the-art for biomedical image segmentation


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,

18 May 2015

A Probabilistic U-Net for Segmentation of Ambiguous Images

Simon A. A. Kohl[1,*,2], Bernardino Romera-Paredes[1], Clemens Meyer[1], Jeffrey De Fauw[1], Joseph R. Ledsam[1], Klaus H. Maier-Hein[2], S. M. Ali Eslami[1], Danilo Jimenez Rezende[1], and Olaf Ronneberger[1]

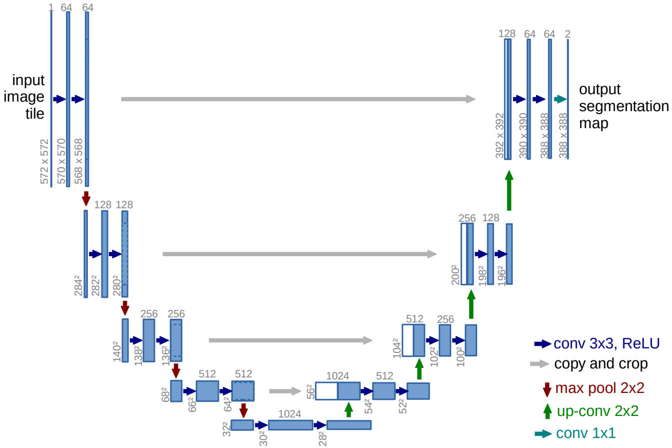
[1]DeepMind, London, UK
[2]Division of Medical Image Computing, German Cancer Research Center, Heidelberg, Germany
{simon.kohl,k.maier-hein}@dkfz.de
{brp,meyerc,defauw,jledsam,aeslami,danilor,olafr}@google.com

.CV] 29 Jan 2019

Abstract

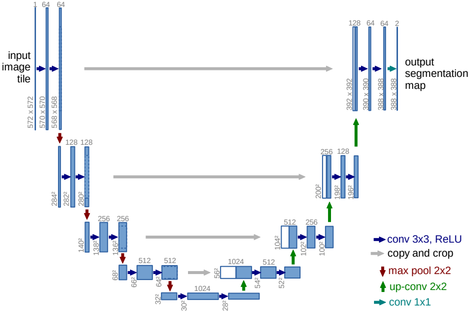
Many real-world vision problems suffer from inherent ambiguities. In clinical applications for example, it might not be clear from a CT scan alone which particular region is cancer tissue. Therefore a group of graders typically produces

The U-net



1
64
64
input image tile
572 x 572
570 x 570
568 x 568
128
128
284²
282²
280²
256
256
140²
138²
136²
512
512
68²
66²
64²
1024
32²
30²
28²
1024
56²
512
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

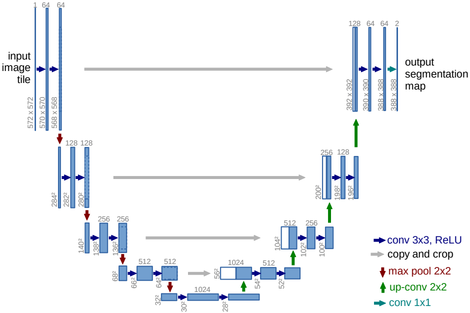
The U-net



Discuss with your neighbor:

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

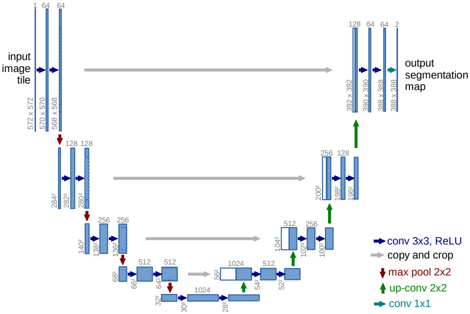
The U-net



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?

Questions?


input
image
tile
output
segmentation
map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

A few hints and observations

- The implementation in Exercise 2 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

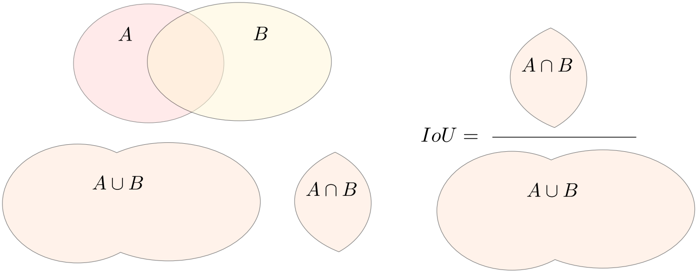
How well did your segmentation do? Validation metrics

Accuracy:

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

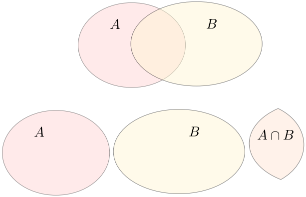
How well did your segmentation do? Validation metrics

Intersection over Union (IoU)/Jaccard

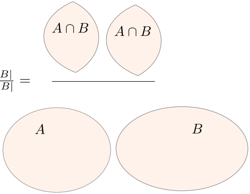


How well did your segmentation do? Validation metrics

Dice



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



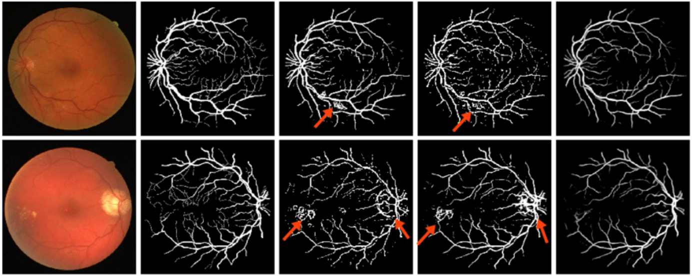
In today's exercise, you will use a version of Dice as a loss function.

How well did your segmentation do? Validation metrics

Can you imagine a situation where these are poor measures?

How well did your segmentation do? Validation metrics

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

How well did your segmentation do? Validation metrics

Sensitivity/Specificity

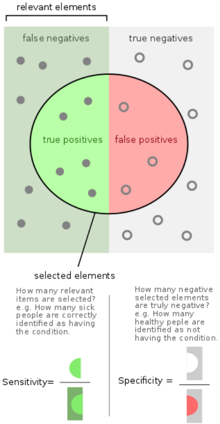


Figure from https://commons.wikimedia.org/wiki/File:Sensitivity_and_specificity.svg

A couple of useful techniques

Dilated convolutions[1]

Alternative to the alternating convolutional / pooling layers:
Exponentially dilate the convolutional kernel:

[1] https://arxiv.org/pdf/1511.07122.pdf

Dilated convolutions

Exponentially enlarged receptive field as you move through the layers:

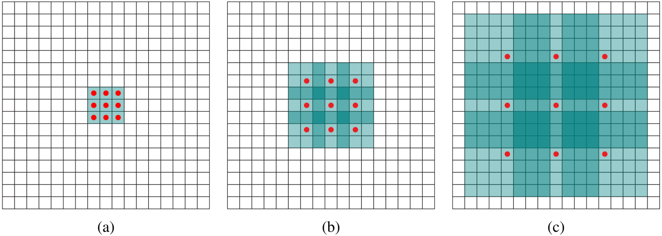
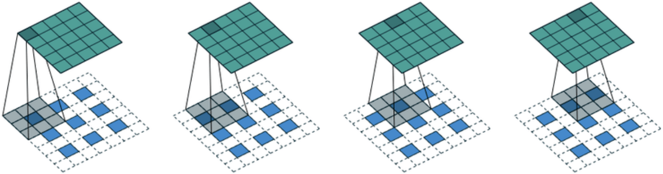


Figure 1: Systematic dilation supports exponential expansion of the receptive field without loss of resolution or coverage. (a) F_1 is produced from F_0 by a 1-dilated convolution; each element in F_1 has a receptive field of 3×3. (b) F_2 is produced from F_1 by a 2-dilated convolution; each element 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

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



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

Now what?

Today, you will work with Exercise 2, which you find on Learn.

Summary

By now 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