

Image Segmentation

*Lecture about Segmentation evaluation methods as well as base-line theory surrounding segmentation.
This week's paper is [1]*

Author: Anton Zhitomirsky

Contents

1 Semantic Segmentation	4
1.1 Uses	4
2 Challenges	4
2.1 noise	4
2.2 partial volume effects	4
2.3 Intensity Inhomogeneities	5
2.4 Anisotropic Resolution	5
2.5 Imaging Artifacts	6
2.6 limited contrasts	6
2.7 Morphological Variability	6
3 Segmentation Evaluation	6
3.1 Ground truth	6
3.2 Gold standard	7
3.3 Evaluation Metrics	7
3.3.1 Precision	7
3.3.2 Accuracy	7
3.3.3 Robustness	7
3.3.4 Confusion matrix	8
3.3.5 Accuracy	8
3.3.6 Precision — positive predictive value	8
3.3.7 Recall — sensitivity — hit rate — true positive rate	8
3.3.8 Specificity — True negative rate	8
3.3.9 F_β score	8
3.3.10 F1 score	9
3.3.11 'IoU' - Overlap based - Jaccard Index	9
3.3.12 'DSC' - Overlap based - Dice Similarity	9

3.3.13 Volume similarity	9
3.3.14 ‘HD’ - Surface Hasudorff distance	9
3.3.15 ‘ASSD’ - Surface (symmetric) average surface distance	10
3.4 Pitfalls in segmentation evalauation	10
3.4.1 Effect of structure size	10
3.4.2 Effect of structure shape	11
3.4.3 Effect of spatial alignment	13
3.4.4 Effect of holes	13
3.4.5 Effect of Annotation noise	14
3.4.6 Effect of empty labelmaps	14
3.4.7 Effect of resolution	15
3.5 Preference for oversegmentatin to undersegmentation	15
4 Segmentation Methods	16
4.1 Intensity-based segmentation — thresholding	16
4.1.1 Advantages	16
4.1.2 Disadvantages	16
4.2 Region-based — region growing	16
4.2.1 Advantages	16
4.2.2 Disadvantages	17
4.3 Atlas-based segmentation	17
4.4 Segmentation using Registration	17
4.5 Multi-Atlas Label Propagation	18
4.5.1 Advantages	18
4.5.2 Disadvantages	18
4.6 Learning-based segmentation — random forests, convolutional neural networks	18
4.6.1 Random forests	18
4.6.2 Advantages	19
4.6.3 Disadvantages	19
5 Segmentation via Dense Classification	20
5.1 LeNet	20
5.1.1 Fully convolutional LeNet	21
6 Encoder-Decoder Networks	22
6.1 U-Net	23
6.1.1 Upsampling	23
6.1.2 Convolutions	23
6.1.3 Transpose convolutions	24
6.1.4 Dilated convolutions	24
6.1.5 Atrous spatial pyramid pooling	25
6.1.6 Padding effects	25
6.1.7 Multi-scale processing	26
6.2 Vision trasnformers	27

Bibliography

29

1 Semantic Segmentation

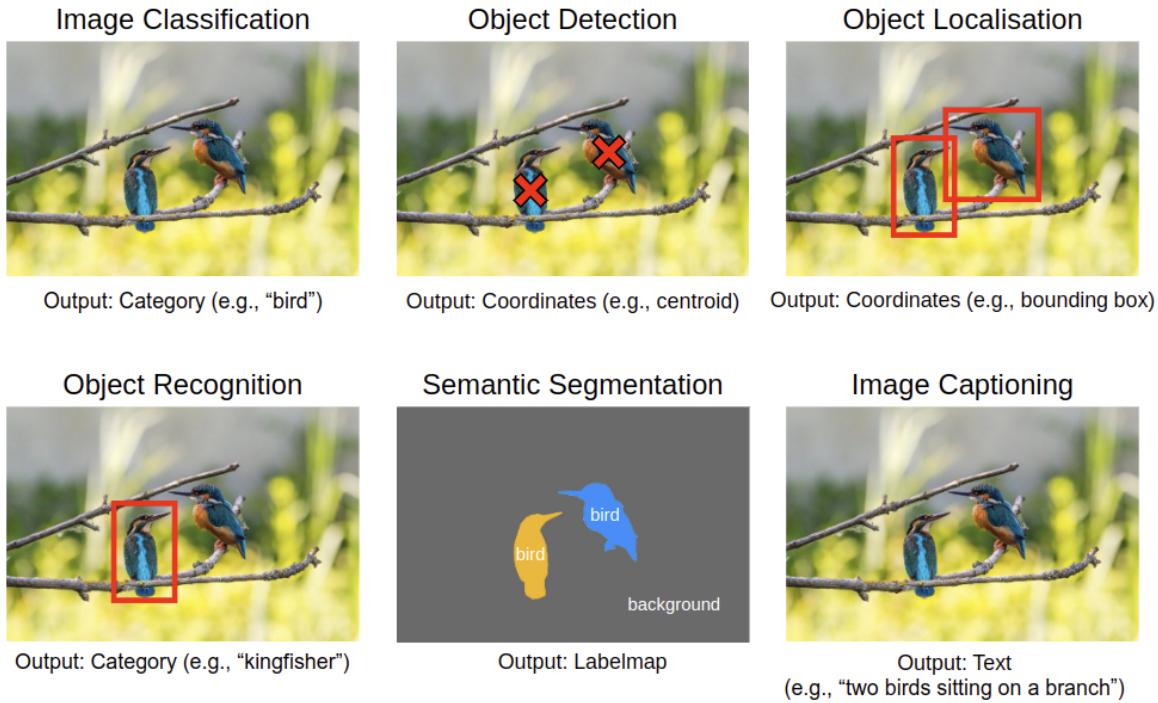


Figure 1: Common image analysis tasks

Definition 1.1 (Semantic Segmentation). Given an input image, we want to label individual pixels in the image according to which object or class they belong to. It is a dense classification where every pixel is being assigned to a specific class.

Each segmented region is assigned a semantic meaning (which contrasts the segmentation based on ‘pure’ clustering of the image into coherent regions).

1.1 Uses

- conducting quantitative analysis, e.g. measuring the volume of a ventricular cavity
- determining the precise location and extent of an organ or certain type of tissue, e.g. a tumour, for treatment such as radiation therapy
- creating 3D models used for simulation, e.g. generating a model of an abdominal aortic aneurysm for simulating stress/strain distributions

2 Challenges

2.1 noise

Noise in images refers to high-frequency pixel variability which is not relevant, or may obscure, the model’s task.

2.2 partial volume effects

The image produced by software is a quantized version of the object. Due to the coarse sampling, the resulting image shows partial volume effects at the boundary of the image. These pixels are not

aligned with real world boundaries. Therefore, the pixels contain a mixture of two different objects. An object may also be elevated, and it is therefore difficult to measure where the extent of the object is because it is unclear where the object starts or ends.

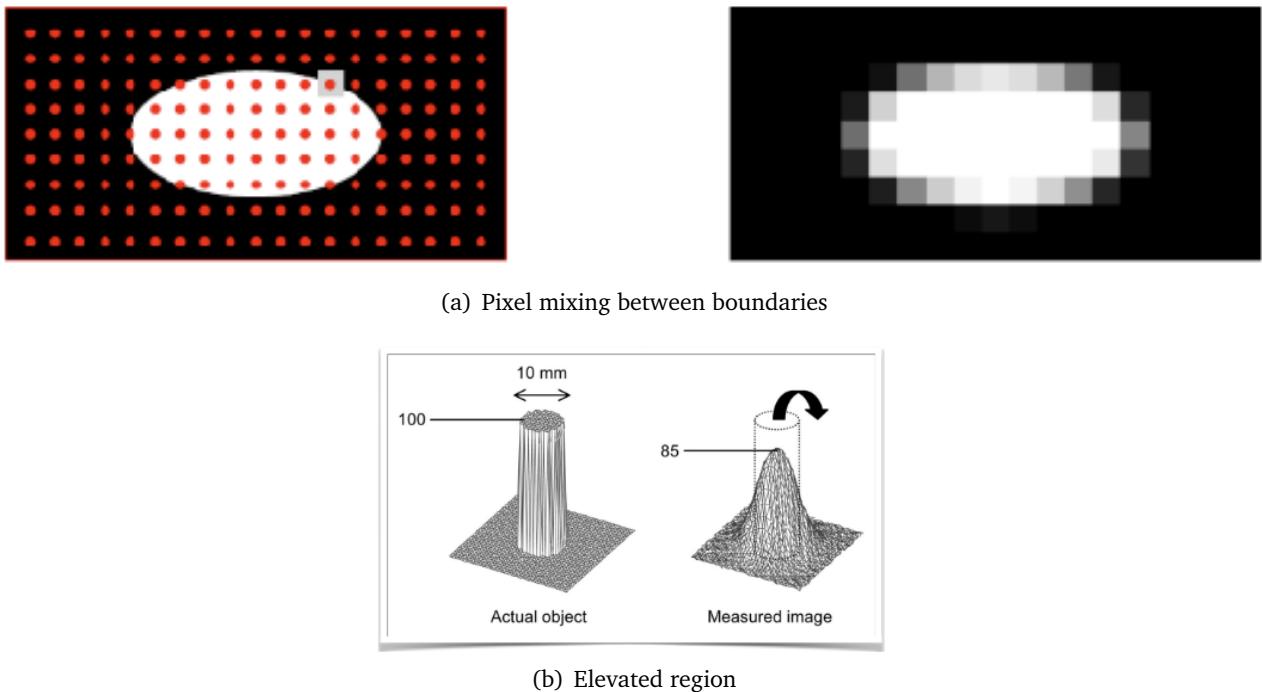


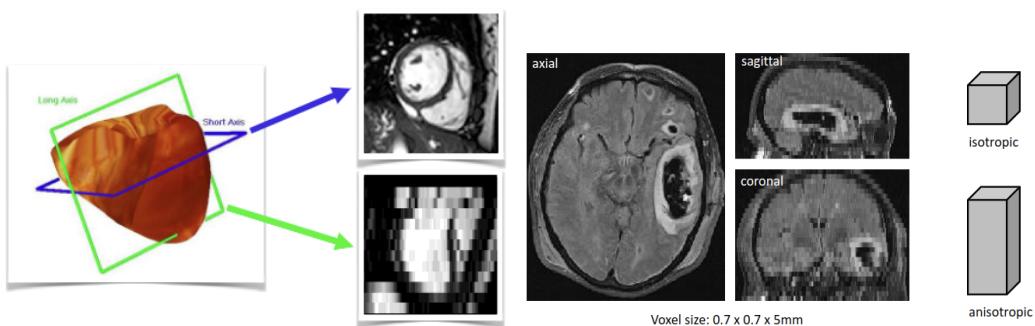
Figure 2: Partial volume effects

2.3 Intensity Inhomogeneities

You have the problem that you might have varying contrast and intensity differences across the image plane. In an ultrasound, the images are acquired from a sensor that sends ultrasound waves into the body so that they may be absorbed by the tissue. This causes lower levels to appear darker on the scan. However, the further down you go, the less signal you get back. Similarly, in an MRI we also have contrasting areas across the image.

2.4 Anisotropic Resolution

Often 2D stacks (x-y dimension) will have high resolution, however, in the z dimension the resolution may be larger, which causes less clarity when looking along this view.



2.5 Imaging Artifacts

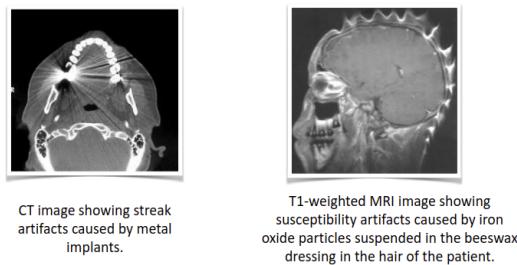


Figure 3: Image artifacts

2.6 limited contrasts

Different tissues can have similar physical properties and thus similar intensity values. Purely intensity-based algorithms are prone to fail or “leak” into adjacent tissues.

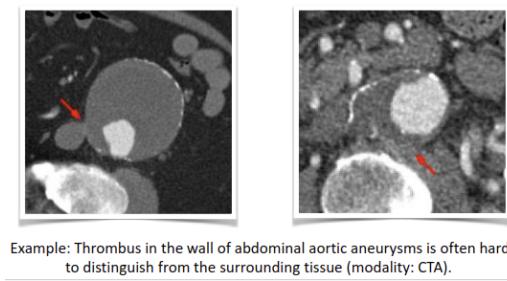


Figure 4: Limited contrast

2.7 Morphological Variability

There is variability between structures we want to segment. It makes it hard to incorporate meaningful prior information or useful shape models.

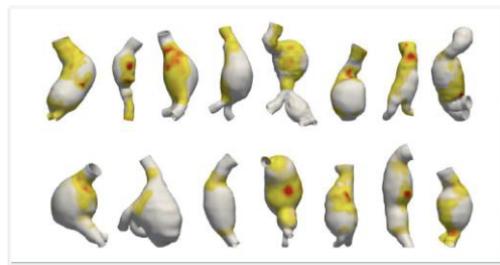


Figure 5: Morphological variability. A collection of abdominal aortic aneurysms acquired with PET-CT and colored by FDG-18 uptake values

3 Segmentation Evaluation

3.1 Ground truth

Definition 3.1. Reference or standard against which a method can be compared, e.g. the optimal transformation, or a true segmentation boundary.

In practice, it is difficult to obtain. We can establish a ground truth with synthetically obtained data, for example, simulated phantoms or around structures we manufactured, such as gel phantoms.

3.2 Gold standard

Usually, an expert manually segments an image.

The disadvantage, is that it

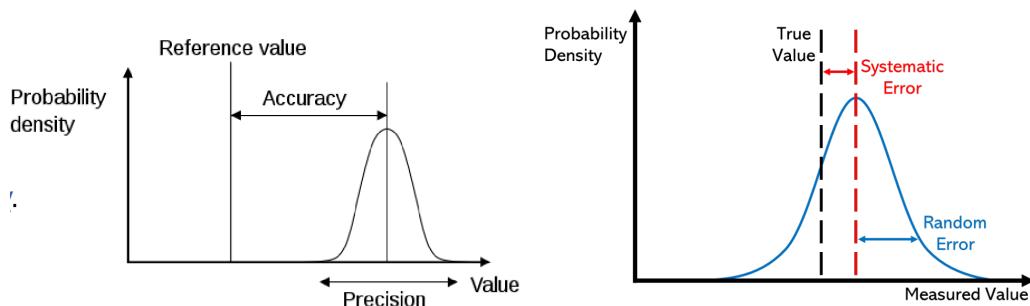
- requires training and is a tedious and time-consuming process.
- There is also intra-observer variability (disagreement between same observer on different occasions)
- and inter-observer variability (disagreement between observers)

The remedy, is that

- human observers can perform segmentation repeatedly,
- Multiple experts can perform segmentations,
- agreement or disagreement can be quantified

3.3 Evaluation Metrics

3.3.1 Precision



Is a description of **random errors**, a measure of **statistical variability**. It is the repeatability or reproducibility of the measurement.

3.3.2 Accuracy

More commonly, it is a description of **systematic errors**, a measure of **statistical bias**; as these cause a difference between a result and a “true” value, ISO calls this **trueness**.

Alternatively, ISO defines accuracy as describing a combination of both types of **observational error** above (random and systematic), so high accuracy requires both high precision and high trueness.

3.3.3 Robustness

The degradation in performance with respect to varying noise levels or varying artefacts.

3.3.4 Confusion matrix

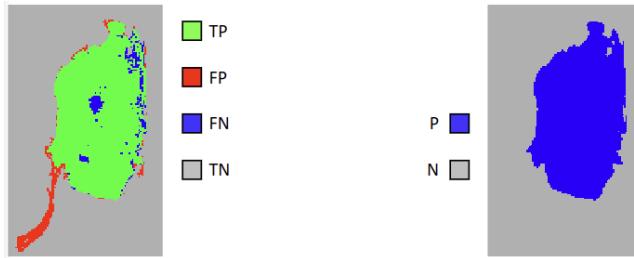


Figure 6: confusion matrix with TP true positive, TN correct rejection, FP false alarm, type I error, FN miss, type II error. Also, P is the number of real positive cases in the dataset, N is the number of real negative cases in the data.

3.3.5 Accuracy

Definition 3.2 (Accuracy).

$$\frac{TP + TN}{P + N} = \frac{TP + TN}{(TP + FN) + (TN + FP)}$$

3.3.6 Precision — positive predictive value

Definition 3.3 (Precision).

$$\frac{TP}{TP + FP}$$

3.3.7 Recall — sensitivity — hit rate — true positive rate

Definition 3.4 (Recall).

$$\frac{TP}{TP + FN}$$

3.3.8 Specificity — True negative rate

Definition 3.5 (Specificity).

$$\frac{TN}{N} = \frac{TN}{TN + FP}$$

3.3.9 F_β score

Definition 3.6 (F_β score).

$$(1 + \beta^2) \frac{Precision \cdot Recall}{(\beta^2 \cdot Precision) + Recall} = \frac{(1 + \beta^2)TP}{(1 + \beta^2)TP + \beta^2FP + FN}$$

“The F-measure is calculated as the harmonic mean of precision and recall, giving each the same weighting” [here](#)

The Fbeta-measure is a generalization of the F-measure that adds a configuration parameter called beta. A default beta value is 1.0, which is the same as the F-measure. A smaller beta value, such as 0.5, gives more weight to precision and less to recall, whereas a larger beta value, such as 2.0, gives less weight to precision and more weight to recall in the calculation of the score.

Where the harmonic mean is the “the reciprocal of the average of the reciprocals” [here](#)

The F-measure was derived so that F_β "measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision".[3] It is based on Van Rijsbergen's effectiveness measure [here](#)

3.3.10 F1 score

Definition 3.7 (F1 score).

$$2 \frac{Precision \cdot Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN}$$

It is tough to use two metrics independently, it is the harmonic mean between the two.

3.3.11 'IoU' - Overlap based - Jaccard Index

Definition 3.8 (Jaccard Index — Intersection over Union).

$$\frac{|A \cap B|}{|A \cup B|}$$

3.3.12 'DSC' - Overlap based - Dice Similarity

The most widely used measure for evaluating segmentation. Assume that A is the reference, and B is the prediction. Therefore, with $|A| = TP + FN$ and $|B| = TP + FP$, DSC is equivalent to F1.

Definition 3.9 (DICE).

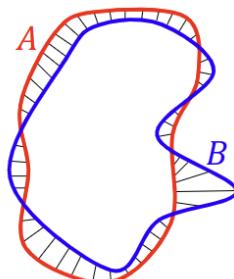
$$2 \frac{|A \cap B|}{|A| + |B|}$$

3.3.13 Volume similarity

Definition 3.10 (Volume similarity).

$$1 - \frac{||A| - |B||}{|A| + |B|} = 1 - \frac{|FN - FP|}{2TP + FP + FN}$$

3.3.14 'HD' - Surface Hasudorff distance



Definition 3.11 (Hausdorff distance).

$$\max(h(A, B), h(B, A)), \quad h(A, B) = \max_{a \in A} \min_{b \in B} ||a - b||$$

3.3.15 'ASSD' - Surface (symmetric) average surface distance

Definition 3.12 (Average surface distance).

$$ASD = \frac{d(A, B) + d(B, A)}{2}, \quad d(A, B) = \frac{1}{N} \sum_{a \in A} \min_{b \in B} \|a - b\|$$

3.4 Pitfalls in segmentation evaluation

“In a field such as athletics, this process is straightforward because the performance measurements (e.g., the time it takes an athlete to run a given distance) exactly reflect the underlying interest (e.g., which athlete runs a given distance the fastest?) [...] If the performance of an image analysis algorithm is not measured according to relevant validation metrics, no reliable statement can be made about the suitability of this algorithm in solving the proposed task, and the algorithm is unlikely to ever reach the stage of real-life application” [2].

3.4.1 Effect of structure size

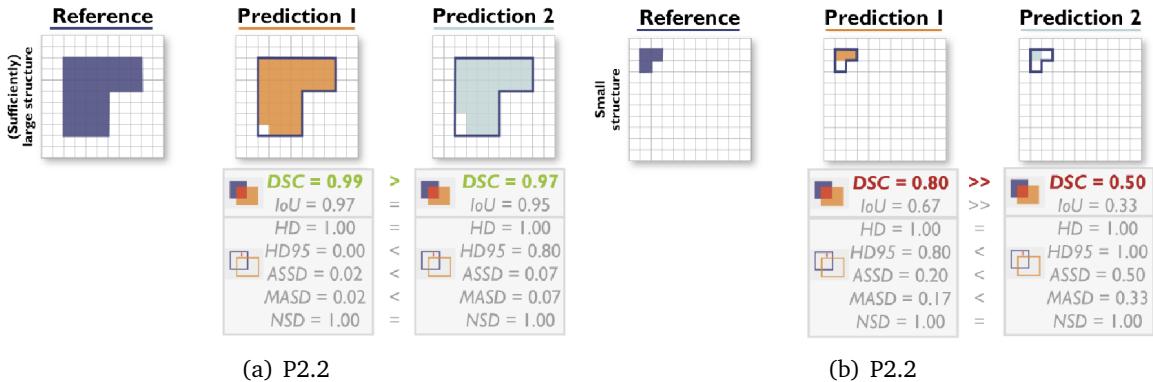


Figure 7: For small structure sizes: The predictions of two algorithms (Prediction 1/2) differ in only a single pixel. In the case of the small structure (right), this has a substantial effect on the corresponding Dice Similarity Coefficient (DSC) metric value (similar for the Intersection over Union (IoU)). This pitfall is also relevant for other overlap-based metrics such as the centerline Dice Similarity Coefficient (clDice), and localization criteria such as Box/Approx/Mask IoU and Intersection over Reference (IoR).

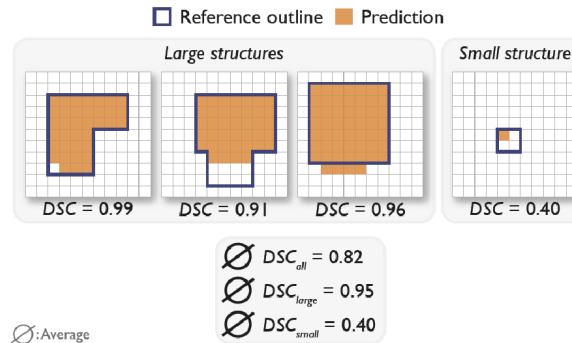


Figure 8

3.4.2 Effect of structure shape

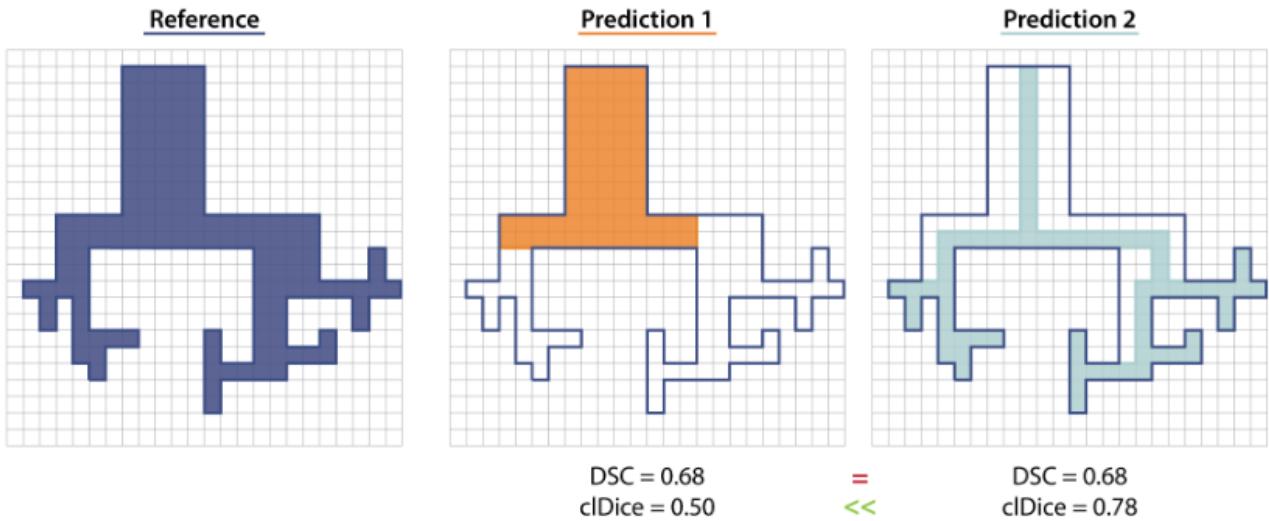


Figure 9: Extended Data Fig. Sn 2.14 [2]

Effect of complex shapes. Common overlap-based metrics such as the Dice Similarity Coefficient (DSC) are unaware of complex structure shapes and treat Predictions 1 and 2 equally. The centerline Dice Similarity Coefficient (cIDice) uncovers that Prediction 1 misses the fine-granular branches of the reference and favors Prediction 2, which focuses on the object's center line and better captures its fine branches. This pitfall is also relevant for other overlap-based metrics such as Intersection over Union (IoU) and pixel-level F_β Score, and localization criteria such as Box/Approx/Mask IoU, Center Distance, Mask IoU ≥ 0 , Point inside Mask/Box/Approx, and Intersection over Reference (IOR).

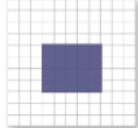
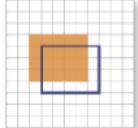
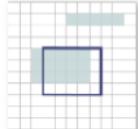
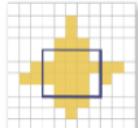
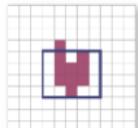
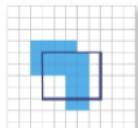
Reference	DSC	IoU	HD	HD95	ASSD	MASD	NSD
							
Prediction 1	DSC = 0.6	IoU = 0.4	HD = 1.4	HD95 = 1.3	ASSD = 0.9	MASD = 0.9	NSD = 1.0
							
Prediction 2	DSC = 0.6	IoU = 0.4	HD = 3.6	HD95 = 3.1	ASSD = 1.0	MASD = 1.0	NSD = 0.7
							
Prediction 3	DSC = 0.6	IoU = 0.4	HD = 3.0	HD95 = 2.0	ASSD = 0.8	MASD = 0.7	NSD = 0.8
							
Prediction 4	DSC = 0.6	IoU = 0.4	HD = 2.2	HD95 = 2.0	ASSD = 0.8	MASD = 0.7	NSD = 0.8
							
Prediction 5	DSC = 0.6	IoU = 0.4	HD = 2.0	HD95 = 1.2	ASSD = 0.8	MASD = 0.8	NSD = 0.9
							

Figure 10

3.4.3 Effect of spatial alignment

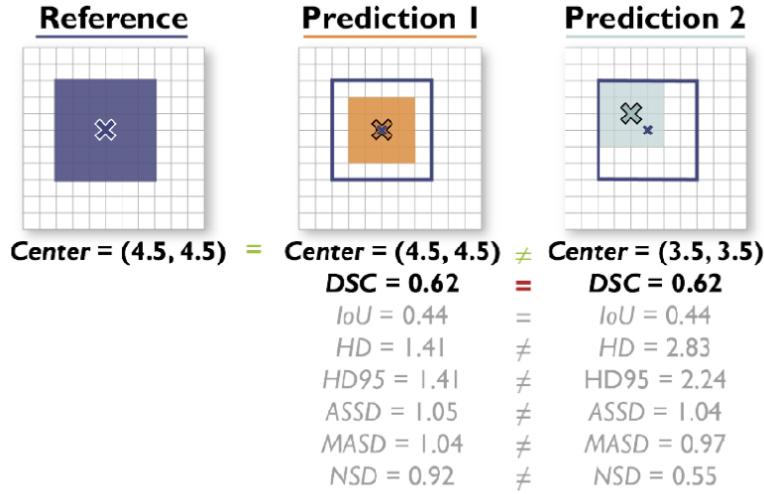


Figure 11: Extended Data Fig. SN 2.7 [2]

The most common counting-based metrics are poor proxies for the center point alignment. Here, Predictions 1 and 2 yield the same Dice Similarity Coefficient (DSC) value although Prediction 1 approximates the location of the object much better. This pitfall is also relevant for other boundary- and overlap-based metrics such as Average Symmetric Surface Distance (ASSD), Boundary Intersection over Union (IoU), Hausdorff Distance (HD), Hausdorff Distance 95th Percentile (HD95), IoU, pixel-level F_β Score, and Mean Average Surface Distance (MASD), and localization criteria such as Box/Approx/Mask IoU, Mask IoU $\neq 0$, Point inside Mask/Box/Approx, Boundary IoU, and Intersection over Reference (IoR).

3.4.4 Effect of holes

Boundary-based metrics commonly ignore the overlap between structures and are thus insensitive to holes in structures.

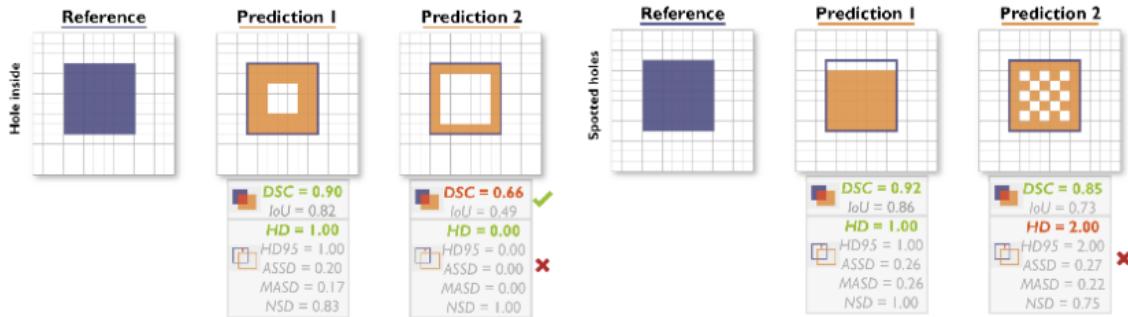


Figure 12: Extended Data Fig. SN 2.6 [2]

Boundary-based metrics commonly ignore the overlap between structures and are thus insensitive to holes in structures. In the examples, the Prediction respectively features a hole or spotted pattern within the object. Boundary-based metrics (here: Normalized Surface Distance (NSD)) do not recognize this problem, yielding (near) perfect metric scores of 1.0 and 0.9, whereas the volumetric difference reflects the fact that the inner area is inadequately predicted. NSD was calculated for $\tau = 2$. This pitfall is also relevant for other boundary-based metrics such as Average Symmetric Surface Distance (ASSD), Boundary Intersection over Union (Boundary IoU), Hausdorff Distance (HD), Hausdorff Distance 95th Percentile (HD95), and Mean Average Surface Distance (MASD), as well as localization criteria such as Center Distance, Mask IoU $\neq 0$, Point inside Mask/Box/Approx, Boundary IoU, Intersection over Reference (IoR), and Mask IoU .

3.4.5 Effect of Annotation noise

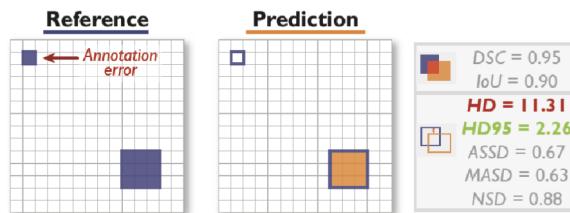
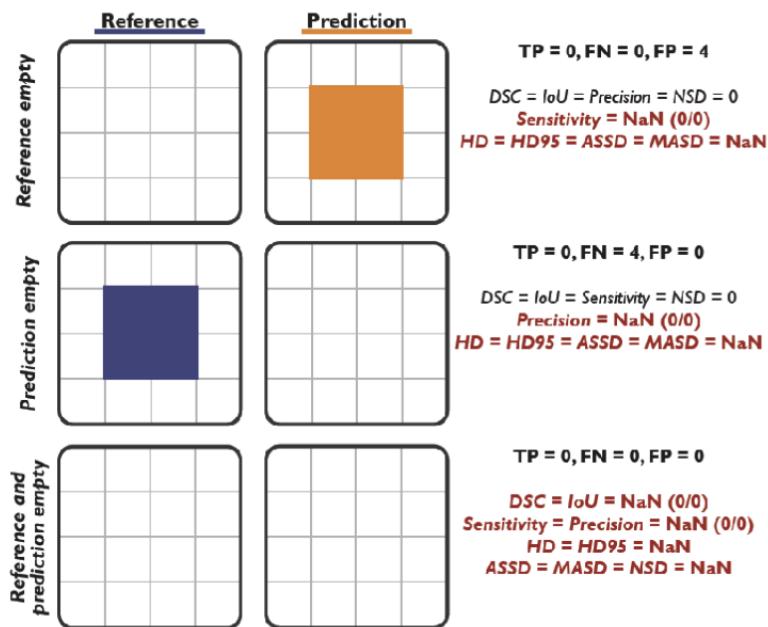


Figure 13

3.4.6 Effect of empty labelmaps



3.4.7 Effect of resolution

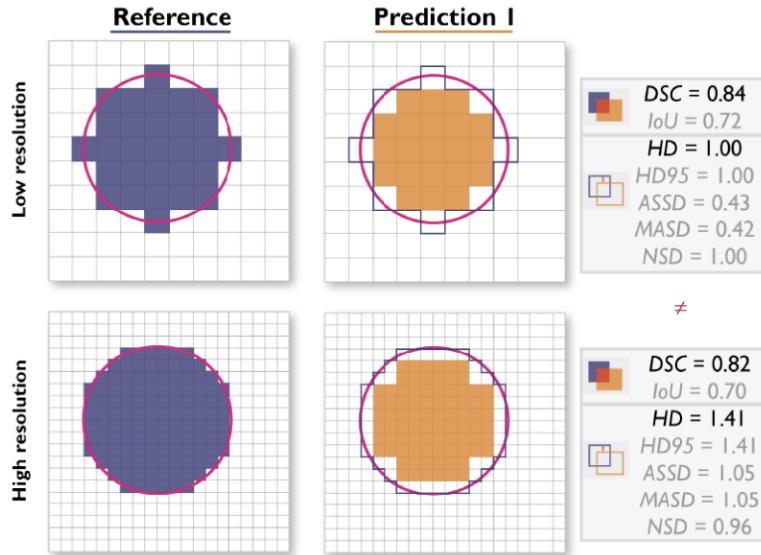


Figure 14: Effect of different grid sizes. Extended Data Fig. SN 2.36 [2]

Differences in the grid size (resolution) of an image highly influence the image and the reference annotation (dark blue shape (reference) vs. pink outline (desired circle shape)), with a prediction of the exact same shape leading to different metric scores. Abbreviations: Dice Similarity Coefficient (DSC), Intersection over Union (IoU), Hausdorff Distance (HD), Hausdorff Distance 95th Percentile (HD95), Average Symmetric Surface Distance (ASSD), Mean Average Surface Distance (MASD), Normalized Surface Distance (NSD).

3.5 Preference for oversegmentation to undersegmentation

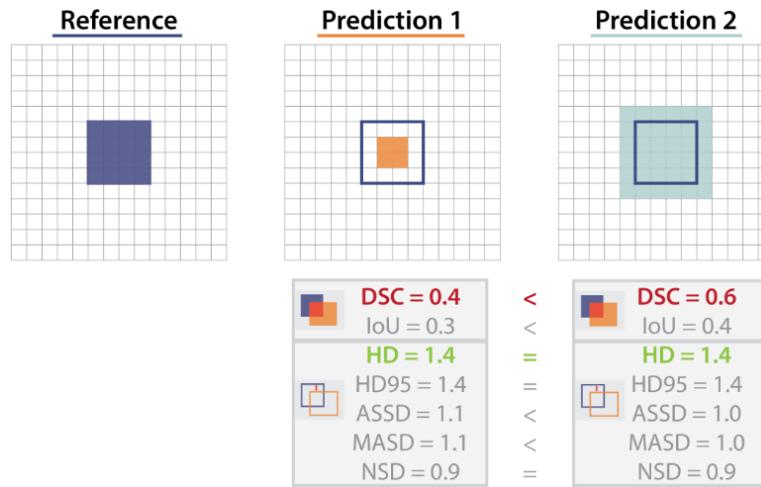


Figure 15: Extended Data Fig. SN 2.10 [2]

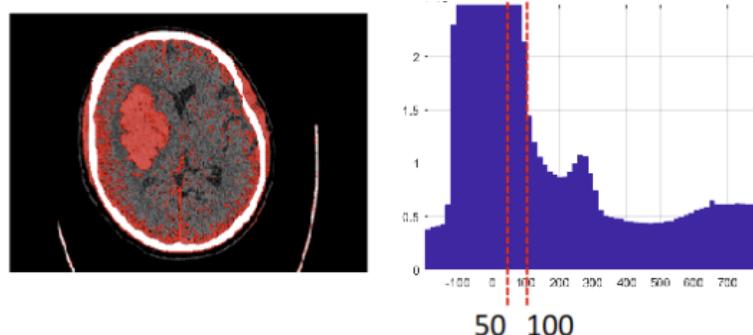
The outlines of the predictions of two algorithms (Prediction 1/2) differ in only a single layer of pixels (Prediction 1: undersegmentation — smaller structure compared to reference, Prediction 2: oversegmentation — larger structure compared to reference).

If penalizing of either over- or undersegmentation is desired (unequal severity of class confusions), other metrics such as the F_β Score provide specific penalties for either depending on the chosen hyperparameter β . This pitfall is also relevant for other overlap-based metrics such as centerline Dice Similarity Coefficient (clDice)

4 Segmentation Methods

4.1 Intensity-based segmentation — thresholding

By generating a histogram of values from a scan, you can choose which pixels to keep based on a threshold. When one threshold doesn't work initially, you can use multiple thresholds with upper and lower thresholds



4.1.1 Advantages

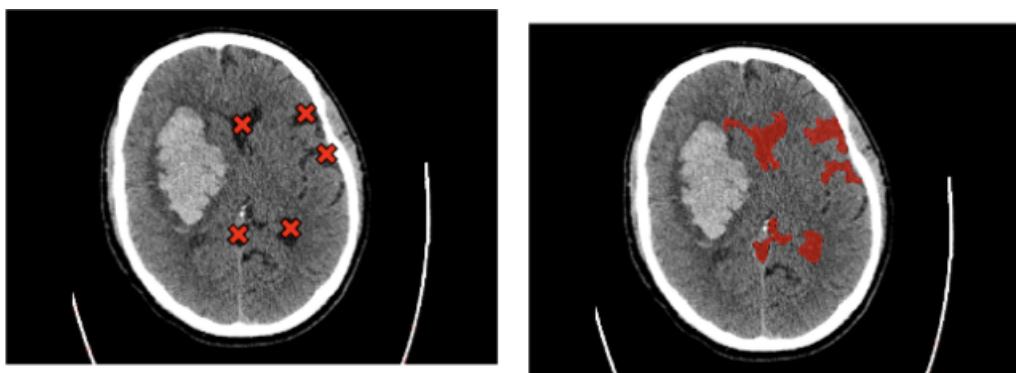
Simple and fast

4.1.2 Disadvantages

- regions must be homogeneous and distinct
- difficulty in finding consistent thresholds across images
- leakages, isolated pixels and ‘rough’ boundaries likely

4.2 Region-based — region growing

Start from (user selected) seed point(s) and grow a region according to an intensity threshold.



4.2.1 Advantages

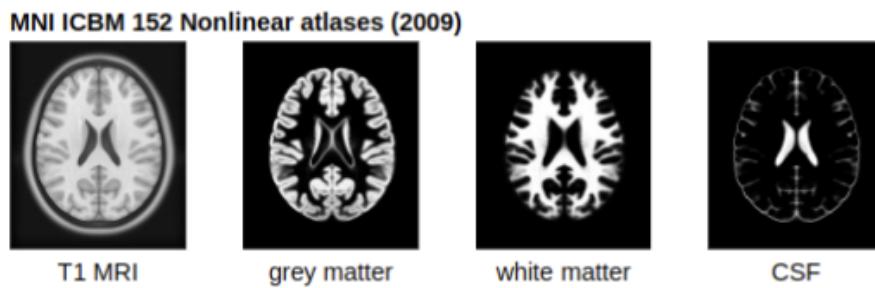
relatively fast and yields connected regions (From a seed point)

4.2.2 Disadvantages

- regions must be homogeneous
- leakages and ‘rough’ boundaries likely
- requires (user) input for seed points

4.3 *Atlas-based segmentation*

An atlas is some kind of prototype or exemplar of the anatomy that we want to segment, or a template of the object we would like to segment.

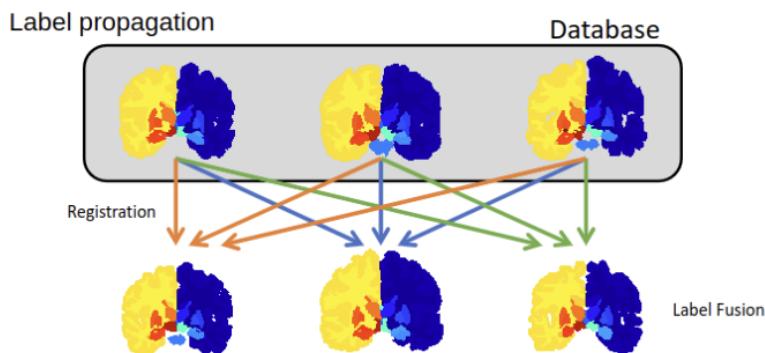


Atlases usually have geometric information about points, curves or surfaces, or label information about voxels (Anatomical regions or function). Atlases are usually constructed from example data: single subjects or populations of subjects e.g. by averaging to produce probabilistic atlases.

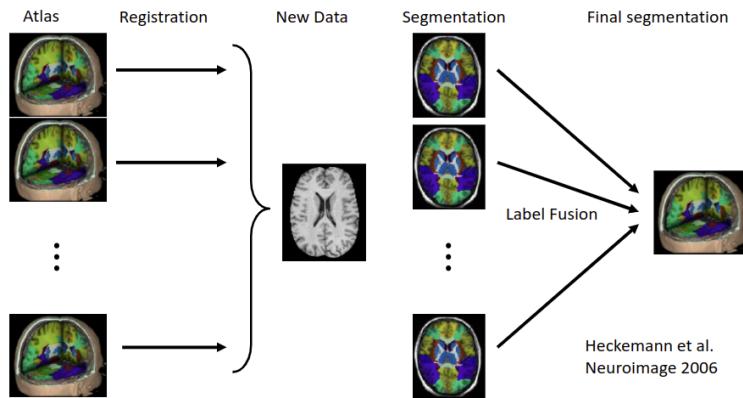
The image above isn't a specific patient, but it is a statistical population average with probability maps for where brain matter is. We can use these as priors to segment other patients.

4.4 *Segmentation using Registration*

Uses atlases to mutate and transform existing segmented images (or atlases) to form the same properties as the image you're trying to segment now. By using more than one image, the target image now has multiple predictions of the segmentation, and a label fusion must occur.



4.5 Multi-Atlas Label Propagation



Upon fusion, it is possible to keep a probability distribution to indicate that there was contention between different sources.

4.5.1 Advantages

- robust and accurate (like ensembles)
- yields plausible segmentations
- fully automatic

4.5.2 Disadvantages

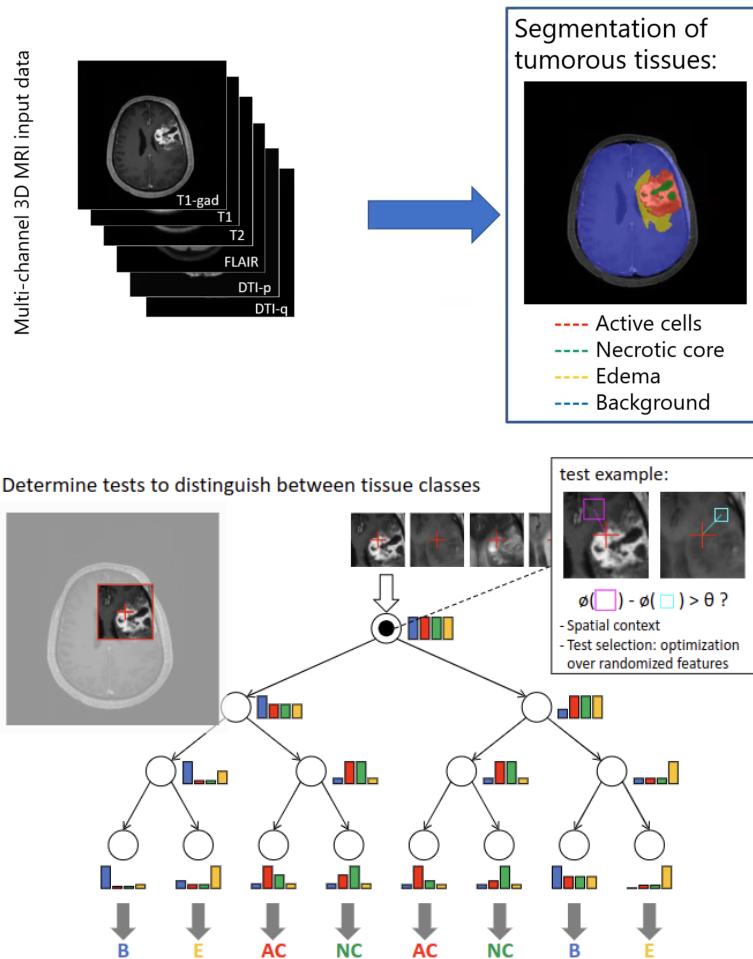
- computationally expensive
- cannot deal well with abnormalities
- not suitable for tumour segmentation

4.6 Learning-based segmentation — random forests, convolutional neural networks

4.6.1 Random forests

Begin by taking different 3D views or obtain multiple contrasts of the same anatomy but in a different way (based on different physical scans) of the same target. Afterwards, combine them into a 4D tensor, and predict a segmentation of the tumorous tissues. It is great because all patients are different.

At each branch, decide a splitting criteria that will give you the best possible split among the data. At training, the tree tries to cluster classes together.



[paper1](#) and [paper2](#)

4.6.2 Advantages

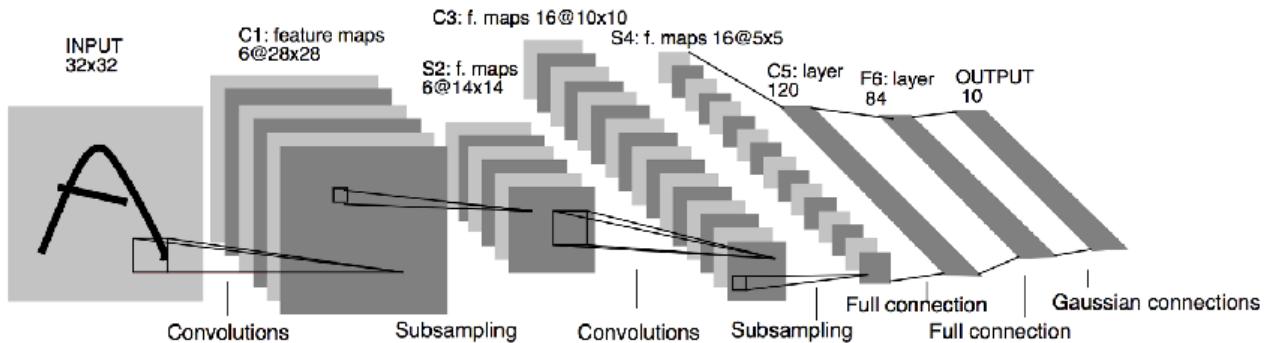
- ensemble classifiers are robust and accurate
- computationally efficient (can run in parallel)
- fully automatic

4.6.3 Disadvantages

- shallow model, no hierarchical features
- no guarantees on connectedness

5 Segmentation via Dense Classification

5.1 LeNet



```

1 # from https://pytorch.org/tutorials/beginner/blitz/neural_networks_tutorial.html
2 # https://www.analyticsvidhya.com/blog/2023/11/lenet-architectural-insights-and-practical-implementation/
3 import torch
4 import torch.nn as nn
5 import torch.nn.functional as F
6
7
8 class Net(nn.Module):
9
10     def __init__(self):
11         super(Net, self).__init__()
12         # 1 input image channel, 6 output channels, 5x5 square convolution
13         # kernel
14         self.conv1 = nn.Conv2d(1, 6, kernel_size=5)
15         self.pool1 = nn.AvgPool2d(2, stride=2)
16         self.conv2 = nn.Conv2d(6, 16, kernel_size=5)
17         self.pool2 = nn.AvgPool2d(2, stride=2)
18         # an affine operation: y = Wx + b
19         self.fc1 = nn.Linear(16 * 5 * 5, 120)
20         self.fc2 = nn.Linear(120, 84)
21         self.fc3 = nn.Linear(84, 10)
22
23     def forward(self, x):
24         x = self.conv1(x)
25         x = F.relu(x)
26         x = self.pool1(x)
27         x = self.conv2(x)
28         x = F.relu(x)
29         x = self.pool2(x)
30
31         x=torch.flatten(x,1) # flatten all dimensions except the batch dimension
32
33         x = self.fc1(x)
34         x = F.relu(x)
35         x = self.fc2(x)
36         x = F.relu(x)
37         x = self.fc3(x)
38
39         return x

```

code/LeNet.py

5.1.1 Fully convolutional LeNet

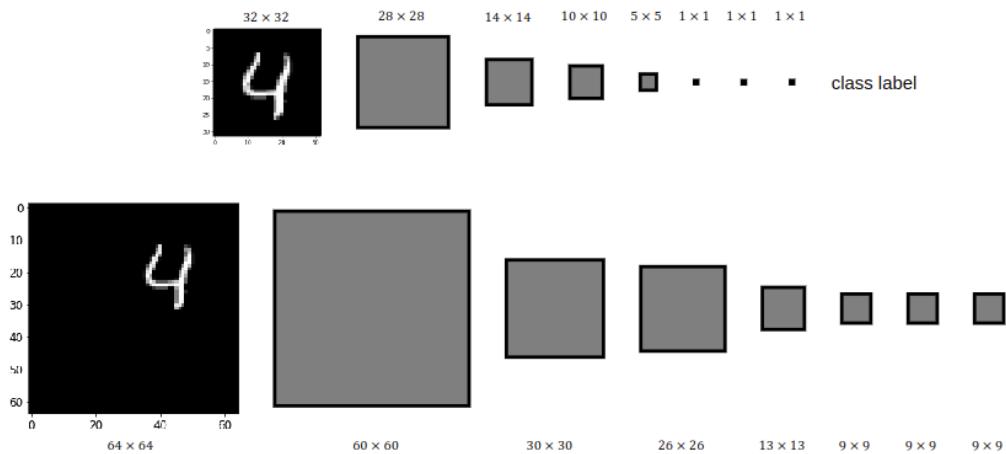
Pay attention to line 19 onwards. If we wanted to translate the network into a fully convolutional LeNet:

```

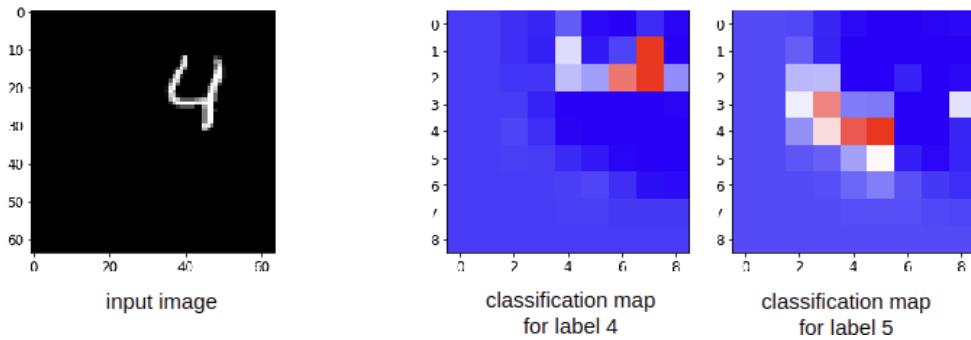
1 import torch
2 import torch.nn as nn
3 import torch.nn.functional as F
4
5
6 class Net(nn.Module):
7
8     def __init__(self):
9         super(Net, self).__init__()
10        # 1 input image channel, 6 output channels, 5x5 square convolution
11        # kernel
12        self.conv1 = nn.Conv2d(1, 6, kernel_size=5)
13        self.pool1 = nn.AvgPool2d(2, stride=2)
14        self.conv2 = nn.Conv2d(6, 16, kernel_size=5)
15        self.pool2 = nn.AvgPool2d(2, stride=2)
16        self.conv3 = nn.Conv2d(16, 120, kernel_size=5) # changed
17        self.conv4 = nn.Conv2d(120, 84, kernel_size=1) # changed
18        self.conv5 = nn.Conv2d(84, 10, kernel_size=1) # changed
19
20    def forward(self, x):
21        x = self.conv1(x)
22        x = F.relu(x)
23        x = self.pool1(x)
24        x = self.conv2(x)
25        x = F.relu(x)
26        x = self.pool2(x)
27
28        x = self.conv3(x)
29        x = F.relu(x)
30        x = self.conv4(x)
31        x = F.relu(x)
32        x = self.conv5
33
34    return F.log_softmax(x, dim=1)

```

code/LeNet–fullyconnected.py

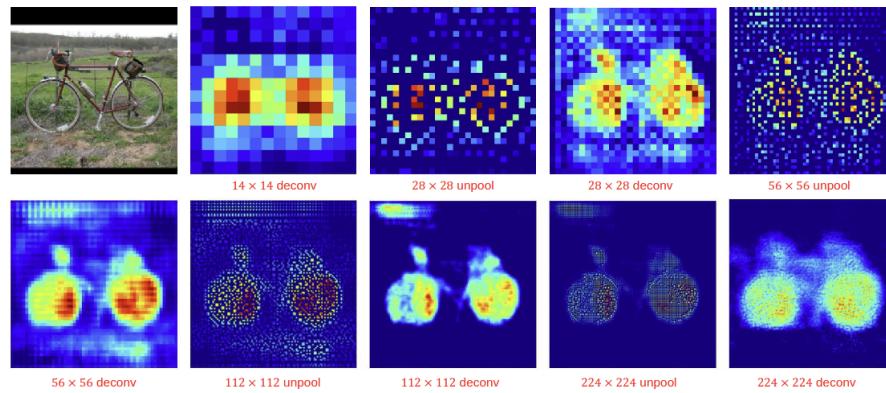
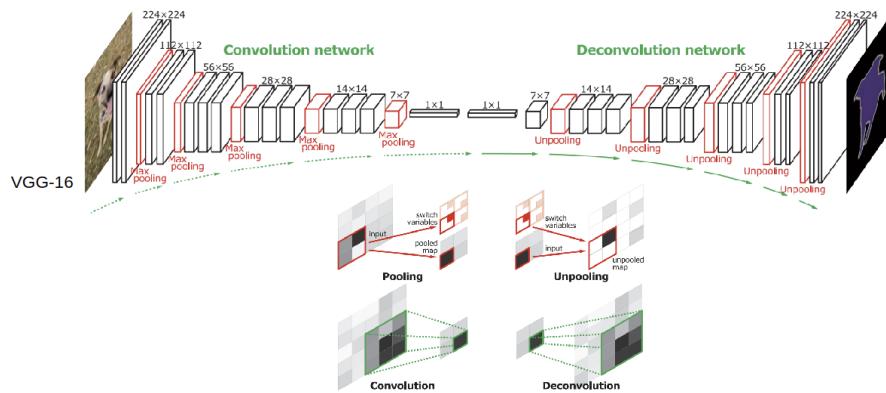


We go beyond classification. If we're given a 64x64 image, at the end of the network, we get an output feature-map.



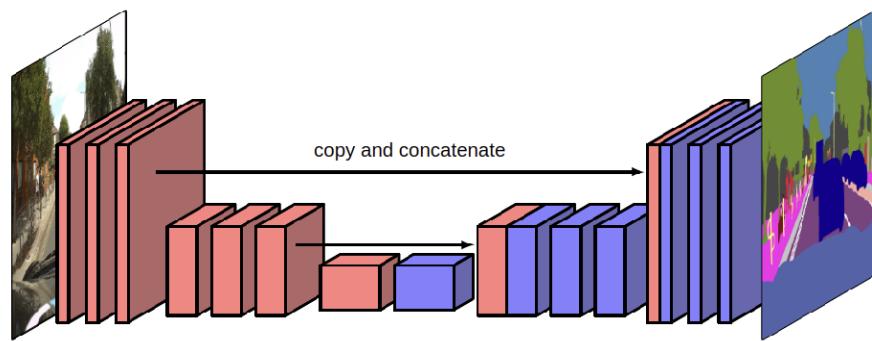
The classification maps can be used to localise where it is, and that there is a number there (a 4). The 5, is garbage because there is a 5. This input is scaled up through up-sampling (Section 6.1.1)

6 Encoder-Decoder Networks



The idea is to take a really low level representation of an image and try to upsample it back to its original resolution. This is what the U-Net is trying to achieve.

6.1 U-Net



6.1.1 Upsampling

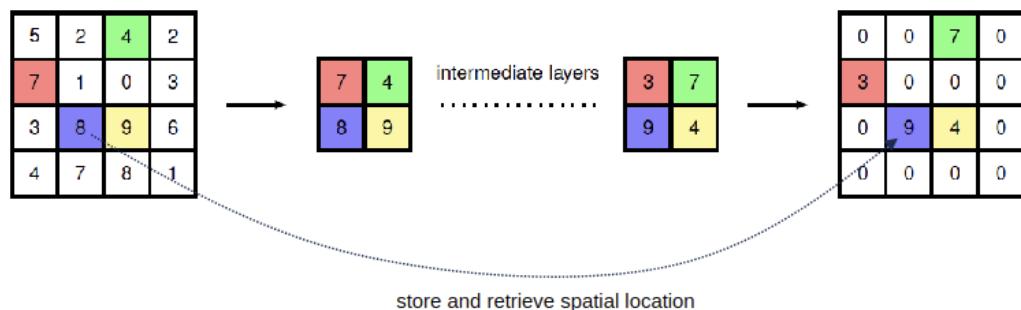
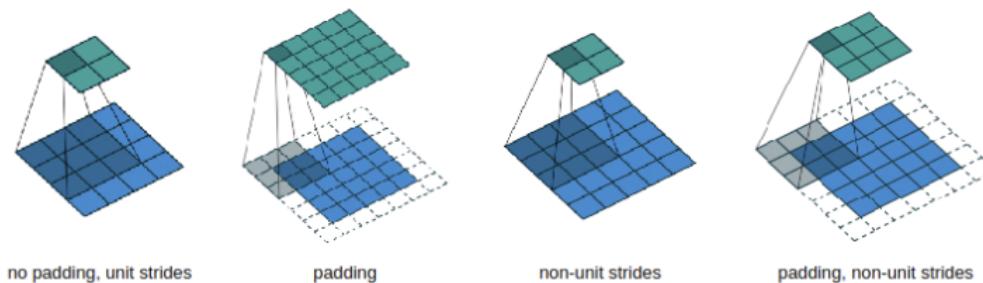


Figure 16

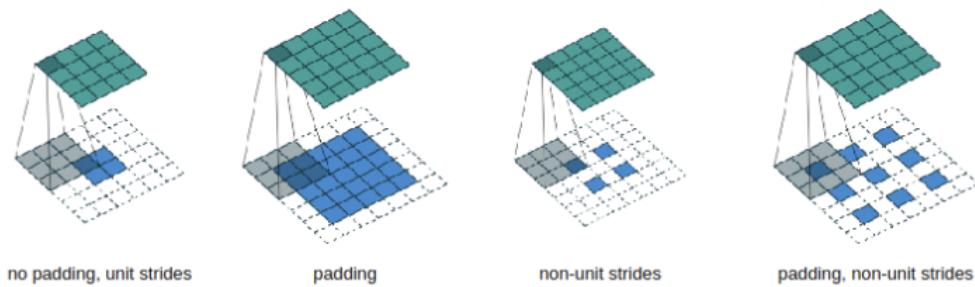
6.1.2 Convolutions



Convolutions can be used for 'learned' down-sampling

[link](#)

6.1.3 Transpose convolutions

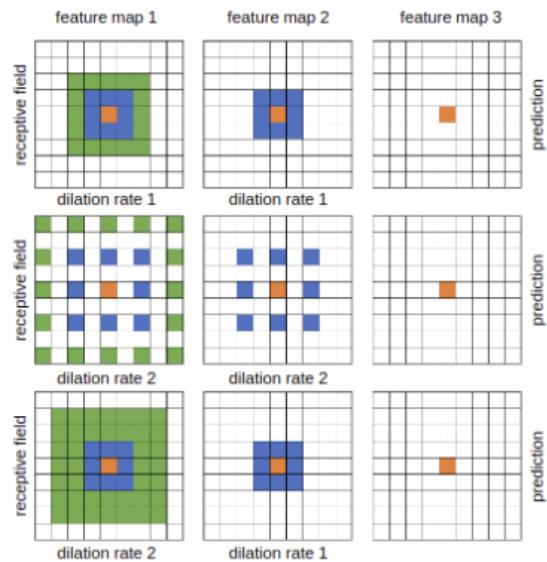
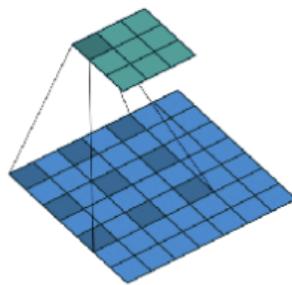


Transpose convolutions can be used for 'learned' up-sampling

[link](#)

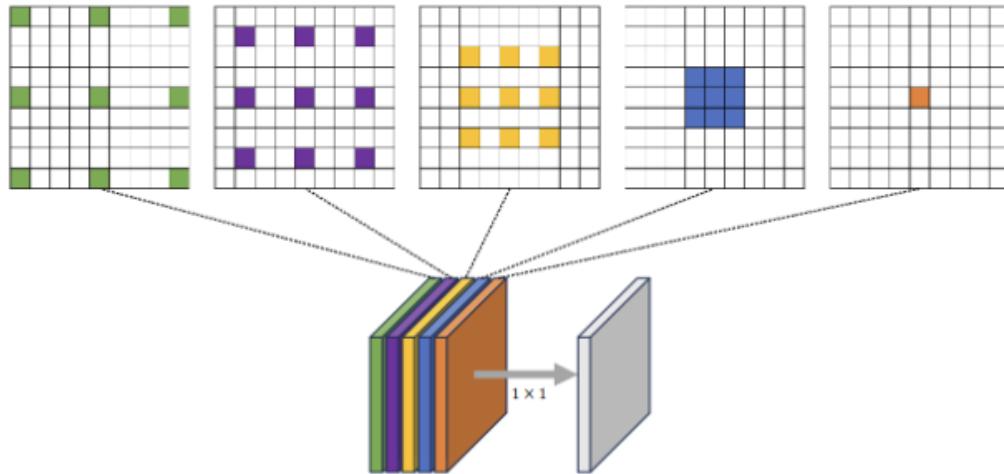
6.1.4 Dilated convolutions

Dilated convolutions



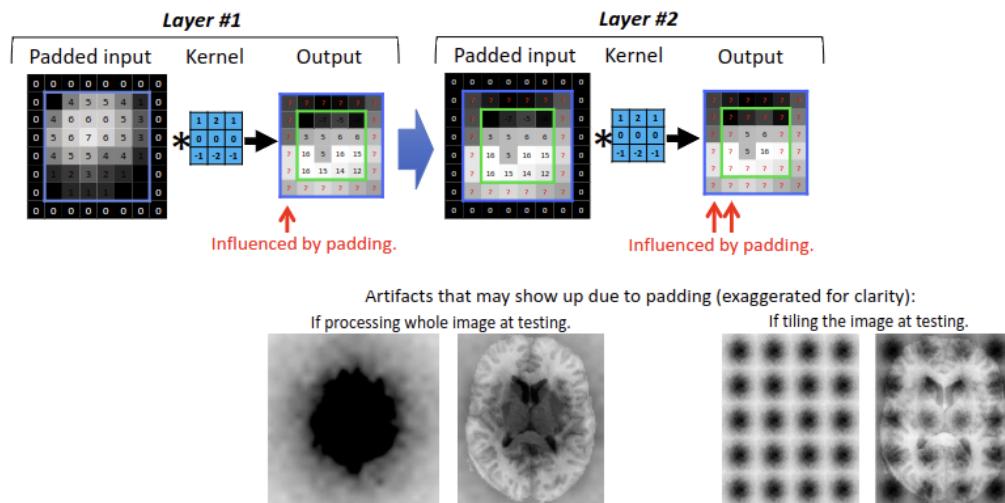
[link](#)

6.1.5 Atrous spatial pyramid pooling



[link](#)

6.1.6 Padding effects

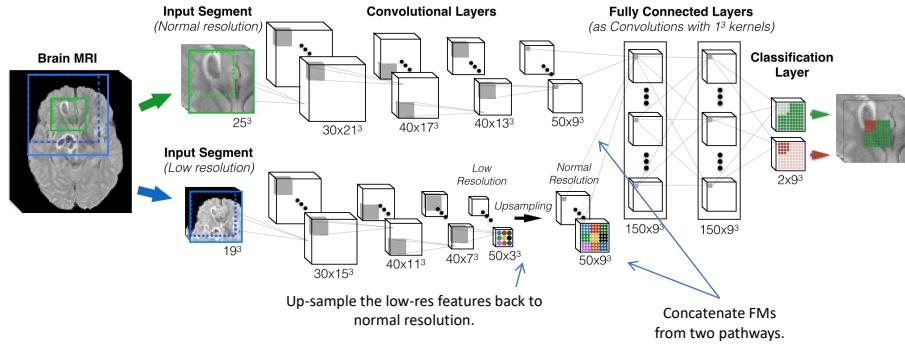


the padding may propagate values in the features, or if we used tiling or sliding window approach, we may get a repeating pattern.

6.1.7 Multi-scale processing

Multi-scale processing

- What else can we do to make the network “seeing” more context?
- Idea: Add pathways which process down-sampled images

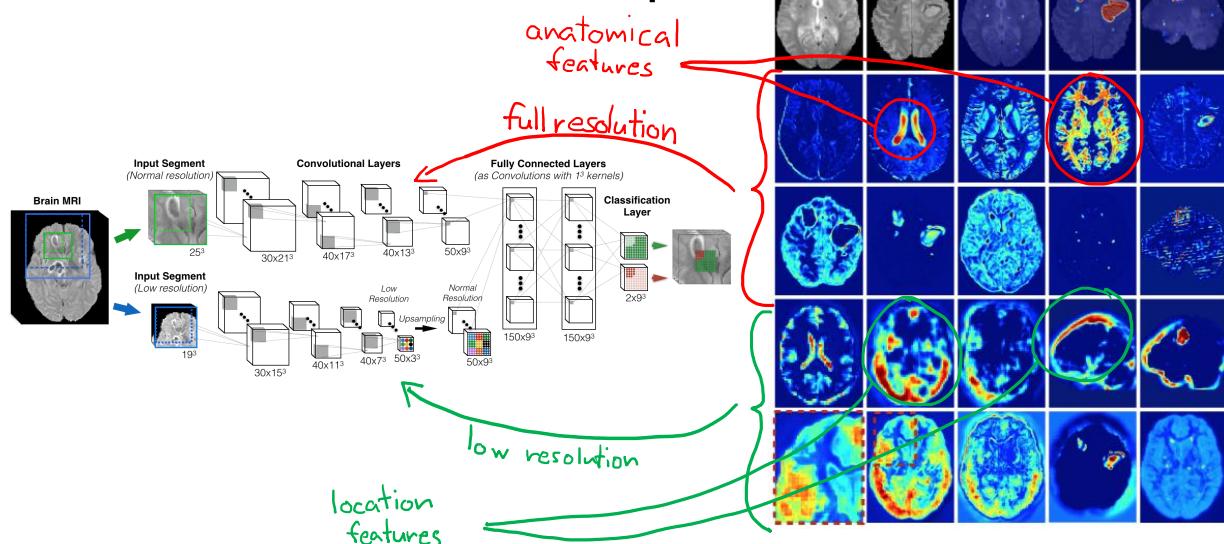


Kamnitsas et al. 2017. [Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation](#)

129

the idea of not using a decoder that given an input patch, we process it (without maxpooling) by applying convolutional layers and then we have the kernel of fully connected layers. Here, we have a receptive field, i.e. how much information we can see. The solution here is to take another network and process in parallel, the image at a lower resolution. Here we can see more spatial context.

Multi-scale feature maps

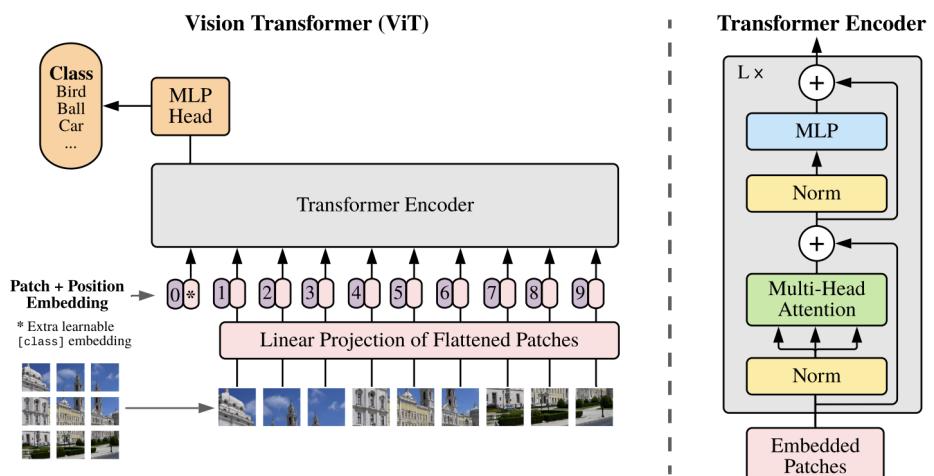


Kamnitsas et al. 2017. [Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation](#)

130

6.2 Vision transformers

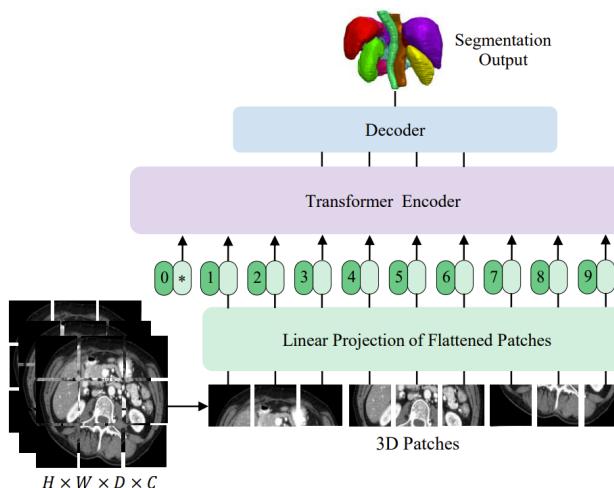
Vision transformers



Dosovitskiy et al. 2021. [An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale](#)

134

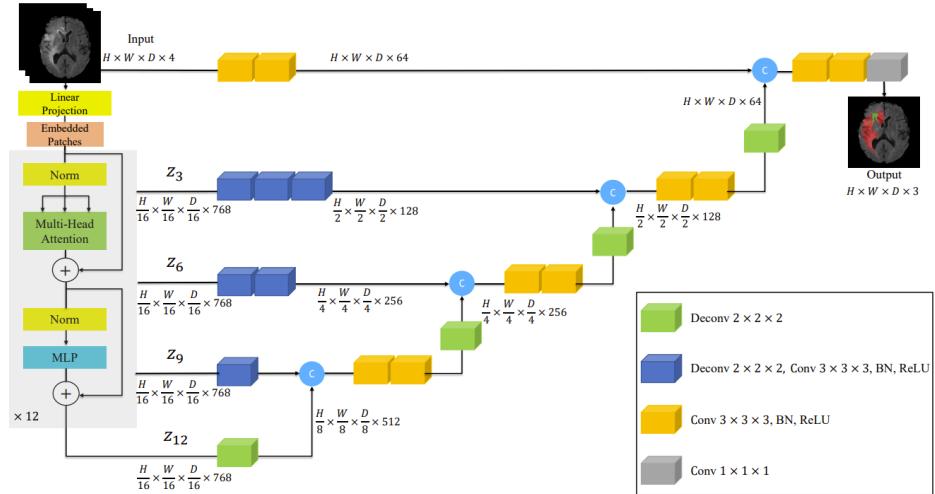
Transformers for image segmentation



Hatamizadeh et al. 2021. [UNETR: Transformers for 3D Medical Image Segmentation](#)

135

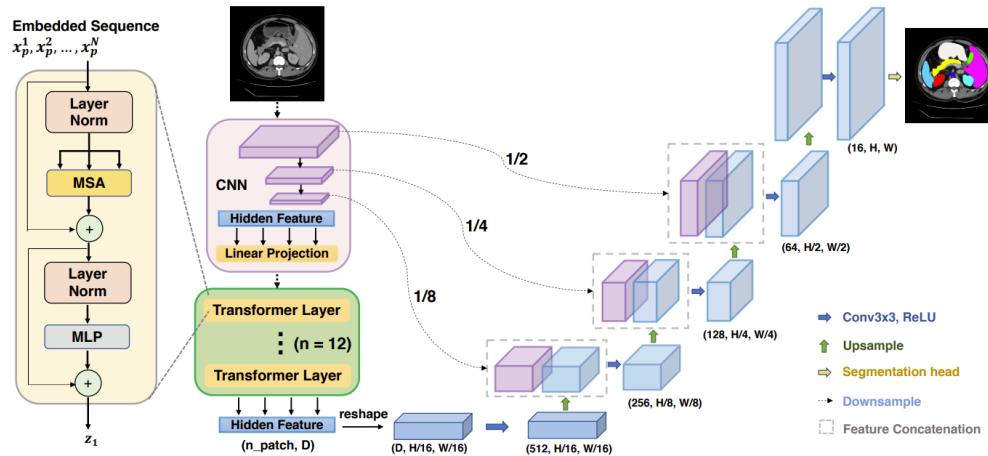
Transformers for image segmentation



Hatamizadeh et al. 2021. [UNETR: Transformers for 3D Medical Image Segmentation](#)

136

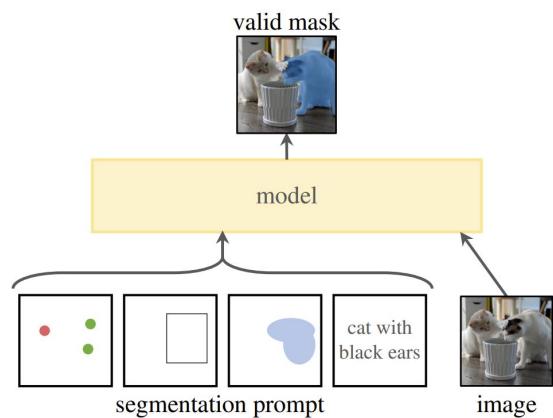
Transformers for image segmentation



Chen et al. 2021. [TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation](#)

137

Transformers for image segmentation



Kirillov et al. 2023. [Segment Anything](#)

138

References

- [1] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2020. arXiv: [2010.11929 \[cs.CV\]](#).
- [2] Annika Reinke et al. *Understanding metric-related pitfalls in image analysis validation*. 2023. arXiv: [2302.01790 \[cs.CV\]](#).