ITEA 3

IMPACT

Intelligence based iMprovement of Personalized medicine And Clinical workflow supporT

DELIVERABLE D2.3.1

Report on state of the art deep learning methods for 3D image segmentation

Project number:	ITEA 17021
Document version no.:	v 1.0
Edited by:	Anders Eklund, Linköping University
Date:	June 26, 2019

ITEA Roadmap domains:
Major: Group

ITEA Roadmap categories:
Major: Content & knowledge
Minor: Interaction

HISTORY

Document version #	Date	Remarks
V0.1	2019-05-15	First version, created by LiU, Elekta, Quantib, LUMC
V0.2	2019-05-28	Minor revisions
V0.3	2019-06-19	Revisions after review
V1.0	2019-06-26	Approved

Deliverable review procedure:

- **2 weeks before due date**: deliverable owner sends deliverable –approved by WP leader– to Project Manager
- **Upfront** PM assigns a co-reviewer from the PMT group to cross check the deliverable
- **1 week before due date**: co-reviewer provides input to deliverable owner
- **Due date:** deliverable owner sends the final version of the deliverable to PM and co-reviewer

Table of Contents

1 Glossary

ANN	Artificial neural network
CNN	Convolutional neural network
CPU	Central processing unit
CT	Computed tomography
FCN	Fully convolutional network
GAN	Generative adversarial network
GPU	Graphics processing unit
MRI	Magnetic resonance imaging
OCT	Optical coherence tomography
PET	Positron emission tomography
SPECT	Single-photon emission computed tomography

2 Introduction

Medical image segmentation is the process of automatic or semi-automatic detection of boundaries within 2D or 3D data. A major difficulty of medical image segmentation is the high variability in medical images. First and foremost, the human anatomy itself shows major modes of variation. Furthermore many different acquisition modalities (X-ray, CT, MRI, microscopy, PET, SPECT, Endoscopy, OCT, and many more) are used to create medical images. The result of the segmentation can be used to obtain further diagnostic insights. Depending on the application, manual segmentation of a single dataset can take several hours. For example, Harari et al. (2010) state up to 4 h for manual segmentation of head & neck patients. Another problem is segmentation variability, e.g. that radiologists do nog agree what the manual segmentation should look like (Nelms et al., 2010, Sandström et al., 2016). For these reasons, fast and fully automatic segmentation is often desirable.

Lately, the performance of traditional image segmentation methods (intensity or model based) has been surpassed by deep learning methods. This document surveys deep learning methods for medical image segmentation of 3D data. The survey contains an overview of 2D solutions, which process a volume slice by slice, 2.5D solutions, which use a limited amount of 3D information, as well as true 3D solutions. It is assumed that the reader has a basic understanding of machine learning, such as artificial neural networks.

3 Survey of 2D deep learning methods for segmentation

Deep learning algorithms, in particular convolutional neural networks (CNN), have rapidly become a methodology of choice for analyzing medical images. One of the first CNNs triggering interest in medical image segmentation is LeNet (LeCun et al., 1998). At that time, the method was restricted to small images as hardware was not available to accommodate larger models. In 2012, hardware advanced and AlexNet (Krizhevsky et al., 2012), a model very similar to LeNet, launched the current deep learning boom by winning the 2012 ILSVRC (ImageNet) competition by a huge margin.

A CNN, like LeNet and AlexNet, is a particular kind of artificial neural network (ANN) aimed at preserving spatial relationships in the data, with very few connections between the layers. The advantage of a CNN compared to an ANN is that it contains much fewer parameters; only the filter coefficients need to be learned. To learn 100 filters of size 3 x 3 only requires learning 900 parameters, plus 100 bias terms. On the other hand, training an ANN would involve learning many weights for every pixel in an image (which can easily have 1 million pixels). The input to a CNN is arranged in a grid structure and then fed through layers that preserve these relationships, each layer operating on a small region of the previous layer. CNNs are able to form highly efficient representations of the input data, well-suited for image-oriented tasks. In addition, CNNs typically have fully connected layers at the end, which compute the final outputs. Other common elements in many modern CNNs include dropout regularization and batch normalization.

Most CNNs were first designed to segment 2D images and later on extended to 3D images. 2D CNNs use 2D convolutional kernels (filters) to predict the segmentation map from a 2D image. Medical images are often 3D data consisting of multiple 2D slices. Segmentation maps are predicted for a full volume by taking predictions one 2D slice at a time. The 2D convolutional kernels are able to leverage context across the height and width of the slice to make predictions. However, because 2D CNNs take a single slice as input, they inherently fail to leverage context from adjacent slices (Lundervold and Lundervold, 2018). Therefore, the major drawback of 2D CNNs compared to 3D CNNs is the segmentation performance. Advantages of 2D CNNs are that the networks are faster to train and they are less hardware demanding as they require less memory than 3D CNNs. An overview of popular 2D CNNs is given in Table 1.

Table 1, overview of popular 2D CNNs. which can be used for segmentation

AlexNet (Krizhevsky et al., 2012)	One of the first convolutional networks that outperformed all the prior competitors. Notable features include the use of RELUs, dropout regularization, splitting the computations on multiple GPUs, and using data augmentation during training.
VGG (Simonyan and Zisserman, 2014)	Popularized the idea of using smaller filter kernels and therefore deeper networks (up to 19 layers for VGG19, compared to 7 for AlexNet), and training the deeper networks using pre-training on shallower versions.
GoogLeNet (Szegedy et al., 2015)	Promoted the idea of stacking the layers in CNNs more creatively, as networks in networks. Inside a relatively standard architecture (called the stem), GoogLeNet contains multiple inception modules, in which multiple different filter sizes are applied to the input and their results concatenated. This multi-scale processing allows the module to extract features at different levels of detail simultaneously. GoogLeNet also popularized the idea of not using fully-connected layers at the end, but rather global average pooling, significantly reducing the number of model parameters.
ResNet (He et al., 2016)	Introduced skip connections, which makes it possible to train much deeper networks. Some features are best constructed in shallow networks, while others require more depth. The skip connections facilitate both at the same time, increasing the network's flexibility when fed input data.
DenseNet (Huang et al., 2016)	Builds on the ideas of ResNet, but instead of adding the activations produced by one layer to later layers, they are simply concatenated together. The original inputs in addition to the activations from previous layers are therefore kept at each layer, preserving some kind of global state. This encourages feature reuse and lowers the number of parameters for a given depth. DenseNets are therefore particularly well-suited for smaller data sets.
GANs (Goodfellow et al., 2014)	A generative adversarial network (GAN) consists of two neural networks pitted against each other. The generative network G is tasked with creating samples that the discriminative network D is supposed to classify as coming from the generative network or the training data. The networks are trained simultaneously, where G

	aims to maximize the probability that D makes a mistake while D aims for high classification accuracy.
U-net (Ronneberger et al., 2015)	A very popular and successful network for segmentation of images. When fed an input image, it is first downsampled through a “traditional” CNN, before being upsampled using transpose convolutions until it reaches its original size. In addition, based on the ideas of ResNet, there are skip connections that concatenates features from the downsampling to the upsampling paths, see Figure 1.

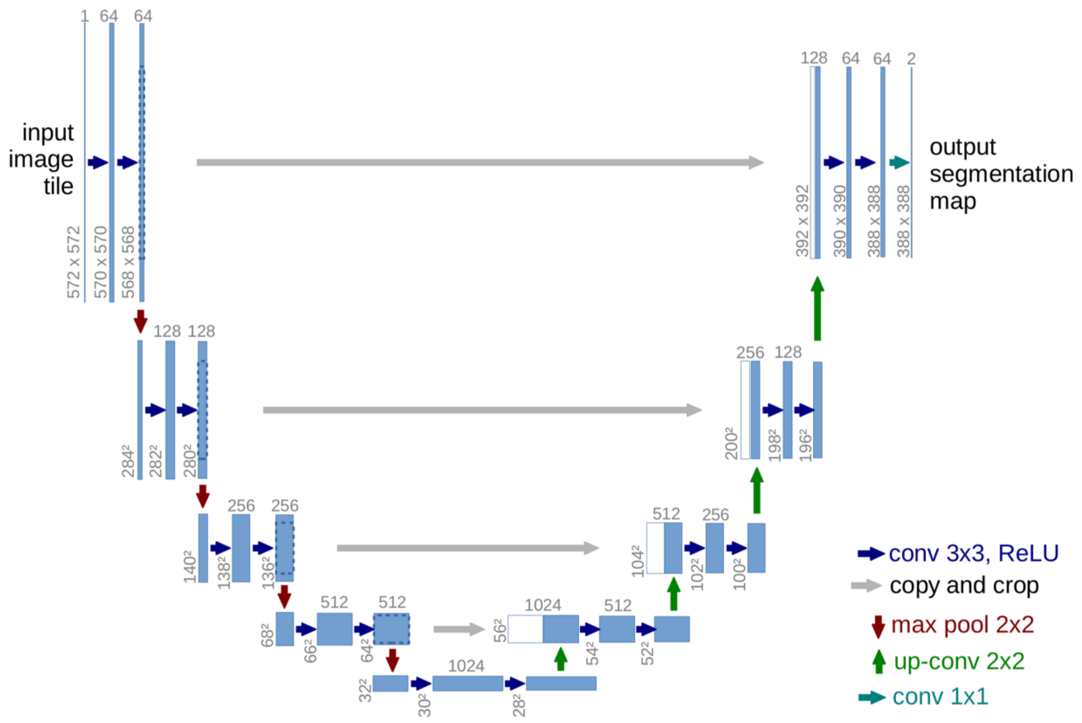

Figure 1. The popular U-Net architecture for segmentation, which has been used in 2D as well as 3D. In every layer of the network convolutions are performed with a number of filters, which are learned through backpropagation. The left part of the network learns a compact representation, while the right part performs upsampling to reconstruct a segmented version of the image. Image from Ronneberger et al. (2015).

4 Survey of 2.5D deep learning methods for segmentation

There have been many attempts at reaping the benefits of 3D segmentation while sidestepping its computational challenges. One class of such methods, referred to as 2.5D methods, are those that modify 2D architectures to incorporate (partial) 3D information.

The first 2.5D approaches used an ensemble of 2D CNNs applied to sagittal, coronal and axial views (Prasoon et al., 2013). Similarly, multiple 2D views may be fed to a 2D CNN as different channels of the input (Roth et al., 2014). Alternatively, the input channels could represent adjacent slices in the 3D data, so that the first layer effectively performs 3D processing of a fixed number of slices (potentially all) but subsequent layers act as a conventional 2D CNN (Mehta & Sivaswamy, 2017).

Alternatively, if a 3D volume is considered as a stack of adjacent 2D slices, then there are two natural ways of ordering them, e.g. up and down if considering axial slices. This point of view has motivated 2.5D approaches based on recurrent neural networks (RNNs), where the 2D slices are processed as an ordered sequence, each of which is segmented using e.g. 2D U-net-like architectures (Chen et al., 2016; Poudel et al., 2016).

A third type of 2.5D methods first use a 2D model to coarsely segment and propose a bounding box that is then segmented with all classes using a 3D method within the smaller ROI. This approach was used by the winning team of the AAPM 2017 challenge on Auto-segmentation for thoracic radiation treatment planning (Yang et al., 2018).

5 Survey of (true) 3D deep learning methods

Compared to 2D and 2.5D CNNs, 3D CNNs try to utilize the full 3D information to improve the segmentation accuracy, at the cost of a higher memory consumption and a longer training time. While 2D CNNs can utilize many layers and many filters in every convolutional layer (e.g. 64 – 256 filters), for 3D CNNs the number of layers and filters must be much smaller, due to the fact that a 256 x 256 x 256 volume requires 256 times more memory compared to a 256 x 256 image.

Dolz et al. (2018) employed a 3D FCNN with concatenation of features at multiple scales to segment subcortical structures form T1 images, and achieved state-of-the-art results on IBSR data, see Figure 2. They also trained their model on 1,112 volumes segmented with FreeSurfer (Fischl, 2012) and showed that they can attain high segmentation performance in a fraction of the time required by atlas-based methods, see Figure 3.

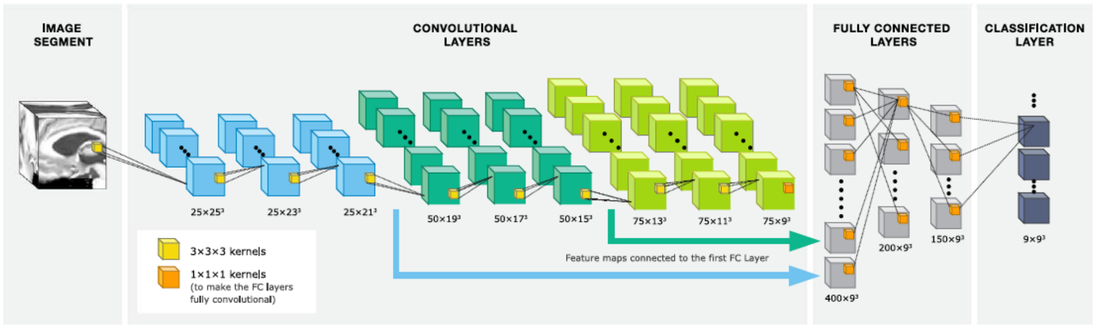

Figure 2. The network presented in Dolz et al. (2018) consists of 9 convolutional layers of size 3x3x3 followed by 3 convolutional layers of size 1x1x1. These latter layers perform similarly to fully connected layers in traditional CNNs, but have the advantage of classifying the whole input patch instead of just the central voxel. Activation maps from the third, sixth and ninth convolutional layers are concatenated to provide features at multiple scales for the final classification.

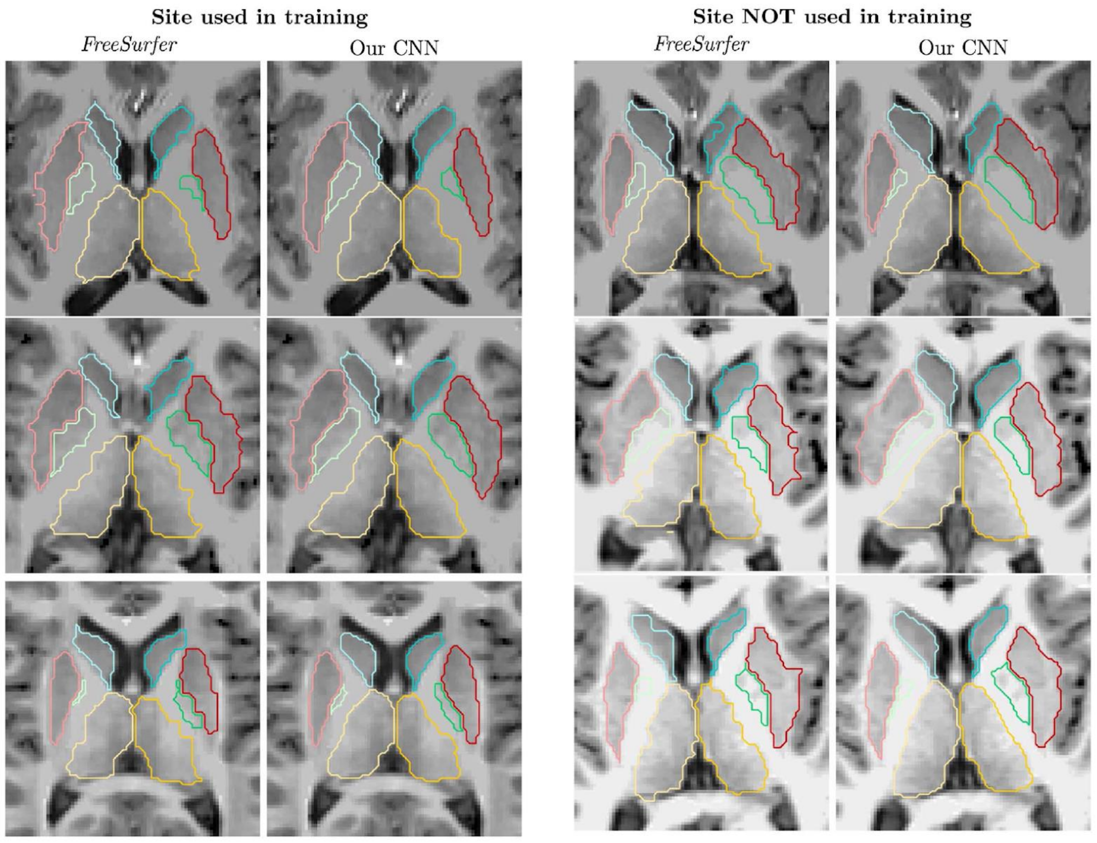

Figure 3. Example segmentations from Dolz et al. (2018). Their results compare favorably with FreeSurfer both when running on subjects from sites used in training or from new sites.

Wachinger et al. (2018) developed DeepNAT, a 3D segmentation architecture based on two stacked sliding-window classifier networks, which they trained on the MICCAI Multi-Atlas Labeling challenge data in order to segment 25 different brain structures from T1 images, see Figures 4, 5 and 6. Notable in their implementation is the use of a first CNN to discriminate between foreground and background, and a second CNN which classifies the foreground voxels into the 25 categories. Other features of note are the incorporation of a novel set of spectral brain coordinates as additional information for the network, the use of multi-task training which provides labels for several voxels for every input patch, and the use of a fully-connected conditional random field to provide the final segmentation results. With such an architecture they managed to match the segmentation performance of PICSL, the winning algorithm in the MICCAI challenge, while shortening segmentation times by an order of magnitude.

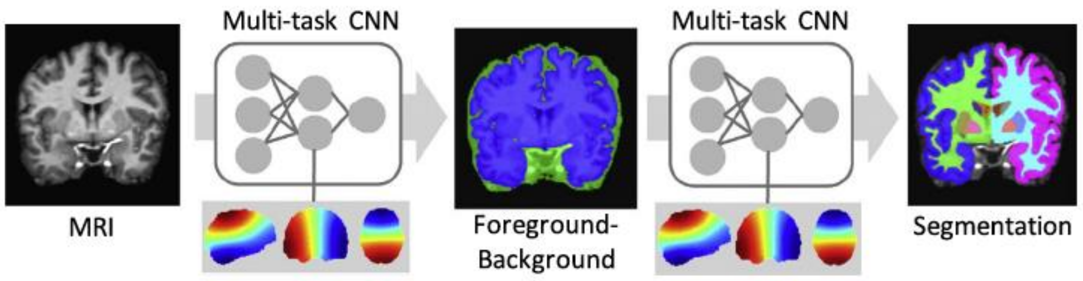

Figure 4. Wachinger et al. (2018) use a first CNN to separate background from foreground, and a second one to segment the foreground voxels into 25 categories.

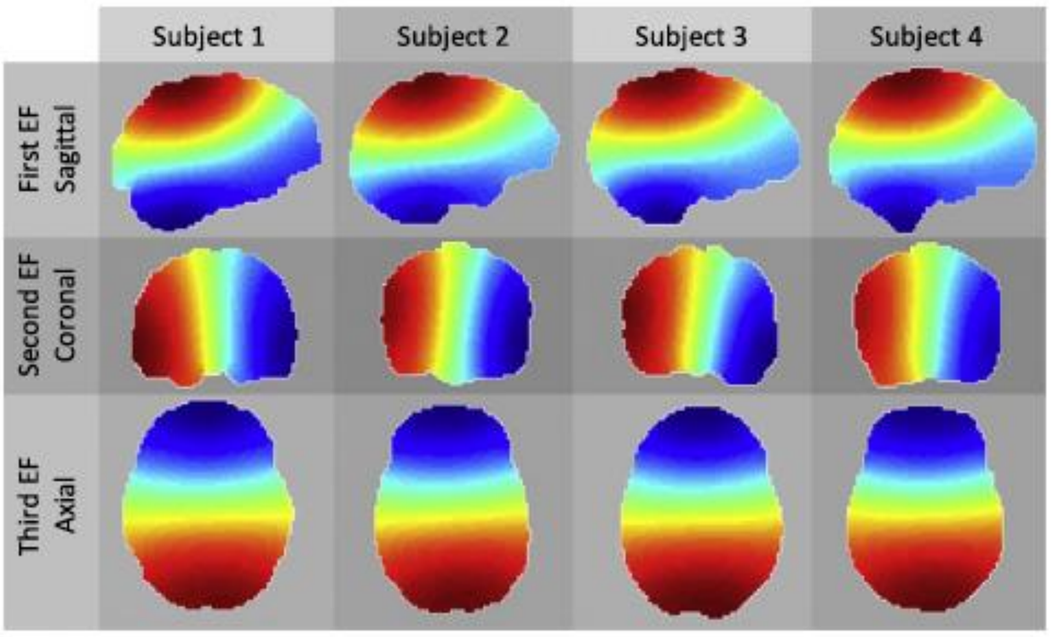

Figure 5. The spectral brain coordinates used in Wachinger et al. (2018) are obtained from the first three eigenvectors of the graph Laplacian of the brain. These generate a mapping within the brain which roughly corresponds with up-down, left-right and front-back positions.

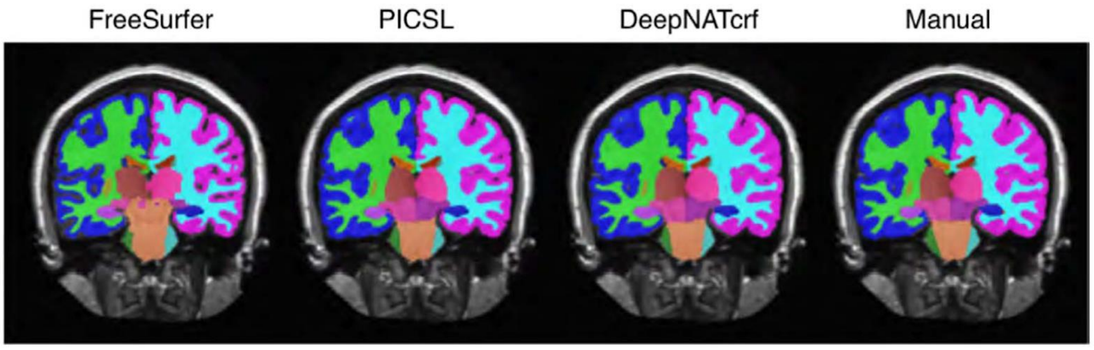

Figure 6. The segmentation results produced by Wachinger et al (2018) closely match the manual segmentation, with better performance than FreeSurfer and similar to PICSL (in a fraction of the time).

Chen et al. (2018) implemented VoxResNet, a voxelwise residual network for segmenting white matter, gray matter and cerebrospinal fluid, see Figures 7 and 8. Their network takes multimodal input from T1, T1-IR and T2-FLAIR images simultaneously, which are shown to give better results than using any single one of those modalities. In addition, they employ auxiliary classifiers at various intermediate network stages and combine their predictions with a final classifier to obtain the segmentation. A further improvement was made in the form of an additional VoxResNet network that refines the predictions of the original one. Their approach was on the MICCAI MRBrainS segmentation challenge data, and achieved state-of-the-art performance.

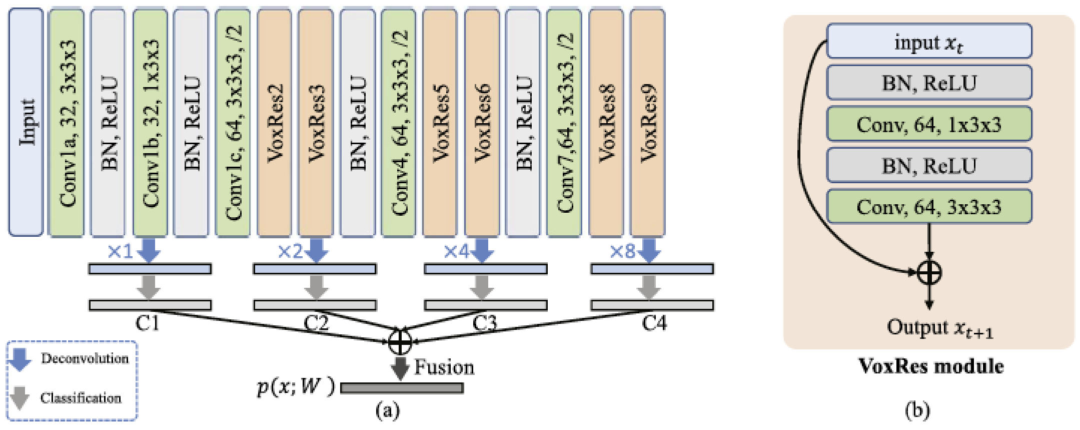

Figure 7. The residual modules employed in the architecture proposed by Chen et al. (2018) make it possible to extend the depth of the network without negatively affecting the backpropagation of gradients. The auxiliary classifiers act on features of different scales, making each of them better suited for identifying specific brain structures.

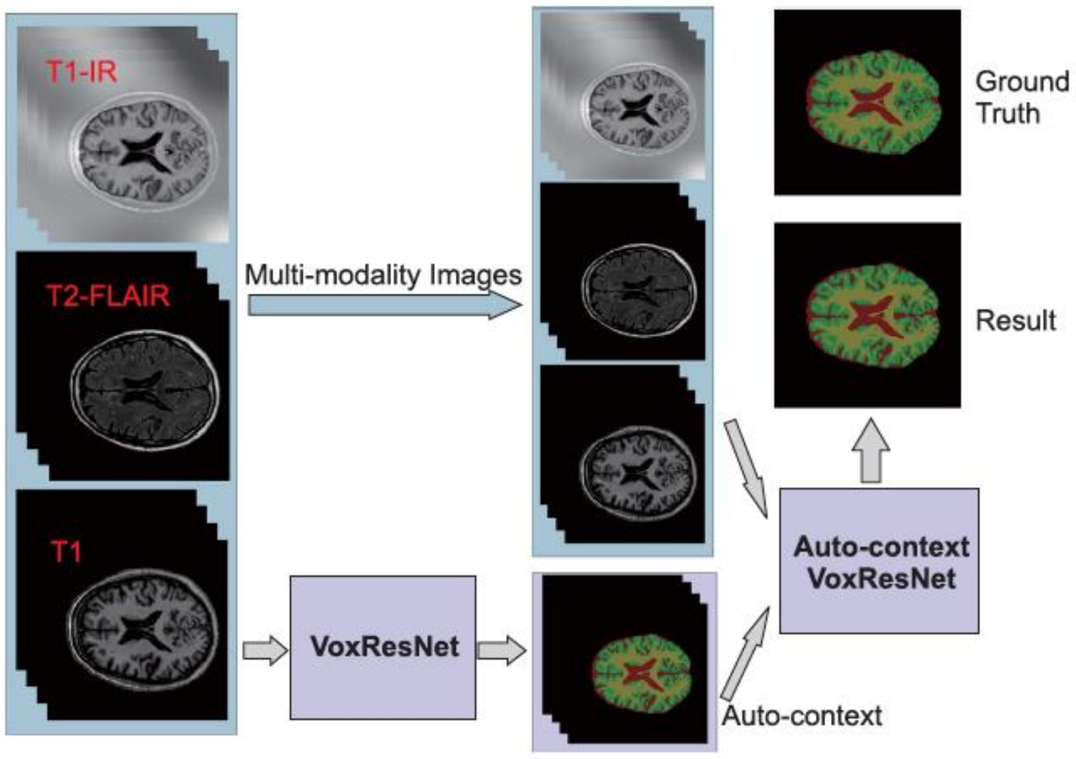

Figure 8. Chen et al. (2018) employ an auto-context VoxResNet which takes the input and output of the original VoxResNet and outputs a refined segmentation.

6 Overview of evaluation metrics

Deep-learning segmentation frameworks rely on the choice of both network architecture and loss function. An essential part of measuring progress in image segmentation is evaluating the quality of the segmentation compared to a gold standard. However, the choice of primary evaluation metric is not trivial and there are many metrics that are used in literature for segmentation to different extents (Taha & Hanbury, 2015). They can be grouped into six categories: spatial overlap based, volume based, pair counting based, information theoretic based, probabilistic based and spatial distance based (Taha & Hanbury, 2017). Depending on the purpose and applications, there might be preference for one category over another. It is not uncommon to have multiple metrics to evaluate the segmentation quality from different perspectives as in challenges (such as BratS and LiTS). It is also common to combine several measures (Udupa et al., 2006, Cárdenes et al., 2009). In addition, segmentation algorithms can be evaluated according to efficiency, i.e. practical use of the algorithm such as segmentation time (Fenster & Chiu 2005). However, here we focus on the performance evaluation. The most common classical metrics are Dice score, Jaccard index, Hausdorff distance, precision and recall.

Dice score

The most commonly used metric for image segmentation is the Dice score, which in different forms can be used both as an evaluation metric and as a loss function. It is a spatial overlap based metric. Given the segmented image, P, and the reference (golden standard) image, R, the Dice score, first defined in (Dice, 1945), can be written as

$$Dice = \frac{2 \cdot |P \cap R|}{|P| + |R|},$$

where |P| and|R| are the cardinality of the sets.

Jaccard index

The Jaccard index, also known as Intersection over Union (IoU), is defined as the ratio of the intersection between two sets and the union of the two sets, i.e.

$$JAC = \frac{|P \cap R|}{|P \cup R|}$$

The Dice score and the Jaccard score are similar, such that

$$DICE = \frac{2JAC}{1 + JAC}$$

Hausdorff distance
Hausdorff distance is a surface-based metric. It is defined as

$$HD(A,B) = max(h(A,B),\ h(B,A)),$$

where

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

where ||a-b|| is some norm, e.g. Euclidean distance.

However, the Hausdorff distance is sensitive to outliers. Average Hausdorff distance and quantile Hausdorff distance have therefore been proposed as alternatives (Taha et al., 2015).

For segmentation in medical images, accuracy is not commonly used alone due to class-imbalance in the dataset and the fact that it does not reflect the number of false positives. Instead, precision and recall tell the proportion of correct positive predictions and actual positive identifications.

Precision
The precision, also known as positive predictive value, quantifies the ability to correctly identify a positive occurrence of a class. It is defined as

$$Precision = \frac{TP}{(TP+FP)}$$

where TP is the number of true positive examples, i.e. the voxels that are classified as positive examples when the ground truth label is positive, and FP is the number of false positive examples, i.e. the voxels classified as positive examples when the ground truth label is not.

Recall
The specificity, also known as recall or true positive rate, quantifies the accuracy in the classification of the background tissue, and it is defined as

$$Recall = \frac{TP}{(TP+FP)}$$

where FN are false negative, i.e. the voxels belong to a certain class in the ground truth but are not classified in the segmentation.

Evaluation of both precision and recall is useful to evaluate the effectiveness of the segmentation performance. However, improving precision typically reduces recall and vice versa. The combination of recall and precision is common to use, e.g. F_1-measure and it is mathematically equivalent to the Dice score.

7 Overview of loss functions

The loss function used for training of the (deep) network should be chosen both with respect to the task at hand and the evaluation metric. The most commonly used loss functions are cross-entropy and Dice loss.

Cross-entropy

In image segmentation, one of the most common loss functions is voxel-wise cross-entropy, which arises when the likelihood is maximized or, equivalently, the negative log-likelihood is minimized. The voxels are assumed to be independent and the class predictions are compared to a one-hot encoded target vector. The loss function is defined as

$$Cross\ entropy \ = \ -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{M} t_{n,c}log(p_{n,c})$$

where we assume that the classification is done into M classes, $t_{n,c}$ is the one-hot encoding of the ground truth and $p_{n,c}$ is the class prediction for voxel n. Then an average over all voxels is computed, ensuring equal training to each voxel.

Dice score

The Dice score can be used directly as an evaluation metric but since it is not differentiable everywhere it cannot be directly used as a loss function for deep learning. However, continuous versions of the Dice score, which can be used as loss functions, have been suggested. For example, a 2-class variant of the Dice loss suggested in (Milletari et al. 2016) can be expressed as

$$DL_2 = 1 - \frac{\sum_{n=1}^{N} p_n r_n + \varepsilon}{\sum_{n=1}^{N} p_n + r_n + \varepsilon} - \frac{\sum_{n=1}^{N}(1-p_n)(1-r_n) + \varepsilon}{\sum_{n=1}^{N} 2 - p_n - r_n + \varepsilon}$$

where pn are the image elements belonging to P and rn the image elements belonging to R. The epsilon term is used to ensure loss function stability.

Cross-entropy is one of the most commonly used loss functions, especially in image segmentation. However, in medical images healthy tissue will generally dominate the unhealthy tissue creating an imbalance in data, which can be difficult for the cross-entropy loss function. There are different ways to deal with it, such as adding weightings to different classes. (Han 2017, Christ et al., 2017, Grovik et al., 2019). Differentiable Dice loss was proposed to replace conventional cross-entropy for class imbalance without assigning weightings to different classes and showed better results (Milletari et al., 2016, Sudre et al., 2017). It has since then been adopted by many (Zhou et al., 2016, Wang et al., 2017, Drozdzal et al., 2017, Chlebus et al., 2018, Roth et al., 2018).

There are also other designed loss functions for medical image segmentation e.g. combination of sensitivity and specificity (Brosch et al., 2016), new Jaccard loss (Cai et al., 2017), and a modified Z-loss (Mortazi et al., 2017). Weight penalties, i.e. L1/L2 regularization, are commonly included in the loss function to avoid over-fitting (Akkus et al., 2017, Gibson et al., 2018). Ensemble models use a combination of different loss functions, e.g. cross-entropy for DeepMedic, IoU loss and Dice for FCNs, cross-entropy for U-net (Kamnitsas et al., 2017). There are different network architectures that use auxiliary losses in addition to the voxel-wise cross-entropy (Dou et al., 2017, Grewal et al., 2018).

8 Overview of deep learning frameworks

A deep learning framework is an interface, library or a tool which allows us to build deep learning models more easily and quickly, without getting into the details of underlying algorithms. They provide a clear and concise way for defining models using a collection of pre-built and optimized components. The five most popular deep learning frameworks in 2018 are Tensorflow, Keras, PyTorch, Caffe and Theano, with Tensorflow being by far the most popular[1]. All frameworks are open source, support Python and contain the most popular deep learning network types. Tensorflow, Pytorch and Caffe also support C++. Theano development has officially stopped in 2017 and is therefore not further discussed. In 2019, Tensorflow is still the most in demand framework and fastest growing[2]. PyTorch is growing rapidly too. Finally, Keras and fastai, which are high level APIs for Tensorflow and PyTorch respectively, are also growing. Virtually all frameworks include support for doing the demanding calculations on one or several (Nvidia) graphics cards.

As a small example, the following Keras code defines a single 2D convolutional layer for k filters of size 7 x 7, with a ReLU activation function, and it is very easy to concatenate many convolutional layers to obtain a deep network. Tensorflow automatically calculates the required derivatives used for updating the weights through back propagation. Functions for 3D convolution are also available.

```
def ConvLayer(self, x, k):

x = Conv2D(filters=k, kernel_size=7, strides=1, padding='valid')(x)
x = Activation('relu')(x)
return x
```

[1] https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a
[2] https://towardsdatascience.com/which-deep-learning-framework-is-growing-fastest-3f77f14aa318

9 Hardware differences between 2D, 2.5D & 3D

Deep learning, and image processing in general, is a very computationally intensive task, which is not ideally suited to the general purpose computing architectures of CPUs. Instead, the massively parallel architectures of graphics cards (GPUs) prove to be much more adequate for performing the necessary simple, yet numerous calculations, since a single modern GPU can have several thousand processor cores. GPUs are also relatively cheap, thanks to the gaming industry. In medical imaging, GPUs have been used for a long time to speedup image reconstruction (of CT and MR images) as well as the most common algorithms (image denoising, image segmentation, image registration) (Eklund et al., 2013). GPUs are well suited for accelerating deep neural networks, including CNNs, where both the forward and the backward pass (backpropagation) involve many matrix operations (convolution can be written as a (sparse) matrix operation).

One downside of GPUs, however, is the need to rely on their dedicated RAM in order to store both data and parameters of the trained model. Despite the fact that both processing and storage capacity of GPUs are steadily increasing, typical graphics cards provide between 3 and 10 times less RAM than can be made available to the CPU. The capacity to expand the GPU memory of a workstation is also much more limited, with multiple GPUs normally being unable to share resources without dedicated hardware for that purpose. Companies like Nvidia provide specialized hardware which can provide more GPU memory, by combining the memory of several graphics cards. For example, the Nvidia DGX station contains four Tesla V100 graphics cards with 32 GB of memory each, which can be used together to provide 128 GB of graphics memory (however, at a lower memory bandwidth compared to using a single graphics card).

Two different aspects of a given deep learning application determine the amount of RAM necessary to train the model:

- The number of model parameters, which affect the size or representational capacity of the model. CNNs provide significant advantages in this regard, since the number of model parameters depends not on the input images, but on the chosen size of the filters used. This allows for deeper networks with higher level features. The convolutional layers require few parameters to store, e.g. 100 filters of size 3 x 3 only require storing 900 parameters. Nevertheless, it is not uncommon to combine convolutional and fully connected layers (especially at the final layer), which reestablishes the dependence on the size of the input images. Storing parameters for a fully connected layer can require substantial memory (e.g. a single fully connected layer with 10 000 input and output nodes requires about 380 MB of memory in 32 bit floats). Model parameters are also increased by the use of advanced methods such as early/late fusion, multiple pathways, and feature map concatenation.

- The size of the input images, which in addition to affecting the model parameters has a direct impact of various intermediate results (e.g. feature maps, gradients) that have to be stored in the GPU during forward and backward passes in training/testing. This latter effect is not helped by the use of convolutional layers, which generate one full output image for every filter at every layer, and it is not uncommon to have 64 - 128 filters in a single convolutional layer (at least for 2D CNNs). Since all training data cannot be stored simultaneously in GPU memory, the user of deep learning frameworks has to define a batch size that defines how many images / volumes to analyze at a time. For 2D images the batch size can range from 4 - 128, while for 3D data it can be difficult to even store a single volume in memory (since also many filter responses need to be stored at the same time). For 3D it is therefore often necessary to first divide a volume into subvolumes that can fit in GPU memory, and here special hardware with large GPU memory (e.g. >= 32 GB) can be very beneficial (for example, the winner of the 2018 BraTS challenge used a 32 GB graphics card). Another potential solution is to use 16 bit floats, instead of 32 bit floats, to reduce the memory usage by a factor 2 (at the cost of lower precision).

Taking into account these considerations we are better equipped to appreciate the effect that the dimensionality of the input data has on the hardware requirements. While it is rather easy to train a 2D CNN for segmentation, training a 3D CNN results in 3D convolutions instead of 2D convolutions, which require substantially more calculations, and 3D CNNs also require more memory. In general, 3D CNNs can therefore not be as deep as 2D CNNs, but Myronenko (2018) showed that the depth of a 3D network did not have a direct effect on the segmentation performance, while increasing the width of the network (i.e. the number of filters in each layer) improved performance.

10 Overview of open datasets available for pre-training

Training CNNs for segmentation requires high quality data with (ground truth) annotations provided by an expert. Depending on the type and degree of data augmentation, see section 13 about data augmentation, the required number of annotated examples can vary substantially. Segmentation challenges are rather popular and connected to different medical imaging conferences. Each challenge normally provides training data with ground truth, and the test data are hidden from the researchers. An overview of current and previous challenges is available[3], it covers spine segmentation, breast cancer metastases, kidney tumor segmentation, liver segmentation, retina vessel segmentation and many more. We will here discuss some of the open datasets in more detail.

Brain tumors

For brain tumor segmentation, the most important open dataset is BraTS, brain tumor segmentation, available through an annual challenge (Bakas et al 2019). The Brats 2018 dataset[4] contains 285 multimodal cases, 210 high-grade gliomas and 75 low-grade gliomas, see Figure 9 for an example. Each case consists of four aligned MRI modality volumes: native (T1), post-contrast T1-weighted, T2-weighted and T2 Fluid Attenuated Inversion Recovery (FLAIR), acquired with different clinical protocols and various scanners from 19 institutions. All the imaging datasets have been segmented manually, by one to four raters, following the same annotation protocol, and their annotations were approved by experienced neuro-radiologists. Annotations comprise the GD-enhancing tumor (ET — label 4), the peritumoral edema (ED — label 2), and the necrotic and non-enhancing tumor core (NCR/NET — label 1), as described in (Menze et al. 2015).

[3] https://grand-challenge.org/challenges/
[4] https://www.med.upenn.edu/sbia/brats2018/data.html

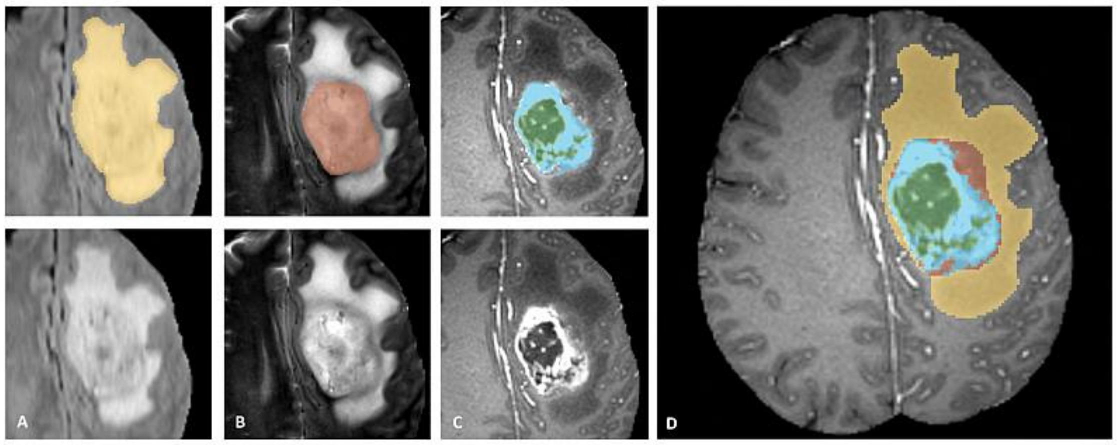

Figure 9. Manual segmentations provided for the BRaTS dataset. Image from (Menze et al., 2015). Shown are image patches with the tumor sub-regions that are annotated in the different modalities (top left) and the final labels for the whole dataset (right). The image patches show from left to right: the whole tumor (yellow) visible in T2-FLAIR (Fig.A), the tumor core (red) visible in T2 (Fig.B), the enhancing tumor structures (light blue) visible in T1Gd, surrounding the cystic/necrotic components of the core (green) (Fig. C). The segmentations are combined to generate the final labels of the tumor sub-regions (Fig.D): edema (yellow), non-enhancing solid core (red), necrotic/cystic core (green), enhancing core (blue).

Liver tumors

The LiTS 2017 dataset[5] contains 201 CT scans (131 for training, 70 for testing) from 7 different hospitals and research institutions along with ground truth segmentations for liver tumors and liver (Bilic et al, 2019), see Figure 10. The training set contains a mean of 6.93 tumors per subject, while the test set contains a mean of 9.41 tumors per subject. The image resolution ranges from 0.56 mm to 1.0 mm in axial and 0.45 mm to 6.0 mm in z direction. The number of slices in z ranges from 42 to 1026. Some images contain imaging artifacts (e.g. metal artefacts), which are present in real life clinical data.

[5] https://lits-challenge.com

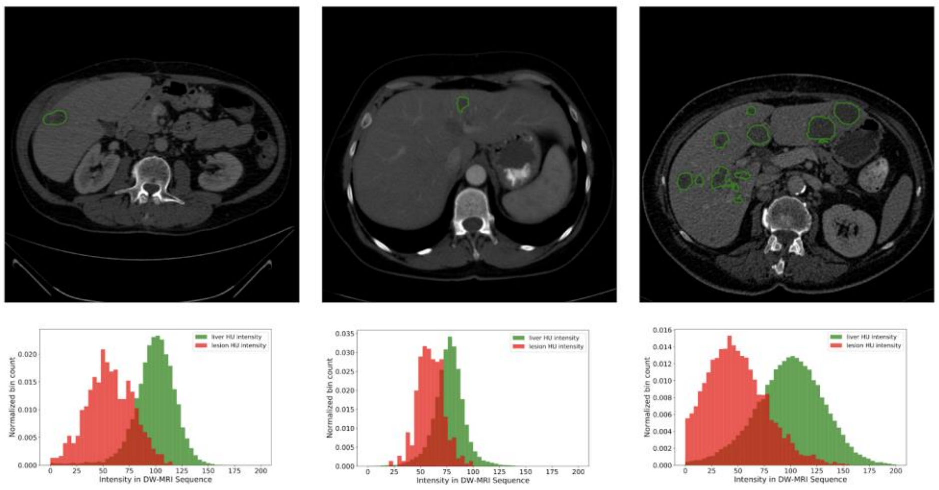

Figure 10. Example CT scans, and annotations of liver tumors, from the LiTS dataset. Image from (Bilic et al., 2019).

Vertebrae

Segmentation of the vertebrae in the spine is a common problem, and several open datasets are available through SpineWeb[6]. A total of 16 different datasets are available, covering X-Ray, CT as well as MRI, of both healthy controls and subjects with fractured vertebrae, see Table 2 for an overview.

Table 2, overview of the 16 datasets available in SpineWeb. Most of the datasets contain manual segmentations that can be used for training.

Dataset	Data
1	30 pairs of spinal CT and MR from lumber
2	Spine CT from 10 subjects
3	Spine-focused CT scans of 125 patients
4	Spine CT scans of 5 patients
5	CT lumbar spine images of 10 controls
6	Multi-modality MRI of 8 patients
7	T2 MRI of 15 patients
8	CT lumbar spine images of 4 patients
9	Dual energy x-ray absorptimetry
10	MR + CT from 20 subjects
11	T1, T2 MRI from 17 patients
12	Dual energy x-ray absorptimetry, 30 patients
13	CT lumbar spine images of 25 patients

[6] http://spineweb.digitalimaginggroup.ca

14	Multi-modality MRI of 24 patients
15	CT lumbar spine, 10 patients
16	Spinal anterior-posterior x-ray, 609 patients

11 Overview of preprocessing steps

11.1 MRI

Preprocessing steps for MRI are generally performed to remove artefacts and standardize the images before further processing. Standardization can be done both with respect to image intensities and the geometric space (coordinate system). Some preprocessing steps are the same for different MR sequences while others are sequence specific.

Preprocessing could be divided into four parts: noise reduction, inhomogeneity correction, registration and intensity normalization. Depending on the anatomical location, specific types of preprocessing may be applied, such as skull-stripping for brain images. Resampling may also be necessary when there are multiple images with different resolution (Akkus et al., 2017, Despotović et al., 2015, Lundervold et al., 2018, Martí-Bonmatí and Alberich-Bayarri 2017).

Noise reduction

Recent denoising methods exploit the sparseness or self-similarity in medical images. Sparsity-based methods approximate the base of signals by making the assumption of the sparsity of noise and signal in low-dimensionality space (Aharon et al., 2006, Bao et al., 2008, Patel et al., 2011). Self-similarity methods use the natural pattern of the images for noise reduction. A popular example is nonlocal means, which uses the statistics of similar patches in the image for denoising (Buades et al., 2005, Manjón et al., 2012, Manjón et al., 2015).

Inhomogeneity correction

Inhomogeneity in MR is caused by variations in the coils' sensitivities and in the main magnetic field of the scanner. It causes objects with the same physical properties to have different signal intensities (Sled et al., 1998). Inhomogeneity correction mainly includes prospective and retrospective approaches. Prospective correction requires specific hardware or sequences, e.g. 3D magnetization-prepared rapid acquisition gradient echo (MP2RAGE) sequences (Marques et al., 2010). Retrospective inhomogeneity correction models the bias field with the image features. There are methods that estimate bias field with the use of a set of low-frequency basic functions (Vovk et al., 2004, Manjón et al., 2007). An algorithm combines N3 bias field correction (Sled et al., 1998) and fuzzy-C-means (Lin et al., 2011). However, N4 bias field correction, the successor of N3, is the most common bias field correction algorithm for deep learning based image segmentation (Akkus et al., 2017, Milletari 2016, Menze 2015). N4 corrects inhomogeneity using B-spline fitting (Tustison et al., 2010).

Intensity standardization

Structural MR image intensity is dependent on the scanner and does not have a quantifiable interpretation. There are different histogram matching methods proposed to standardize the MR image intensities. (Nyul et al., 2000, Hellier, 2003) A popular algorithm proposed by Nyul et al. is to use piecewise linear mapping to normalize the histogram intensity of MR images. It matches the input image histogram onto a standard histogram landmarks. Z-score normalization is another popular intensity standardization for feature scaling. (Pereira et al., 2016, Kamnitsas et al., 2017, Akkus et al., 2017) showed that the intensity normalization proposed by Nyul et al. combined with z-score normalization has a positive impact on the segmentation results using CNN with multi-site and multi-scanner data.

Skull-stripping

For brain MR images, it is common to remove non-brain tissue, i.e. skull and background such that the preprocessing steps and deep learning models focus on intracranial tissues. Some common methods for skull-striping are BET (Smith, 2002) (or its modified version OptiBET (Lutkenhoff et al., 2014)) , BEaST (Eskildsen, 2012), Robex (Iglesias et al., 2011) and SPECTRE (Carass, 2011). One also proposed a skull stripping method that is pathology-insensitive (Roy, 2017). Latest research also include the use of deep learning for brain extraction. (Kleesiek, 2016)

Given the complex modelling of deep learning, some of the preprocessing steps became less important for segmentation. Aligning volumes to a standard geometric space is computationally expensive and it reduces the generalization power of deep learning to unprocessed data, but it can potentially also make the learning easier. Dolz et al. (2018) suggested a preprocessing step with only volume-wise intensity normalization, bias field correction and skull-stripping. Kamnitsas et al. compared the preprocessing steps of having (1) only z-score normalization, (2) bias field correction followed by (1), and (3) bias field correction, followed by piece-wise linear normalization proposed by Nyul et al., and (1). The preprocessing steps were inconclusive but instead the ensembling of all three were used. It is noted that the data used by Kamnitsas et al. is from a co-registered and skull-stripped public dataset. Simply put, many deep learning based MRI segmentations perform basic preprocessing steps with image registration, inhomogeneity correction, skull-stripping (for brain images) and intensity standardization. Latest research also shows the use of deep learning for the above steps.

11.2 CT

Unlike MRI, Hounsfield Unit (HU) is a standard unit for tissues radiosensitivity in CT. The intensity values of the CT images are represented by HU. The preprocessing of CT is similar to that of MR, yet in general there are less preprocessing steps in CT. Depending on the tissue type and region-of-interest (ROI), preprocessing in CT usually includes choosing the window of the HU number. For instance, HU of [-100, 400] was chosen for liver and lesion and to exclude irrelevant objects (Christ et al., 2016, Christ et al., 2017) or [-200,250] in (Li et al., 2016, Li et al., 2018)

Z-score standardization (Li et al., 2015, Thong et al., 2018) and histogram equalization (Christ et al., 2016, Men 2017) are common for intensity standardization in preprocessing.
Resampling may also be involved in some preprocessing pipelines too, e.g. for coarse resolution. (Li et al., 2016, Han et al., 2017, Li et al., 2018)

Anisotropic smoothing filtering is used in (Mortazi et al. 2017). Gaussian smoothing is also used for de-nosing (Li et al., 2015, Cha et al., 2016). In Cha et al., 2016, anisotropic diffusion, gradient filters, and the rank transform of the gradient magnitude are applied in preprocessing steps as well.

Cropping is another common technique in preprocessing steps in order to focus on ROI and increase the GPU memory efficiency. (Skourt et al., 2018) Apart from cropping, downsampling is also used to reduce the memory usage. (Li et al., 2015, Roth et al., 2018)

11.3 Registration

Registration is required to allow intra-subject multimodal image segmentation or motion correction for spatial alignment. As part of registration, images such as CT and MR are commonly resampled into isotropic resolution. Linear or spline-based interpolation have been popular to align the multi-modal resolutions (Thevenaz et al., 2000). Super-resolution has also been proposed for this matter and showed promising results (Isaac & Kulkarni, 2015).

Rigid transformations are usually sufficient for non-deformable objects within the same subject, e.g. brain imaging, while affine or nonrigid transformations are needed for objects that deform, such as lung and liver. Deformable transformation may also be required within the same non-deformable subject for example the brain development in children at different stages. It is also common to standardize the brain images into the Montreal Neurological Institute (MNI) space for healthy subjects. The geometric transformation is usually optimized with the use of a similarity measure such as mean-square difference, correlation coefficient and normalized mutual information. Some typical nonrigid registration are elastic (Shen & Davatzikos, 2002), fluid deformable models (D'Agostino et al., 2002), a linear combination of smooth basis functions (Ashburner and Friston, 1999), free-form deformation IRTK (Rueckert et al., 1999), FNIRT (Andersson et al. 2008), ART (Ardekani et al., 2009), JRD-fluid (Chiang et al., 2007), SyN (Avants et al., 2008) and DARTEL (Ashburner et al., 2007). Latest research also shows the increasing use of deep learning for image registration. (Simonovsky et al., 2016, Yang et al., 2017, Haskins 2019) For segmentation challenges, public datasets are usually registered for multi-modal images. There are thus less registration steps in pre-processing for deep learning based segmentation when challenge datasets are used.

12 Overview of postprocessing steps

After inference of the deep learning network, the segmentation result has to be adapted to match the input image. As deep learning networks often require a specific format of the input image (such as a certain resolution), the results need to be resampled to match the input image. Resampling is done using nearest neighbor in case of a label image, or B-spline interpolation in case of a continuous image. If an ROI of the input image was used, the segmentation result needs to be padded such that it is the same size as the input image.

In some cases to reduce the GPU memory usage the image is divided in sub-images which can be processed separately on the GPU. When this approach is used the sub-image results need to be re-combined together to re-ensemble the original image.

Another post processing step is cleaning up the segmentation result. Noise can be removed using binary opening/closing operations or by applying smoothing. In some cases only the largest element is needed, then connected components analysis is used to find the largest component and the other components are then removed from the segmentation.

13 Overview of data augmentation techniques

Deep learning requires large datasets for training the deep (convolutional) neural networks, but collecting and annotating high quality (medical) data is costly and time consuming. It is therefore common to apply different data augmentation techniques, to increase the amount of training data without collecting and annotating more data. In this section we will cover simple augmentation techniques (e.g. rotation and scaling), intermediate approaches (e.g. applying smooth elastic deformations) as well as advanced approaches (mainly generative adversarial networks, GANs).

13.1 Simple data augmentation techniques

If a CNN has been trained on images where the object of interest is always oriented at the same angle, the network cannot handle cases where the object has been rotated. Similarly, if a CNN has only been trained on images where the object has a certain size, or is located in a specific part of the image, it will not perform well for images where the object has another size or is located in another part of the image. Traditionally, this has been solved by first deriving features that are invariant to rotation and scaling, and then feed these features into a neural network. For example, the log-polar transform can be used to become invariant to scale and rotation (see e.g. Wolberg and Zokai (2000), Pun and Lee (2003)). In deep learning, the problem is instead solved through data augmentation, i.e. by applying random rotations, translations and scalings to the available training data, and using the transformed data as additional training data, see Figure 11. For segmentation problems, the same random transformation is then applied to the (binary) ground truth mask (normally obtained through manual segmentation).

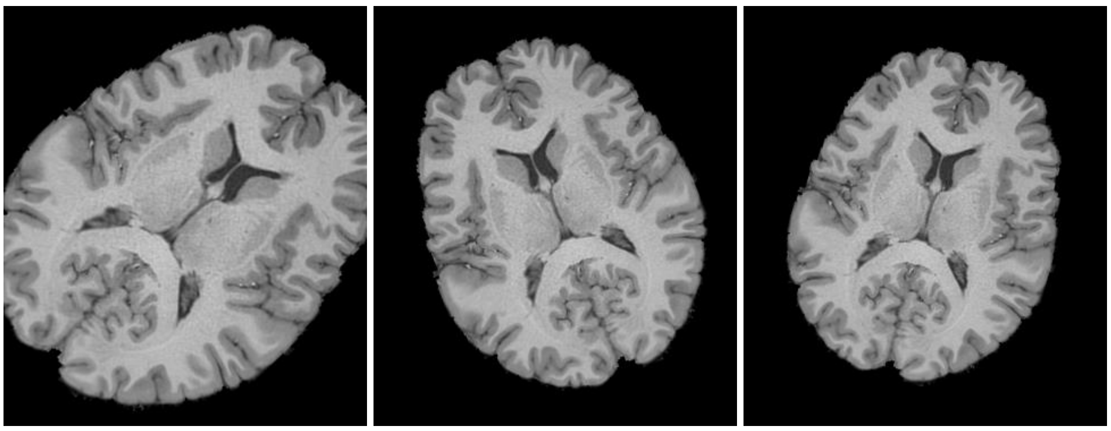
Figure 11. Data augmentation by transforming a brain image with a random rotation and scaling.

There is no limit on the number of augmented images that can be added for the training, but training the CNN will of course take more and more time when the number of training images increase. In the most extreme case, a single annotated image or volume can be used for training a CNN for segmentation (Gaonkar et al., 2018), if the image is augmented in many different ways, see Figure 12. Other simple augmentation techniques include shearing, vertical flipping, horizontal flipping, random cropping and adding random noise to each image.

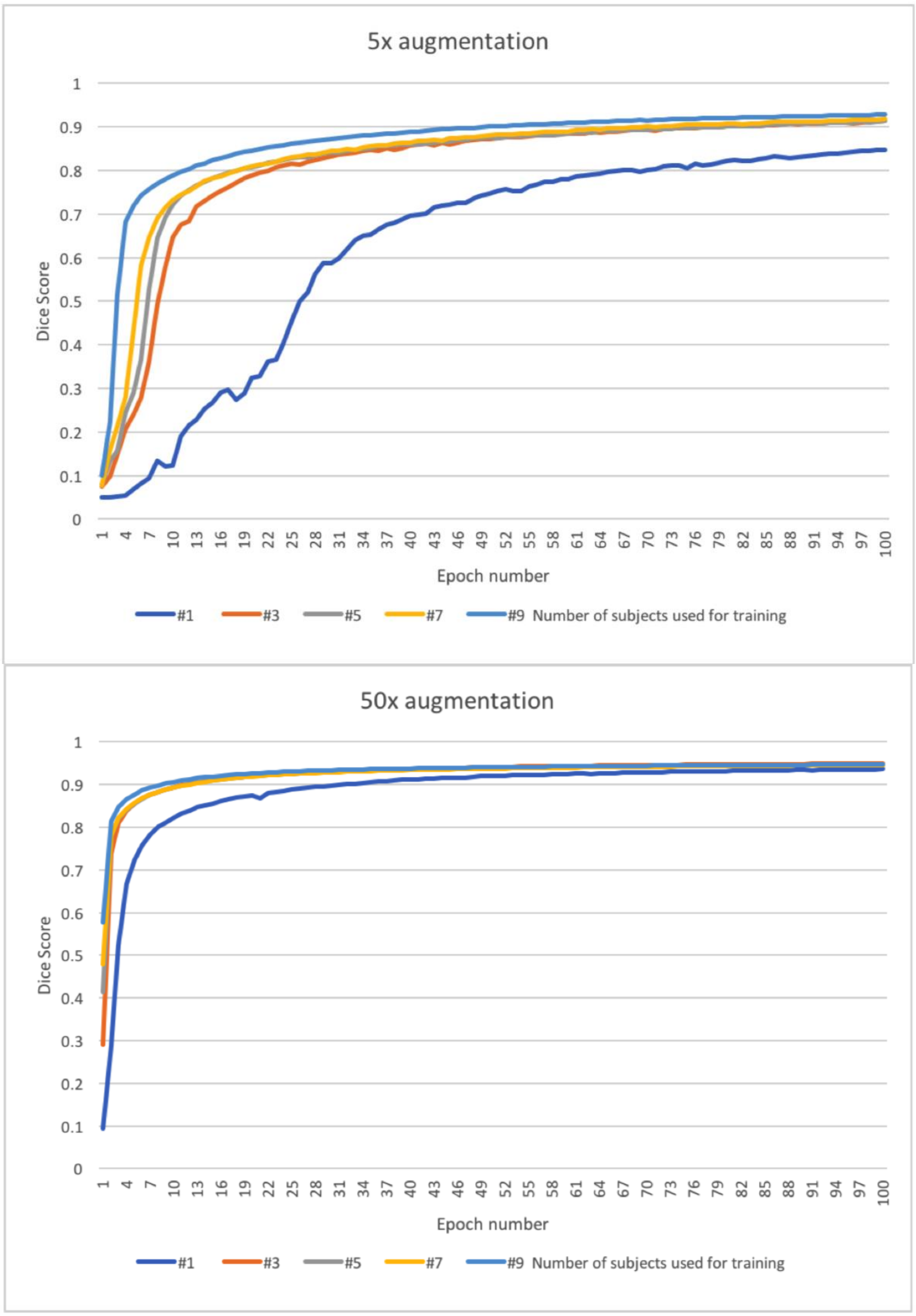

Figure 12. Segmentation performance (Dice score) as function of the degree of data augmentation (number of extra training images with a random rotation, scaling and translation). Plots from Gaonkar et al., 2018. Left: For a low degree of augmentation (5 times), there is a clear difference in the Dice score when using annotated training data from 1 or 9 subjects. Right: For a high degree of augmentation (50 times), the difference between using training data from 1 or 9 subjects is very small.

Several deep learning frameworks provide real-time data augmentation, but the built in support is mainly for 2D and not for 3D. Hence, for 3D segmentation the user has to implement new functions for 3D augmentation.

13.2 Intermediate data augmentation techniques

For image segmentation, further augmentation can be achieved through intermediate techniques. One such technique is to apply (smooth) non-linear deformation fields, also called elastic deformations, as a simple way to simulate data from additional subjects. For the U-Net segmentation architecture (Ronneberger et al., 2015), elastic deformations were listed as a key feature to train the network with few annotated examples. It is important that the deformation fields are smooth, normally obtained by Gaussian smoothing of the x- and the y-displacement, since otherwise the resulting images will not be realistic, see Figures 13, 14, 15 and 16.

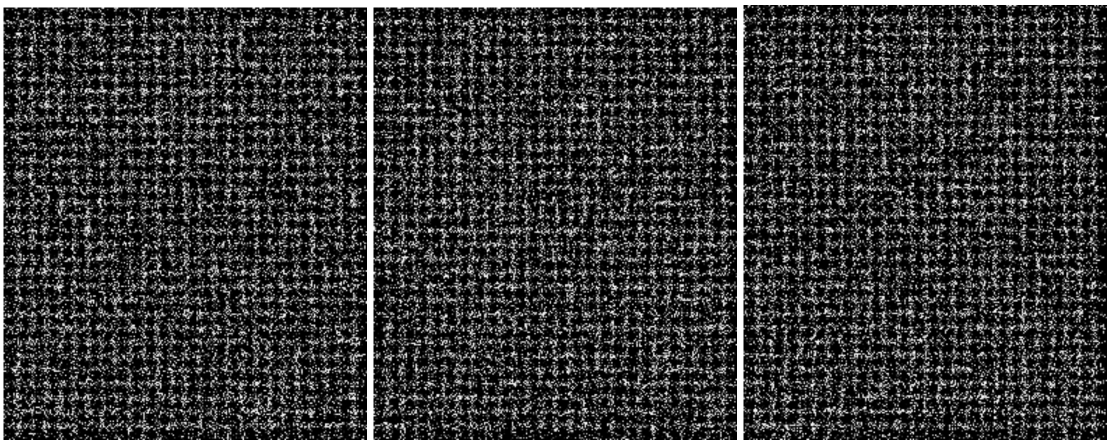
Figure 13. Non-smooth deformation fields used to generate new training images.

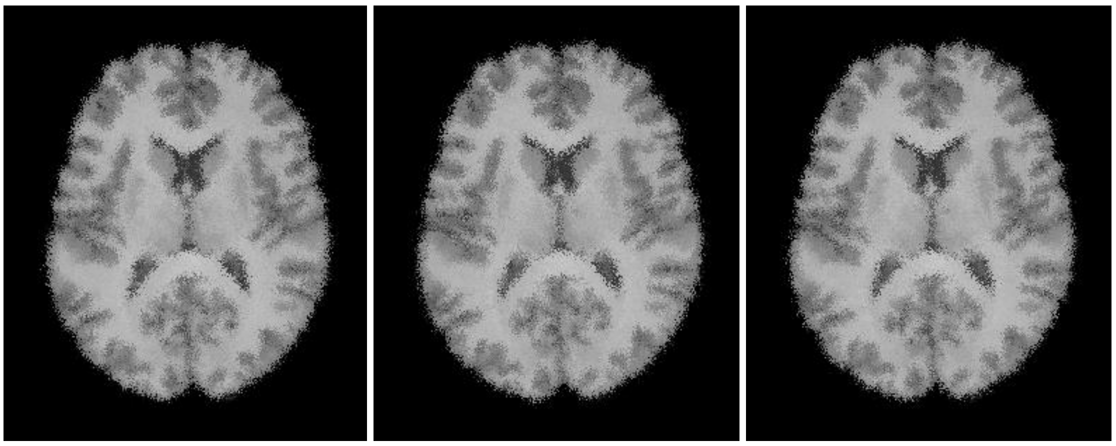
Figure 14. Data augmentation by transforming a brain image with a random displacement field, giving unrealistic augmented brains, since the displacement field is not smooth.

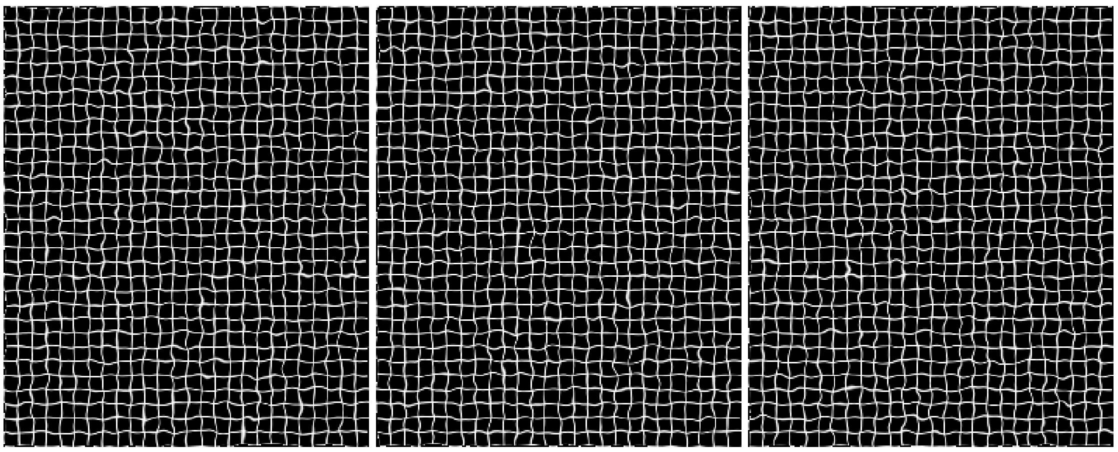
Figure 15. Smooth deformation fields used to generate new training images.

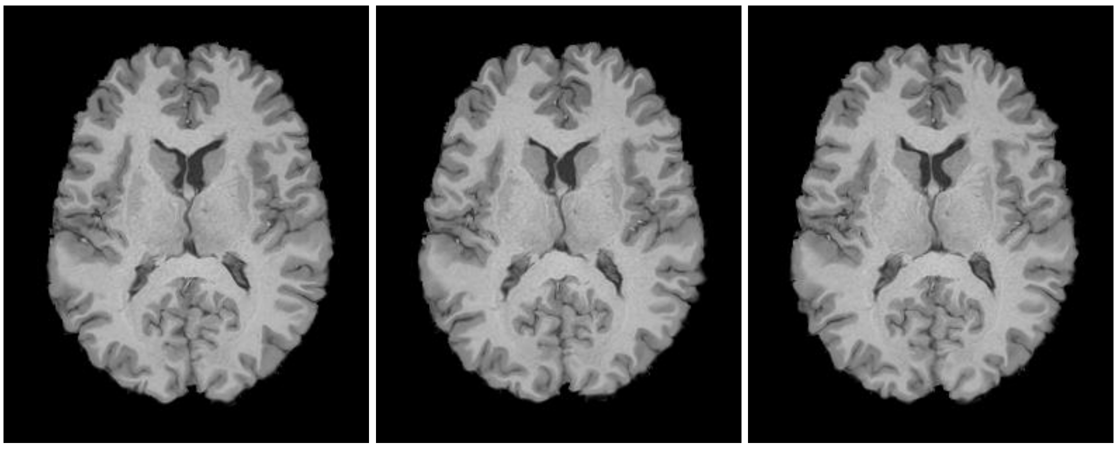
Figure 16. Data augmentation by transforming a brain image with a random smooth displacement field, giving rather realistic augmented brains.

13.3 Advanced data augmentation techniques

The so far mentioned data augmentation techniques do not create completely new images, but only modify existing images through different transformations. Generative adversarial networks (GANs) (Goodfellow et al., 2014) are a rather new innovation, which can learn the mapping between a noise vector and an image, or learn to transform one type of image (e.g. a T1 weighted MR image) to another type of image (e.g. a T2 weighted MR image). A GAN contains two CNNs; the generator is trained to convert a noise vector to an image, and the discriminator is trained to classify if an image is real or fake. The two CNNs are trained together, hence the term adversarial, and therefore become better and better at generating images and classifying images as real or fake. Once the training is finished, a noise vector is given to the generator to obtain a new image, and it will be similar to the distribution of the training images. GANs can be difficult to train, since the balance between the generator and the discriminator is very important. The mode collapse problem means that the generator generates images that are all very similar. To make the training more stable, and to enable higher resolution synthetic images, progressive growing GANs (Karras et al., 2017) were proposed. Instead of directly learning a mapping from a 512 x 1 noise vector to a 1024 x 1024 image, the training is progressive, such the network first learns to generate 8 x 8 images, then 16 x 16 images and so on. Figure 17 shows some very realistic faces (1024 x 1024 pixels) generated by a progressive GAN.

GANs have so far mainly been used to generate 2D images, see (Kazeminia et al., 2018, Yi et al., 2018) for an overview of how GANs have been used in medical imaging. To use GANs to generate 3D volumes is in theory possible, but requires more calculations and much more memory. Gao et al. (2019) used 3D GANs to generate realistic synthetic lung nodules, but the generated volumes are rather small (40 x 40 x 18 voxels). Another drawback of GANs is that the training requires many real images, which results in a catch 22 situation where many real images are needed to generate new images, but only a small number of (annotated) images are normally available. A potential solution to this problem is to use large (open) datasets (e.g. containing T1-weighted MR images from healthy controls) to first train the GAN to synthesize images that are similar to the images of interest, and then fine tune the GAN by providing a lower number of images from a specific cohort (e.g. T1 weighted MR images from brain tumor patients).

Figure 17. Realistic synthetic faces, 1024 x 1024 pixels, created by a generative adversarial network (GAN), i.e. these persons do not exist but have been created by a computer from a noise vector (image from Karras et al., 2018). In a similar manner, a GAN can be used to generate realistic medical images which can be used for training of a CNN which performs segmentation.

Conditional GANs (Isola et al., 2015) can be used for another type of data augmentation, for example to generate T2-weighted MR images if only T1-weighted MR images are available (or generate CT from MR, or vice versa). Multi-channel CNNs can then be used for improved segmentation, compared to CNNs that use images from a single modality. Architectures such as CycleGAN (Zhu et al. , 2017), Figure 18, contain four CNNs; two CNNs are used for converting an image of type A to an image of type B, and two CNNs are used for converting from B to A. See Figures 19 and 20 for examples of converting between T1 and T2 weighted images, and converting between a T1 weighted image and a map of fractional anisotropy (FA) (obtained through diffusion MRI). Compared to using CNNs for segmentation, conditional GANs require training of four CNNs (two generators and two discriminators) and are therefore even more memory demanding, and very little work has therefore been done on 3D conditional GANs. An exception is the work by Näppi et al. (2019), who worked on volumes of size 96 x 96 x 96 voxels using a 16 GB graphics card, while MR volumes are often of the size 256 x 256 x 200 voxels.

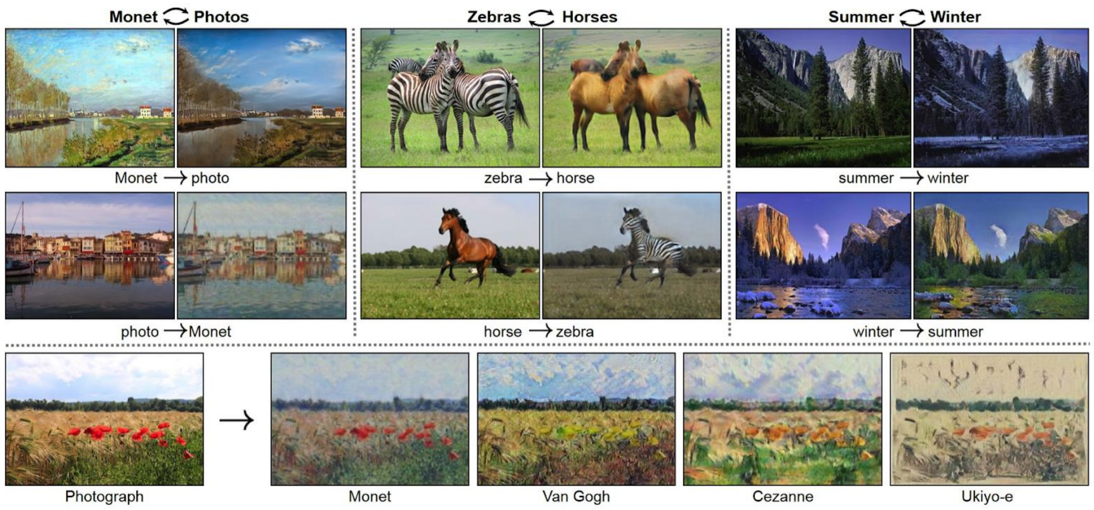

Figure 18. Conditional GANs, such as CycleGAN, can be trained to map an image from one domain to another domain, e.g. from winter to summer, which is a more advanced type of data augmentation compared to applying random rotations and scalings. Image from Zhu et al., 2017.

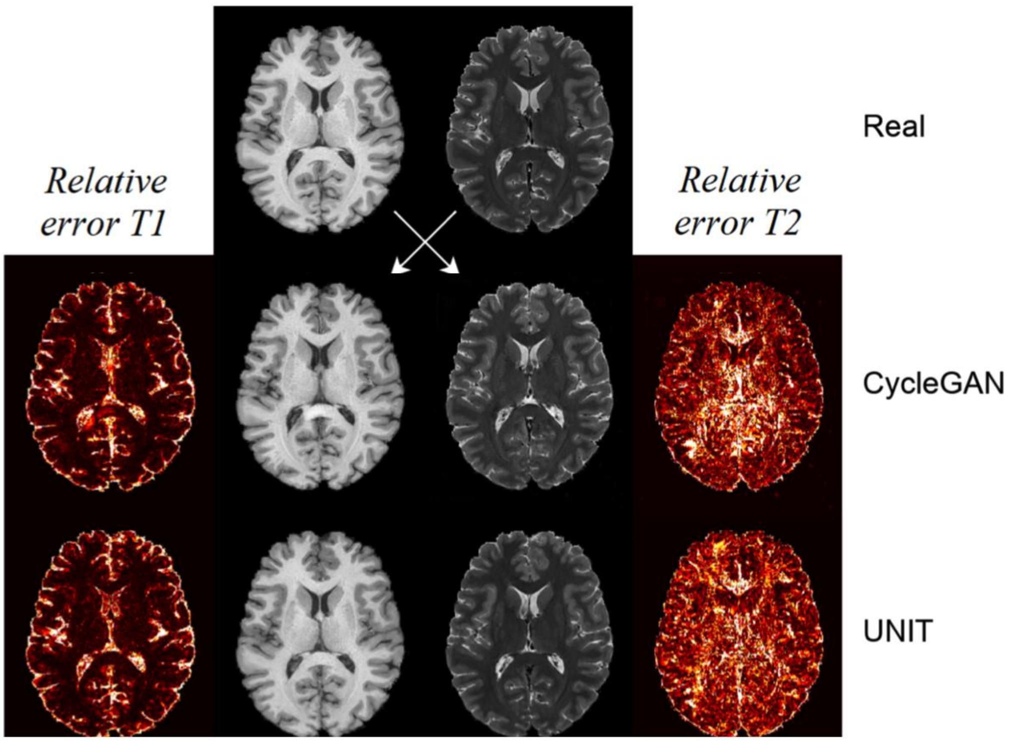

Figure 19. Using CycleGAN and UNIT to generate a T2-weighted image from a T1-weighted image, and vice versa. Image from (Welander et al., 2018).

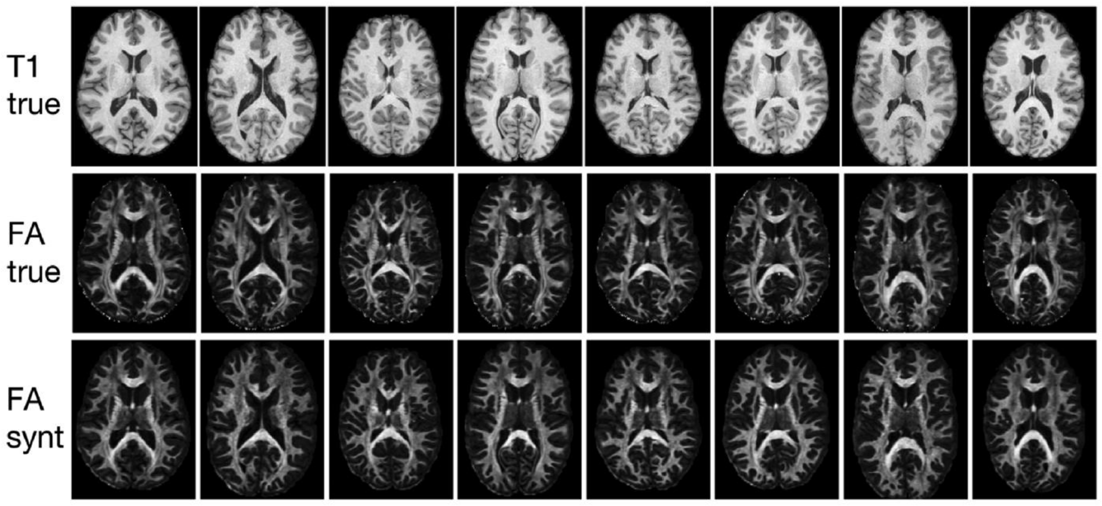

Figure 20. Using CycleGAN to generate an FA (fractional anisotropy) map from a T1-weighted image. Image from (Gu et al., 2019).

Rather than learning many one-to-one mappings (e.g. T1 to T2, T1 to FA etc), it is possible to use multi-conditional GANs to use all the available data to predict a missing modality. This can be very useful if a number of MR images are collected for every subject (e.g. T1, T1 contrast, T2, T2 FLAIR, like in the BraTS dataset) and one image is unusable due to head motion. The main idea of CollaGAN (Lee et al., 2019), Figure 21, is to combine all available data to improve the predictions of a missing image. According to the authors, this results in better predictions compared to CycleGAN.

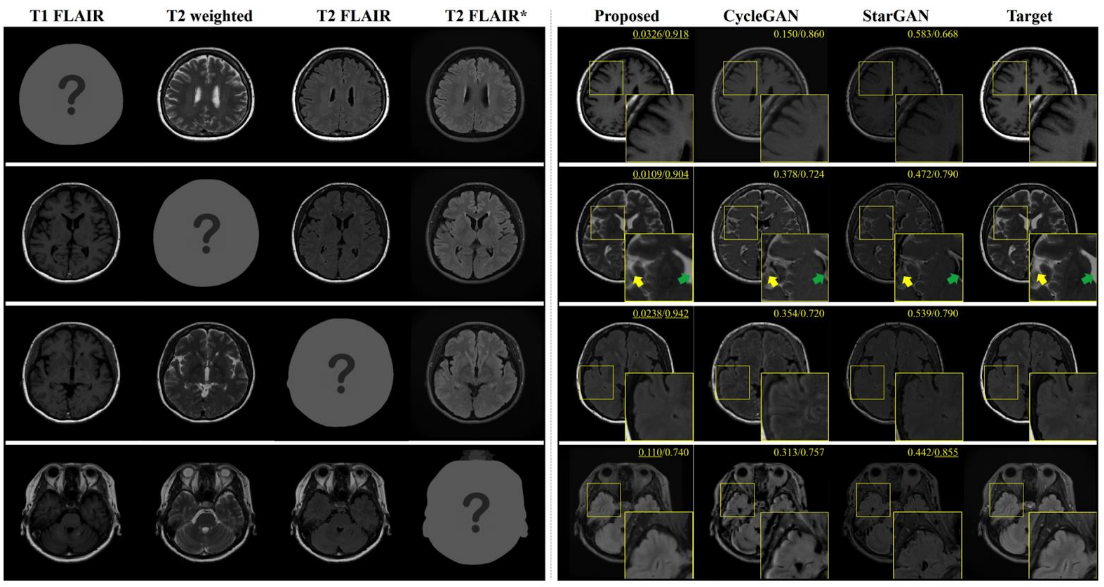

Figure 21. Instead of learning many one-to-one mappings for image-to-image translation (e.g. T1 to T2), CollaGAN uses all the available data to make better predictions of a missing image. Image from (Lee et al., 2019).

14 Overview of segmentation for MRI and CT using Deep Learning

This section contains an overview of the State of the Art Deep Learning segmentation methods for brain tumors, liver tumors and the lung. Where possible we report the results of deep learning segmentation applied to recent and publicly available (grand) challenge datasets as these allow for an objective evaluation and comparison.

Brain Tumor Segmentation in MRI

BraTS, the brain tumor segmentation challenge (Bakas et al 2019), has been held annually from 2012 to 2018 in conjunction with the MICCAI conference (International Conference on Medical Image Computing and Computer Assisted Intervention). The Brats 2018 dataset contains 285 multimodal cases, 210 high-grade gliomas and 75 low-grade gliomas. Each case consists of four aligned MRI modality volumes: native (T1), post-contrast T1-weighted, T2-weighted and T2 Fluid Attenuated Inversion Recovery (FLAIR), acquired with different clinical protocols and various scanners from 19 institutions. The submissions were ranked relative to their competitors for each of the testing subjects, evaluated regions and evaluations measures (e.g. dice and Hausdorff (95%)).

In 2018, all the best-ranked segmentation methods used Deep Learning methods although none outperformed the inter-rater agreement, across expert clinicians with years of training. The winning method used "an encoder-decoder based Convolutional Neural Network (CNN) architecture with an asymmetrically larger encoder to extract image features and a smaller decoder to reconstruct the segmentation mask" (Myronenko, 2018). Training the model for 300 iterations on all 285 cases took 2 days on a single professional-grade GPU (Nvidia Tesla V100 32GB) and 6 hours on an Nvidia DGX-1 server with 8 Tesla V100 GPUs [JD1]. Reported Dice values for enhancing tumor, whole tumor and tumor core on the test dataset were 0.77, 0.88 and 0.81 respectively.

The runner up approach, (Isensee et al., 2018), demonstrated that a well-trained U-net with minor modifications (e.g., region based training and a combination of loss functions) together with additional training data produces very competitive results indicating that a well-constructed and performed training process is at least as important as focusing on novel architectural modifications when it comes to segmentation. The shared third place went to (McKinley et al, 2018) who use a DenseNet, in which pooling layers are replaced by dilated convolutions embedded in a U-net style network and (Zhou et al, 2019) who use a 3D variant of FusionNet.

Although accuracy of the individual automated segmentation methods appears to have improved in the latest 2018 challenge, compared to earlier editions of the challenge, the authors find that the level of robustness, i.e. inter-rater agreement, is still inferior to expert-performance. This can potentially be solved in the future as the training set is extended with more diverse patient data, and through improvements in machine learning architectures and training schemes.

Liver Tumor Segmentation in CT

LiTS, the liver tumor segmentation challenge which was organized in conjunction with the IEEE International Symposium on Biomedical Imaging 3 (ISBI) 2017 and MICCAI 2017 conferences but is still running. The LiTS 2017 dataset contains 201 CT scans (131 for training, 70 for testing) from 7 different hospitals and research institutions (Bilic et al, 2019). Ultimately, not a single algorithm performed best for both liver and tumor segmentation. The best liver segmentation had a Dice score of 0.96 achieved at MICCA, while the best tumor segmentation Dice scores were 0.67 for ISBI and 0.70 for MICCAI.

Most of the automatic segmentation methods submitted used U-net derived architectures with only 2 submissions using a modified VCG-net and k-CNN approach respectively. The majority of the U-net approaches used a cascade of U-nets with separate U-nets focusing on liver segmentation and tumor segmentation. The most common optimizers where ADAM (Kingma &Ba 2014) and Stochastic gradient descent with momentum. Most approaches also used data pre-processing with HU-value clipping, normalization and standardization. The winning ISBI 2017 method by X. Han from Elekta Inc (Dice score 0.67) as well as the second best ISBI 2017 method by Vorontsov (Dice score 0.65) both relied on integrated residual connections which allow information to flow through the deep learning network and give later layers access to feature maps of previous layers.

For MICCAI the challenge was extended with liver segmentation and tumor burden estimation tasks. In general the winning submissions for the MICCAI 2017 challenge integrated and improved on the results from the ISBI 2017 challenge resulting in greatly improved Dice scores winner was Tian et al from Lenovo with a Dice score of 0.70). As of this writing, the current runner-up method is the nnU-net (No-New U-net) approach by Isensee et al, 2019b , with a tumor dice score of 0.74 and a liver dice score of 0.96, which also scored second place in the BraTS 2018 brain tumor challenge.

Lung Segmentation

For the lungs there has not been a recent segmentation challenge so we will report some of the most interesting recent publications.

In Ferreira et al. (2018), the authors present a lung lobe segmentation method using a 3D Fully Convolutional Neural Network based on the V-Net architecture, which is a 3D extension to U-net, with specifically selected regularization techniques. These regularization techniques are used to prevent overfitting in small training sets resulting in poor generalization. Since the number of training and validation datasets was quite small and the ground-truth data was generated by a single radiologist it is hard to compare the performance to other methods.

In Yun et al. (2019), the authors present a novel airway segmentation method on CT using a 2.5 dimensional CNN trained in a supervised manner which uses three adjacent slices in each of the orthogonal directions (axial, sagittal and coronal) and fine-tuned the parameters. The method was trained on 69 selected multi-center CT scans from the Korean obstructive lung disease (KOLD) cohort. The method was validated on 8 separate test cases from the KOLD cohort as well as the 20 test cases from the EXACT'09 public dataset. As the EXACT'09 challenge is no longer running, it is difficult to compare the results directly but the method of Yun et al 2019 seems very competitive both in accuracy and speed.

15 References

Bakas et al. “Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge (rev2)”. arXiv preprint arXiv:1811.02629 (2019)

Akkus, Z., Galimzianova, A., Hoogi, A., Rubin, D. L., & Erickson, B. J. (2017). Deep learning for brain MRI segmentation: state of the art and future directions. Journal of digital imaging, 30(4), 449-459.

Bilic, P. et al. (2019). The Liver Tumor Segmentation Benchmark (LiTS). *arXiv preprint arXiv:1901.04056*.

Chen, Hao, et al. "VoxResNet: Deep voxelwise residual networks for brain segmentation from 3D MR images." *NeuroImage* 170 (2018): 446-455.

Chen, J., Yang, L., Zhang, Y., Alber, M. and Chen, D.Z., 2016. Combining fully convolutional and recurrent neural networks for 3d biomedical image segmentation. In Advances in neural information processing systems (pp. 3036-3044).

Despotović, I., Goossens, B., & Philips, W. (2015). MRI segmentation of the human brain: challenges, methods, and applications. Computational and mathematical methods in medicine, 2015.

Diederik P. Kingma and Jimmy Lei Ba. Adam : A method for stochastic optimization. 2014. arXiv:1412.6980v9

Dolz, Jose, Christian Desrosiers, and Ismail Ben Ayed. "3D fully convolutional networks for subcortical segmentation in MRI: A large-scale study." *NeuroImage* 170 (2018): 456-470.

Eklund, A., Dufort, P., Forsberg, D., & LaConte, S. M., Medical image processing on the GPU–Past, present and future. *Medical image analysis*, *17*(8), 1073-1094, 2013

Gao, C., Clark, S., Furst, J., & Raicu, D. (2019, March). Augmenting LIDC dataset using 3D generative adversarial networks to improve lung nodule detection. In *Medical Imaging 2019: Computer-Aided Diagnosis* (Vol. 10950, p. 109501K). International Society for Optics and Photonics.

Gaonkar, Bilwaj, et al. "Extreme Augmentation: Can deep learning based medical image segmentation be trained using a single manually delineated scan?." *arXiv preprint arXiv:1810.01621*, 2018

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y,
Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672-2680, 2014

Gu, X., Knutsson, H., Nilsson, M., Eklund, A., Generating diffusion MRI scalar maps from T1 weighted images using generative adversarial networks, Scandinavian Conference on Image Analysis (SCIA), 2019

Harari, Paul M., Shiyu Song, and Wolfgang A. Tomé. "Emphasizing conformal avoidance versus target definition for IMRT planning in head-and-neck cancer." *International Journal of Radiation Oncology* Biology* Physics* 77.3 (2010): 950-958

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q., 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).

Isensee F., Kickingereder P., Wick W., Bendszus M., Maier-Hein K.H. (2019a) No New-Net. In: Crimi A., Bakas S., Kuijf H., Keyvan F., Reyes M., van Walsum T. (eds) Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. BrainLes 2018. Lecture Notes in Computer Science, vol 11384. Springer

Isensee, Fabian, et al. "nnU-Net: Breaking the Spell on Successful Medical Image Segmentation." arXiv preprint arXiv:1904.08128 (2019b).

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A.,. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125-1134, 2017

Karras, T., Aila, T., Laine, S., & Lehtinen, J., Progressive growing of GANs for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017

Kazeminia, S., Baur, C., Kuijper, A., van Ginneken, B., Navab, N., Albarqouni, S., & Mukhopadhyay, A. (2018). GANs for medical image analysis. *arXiv preprint arXiv:1809.06222*.

Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), pp.2278-2324.

Lee, D., Kim, J., Moon, W. J., & Ye, J. C. (2019). CollaGAN: Collaborative GAN for Missing Image Data Imputation. *arXiv preprint arXiv:1901.09764*.

Lundervold, A.S. and Lundervold, A., 2018. An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift für Medizinische Physik*.

McKinley R., Meier R., Wiest R. (2019) Ensembles of Densely-Connected CNNs with Label-Uncertainty for Brain Tumor Segmentation. In: Crimi A., Bakas S., Kuijf H., Keyvan F., Reyes M., van Walsum T. (eds) Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. BrainLes 2018. Lecture Notes in Computer Science, vol 11384. Springer

Mehta, R. and Sivaswamy, J., 2017. M-net: A convolutional neural network for deep brain structure segmentation. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)* (pp. 437-440). IEEE.

Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., ... & Lanczi, L. (2015). The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE transactions on medical imaging*, *34*(10), 1993-2024.

Myronenko A. (2019) 3D MRI Brain Tumor Segmentation Using Autoencoder Regularization. In: Crimi A., Bakas S., Kuijf H., Keyvan F., Reyes M., van Walsum T. (eds) Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. BrainLes 2018. Lecture Notes in Computer Science, vol 11384. Springer

Nelms, Benjamin E., et al. "Variations in the contouring of organs at risk: test case from a patient with oropharyngeal cancer." *International Journal of Radiation Oncology* Biology* Physics* 82.1 (2012): 368-378

Näppi, J. J., & Yoshida, H. (2019, March). Cycle-consistent 3D-generative adversarial network for virtual bowel cleansing in CT colonography. In *Medical Imaging 2019: Image Processing* (Vol. 10949, p. 109492Z). International Society for Optics and Photonics.

Poudel, R.P., Lamata, P. and Montana, G., 2016. Recurrent fully convolutional neural networks for multi-slice MRI cardiac segmentation. In *Reconstruction, segmentation, and analysis of medical images* (pp. 83-94). Springer, Cham.

Prasoon, A., Petersen, K., Igel, C., Lauze, F., Dam, E. and Nielsen, M., 2013, September. Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In *International conference on medical image computing and computer-assisted intervention* (pp. 246-253). Springer, Berlin, Heidelberg.

Pun, Chi-Man, and Moon-Chuen Lee. "Log-polar wavelet energy signatures for rotation and scale invariant texture classification." *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 25, 590-603, 2003

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

Roth, H.R., Lu, L., Seff, A., Cherry, K.M., Hoffman, J., Wang, S., Liu, J., Turkbey, E. and Summers, R.M., 2014, September. A new 2.5 D representation for lymph node detection using random sets of deep convolutional neural network observations. In *International conference on medical image computing and computer-assisted intervention* (pp. 520-527). Springer, Cham.

Sandström, Helena, et al. "Assessment of organs-at-risk contouring practices in radiosurgery institutions around the world–The first initiative of the OAR Standardization Working Group." *Radiotherapy and Oncology* 121.2 (2016): 180-186

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*(pp. 1-9).

Wachinger, Christian, Martin Reuter, and Tassilo Klein. "DeepNAT: Deep convolutional neural network for segmenting neuroanatomy." *NeuroImage* 170 (2018): 434-445.

Welander, P., Karlsson, S., & Eklund, A., Generative adversarial networks for image-to-image translation on multi-contrast MR images-A comparison of CycleGAN and UNIT. *arXiv preprint arXiv:1806.07777*, 2018

Wolberg, George, and Siavash Zokai. "Robust image registration using log-polar transform." *IEEE International Conference on Image Processing,* 2000

Yang, J., Veeraraghavan, H., Armato III, S.G., Farahani, K., Kirby, J.S., Kalpathy-Kramer, J., van Elmpt, W., Dekker, A., Han, X., Feng, X. and Aljabar, P., 2018. Autosegmentation for thoracic radiation treatment planning: A grand challenge at AAPM 2017. *Medical physics*, *45*(10), pp.4568-4581.

Yi, X., Walia, E., & Babyn, P. (2018). Generative adversarial network in medical imaging: A review. *arXiv preprint arXiv:1809.07294*.

Yun et al. Improvement of fully automated airway segmentation on volumetric computed tomographic images using a 2.5 dimensional convolutional neural net. Medical Image Analysis vol 51, pages 13-20, 2019

Zhou C., Chen S., Ding C., Tao D. (2019) Learning Contextual and Attentive Information for Brain Tumor Segmentation. In: Crimi A., Bakas S., Kuijf H., Keyvan F., Reyes M., van Walsum T. (eds) Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. BrainLes 2018. Lecture Notes in Computer Science, vol 11384. Springer

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A., 2017, Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223-2232, 2017