Medical Image Segmentation Using Deep Learning

Han Liu, Dewei Hu, Hao Li, Ipek Oguz

Chapter 13

Medical Image Segmentation Using Deep Learning

Han Liu, Dewei Hu, Hao Li, and Ipek Oguz

Abstract

Image segmentation plays an essential role in medical image analysis as it provides automated delineation of specific anatomical structures of interest and further enables many downstream tasks such as shape analysis and volume measurement. In particular, the rapid development of deep learning techniques in recent years has had a substantial impact in boosting the performance of segmentation algorithms by efficiently leveraging large amounts of labeled data to optimize complex models (supervised learning). However, the difficulty of obtaining manual labels for training can be a major obstacle for the implementation of learning-based methods for medical images. To address this problem, researchers have investigated many semi-supervised and unsupervised learning techniques to relax the labeling requirements. In this chapter, we present the basic ideas for deep learning-based segmentation as well as some current state-of-the-art approaches, organized by supervision type. Our goal is to provide the reader with some possible solutions for model selection, training strategies, and data manipulation given a specific segmentation task and dataset.

Key words Image segmentation, Deep learning, Semi-supervised method, Unsupervised method, Medical image analysis

1 Introduction

Image segmentation is an essential and challenging task in medical image analysis. Its goal is to delineate the object boundaries by assigning each pixel/voxel a label, where pixels/voxels with the same labels share similar properties or belong to the same class. In the context of neuroimaging, robust and accurate image segmentation can effectively help neurosurgeons and doctors, e.g., measure the size of brain lesions or quantitatively evaluate the volume changes of brain tissue throughout treatment or surgery. For instance, quantitative measurements of subcortical and cortical structures are critical for studies of several neurodegenerative diseases such as Alzheimer's, Parkinson's, and Huntington's diseases.

Authors Han Liu, Dewei Hu, and Hao Li have equal contributors to this chapter

Olivier Colliot (ed.), *Machine Learning for Brain Disorders*, Neuromethods, vol. 197, https://doi.org/10.1007/978-1-0716-3195-9_13,

Automatic segmentation of multiple sclerosis (MS) lesions is essential for the quantitative analysis of disease progression. The delineation of acute ischemic stroke lesions is crucial for increasing the likelihood of good clinical outcomes for the patient. While manual delineation of object boundaries is a tedious and time-consuming task, automatic segmentation algorithms can significantly reduce the workload of clinicians and increase the objectivity and reproducibility of measurements. To be specific, the segmentation task in medical images usually refers to semantic segmentation. For example, for paired brain structures (e.g., left and right pairs of subcortical structures), the instances of the same category will not be specified in the segmentation, in contrast to instance and panoptic segmentation.

There are many neuroimaging modalities such as magnetic resonance imaging, computed tomography, transcranial Doppler, and positron emission tomography. Moreover, neuroimaging studies often contain multimodal and/or longitudinal data, which can help improve our understanding of the anatomical and functional properties of the brain by utilizing complementary physical and physiological sensitivities. In this chapter, we first present some background information to help readers get familiar with the fundamental elements used in deep learning-based segmentation frameworks. Next, we discuss the learning-based segmentation approaches in the context of different supervision settings, along with some real-world applications.

2 Methods

2.1 Fundamentals

2.1.1 Common Network Architectures for Segmentation Tasks

Convolutional neural networks (CNNs) dominated the medical image segmentation field in recent years. CNNs leverage information from images to predict segmentations by hierarchically learning parameters with linear and nonlinear layers. We begin by discussing some popular models and their architectures: (1) U-Net [1], (2) V-Net [2], (3) attention U-Net [3, 4], and (4) nnU-Net [5, 6].

U-Net is the most popular model for medical image segmentation, and its architecture is shown in Fig. 1. The network has two main parts: the encoder and the decoder, with skip connections in between. The encoder consists of two repeated 3×3 convolutions (conv) without zero-padding, a rectified linear unit (ReLU) activation function. A max-pooling operation with stride 2 is used for connecting different levels or downsampling. We note that the channel number of feature maps is doubled at each subsequent level. In the symmetric decoder counterpart, a 2×2 up-convolution (up-conv) is used not only for upsampling but also for reducing the number of channels by half. The center-cropped feature map from the encoder is delivered to the decoder

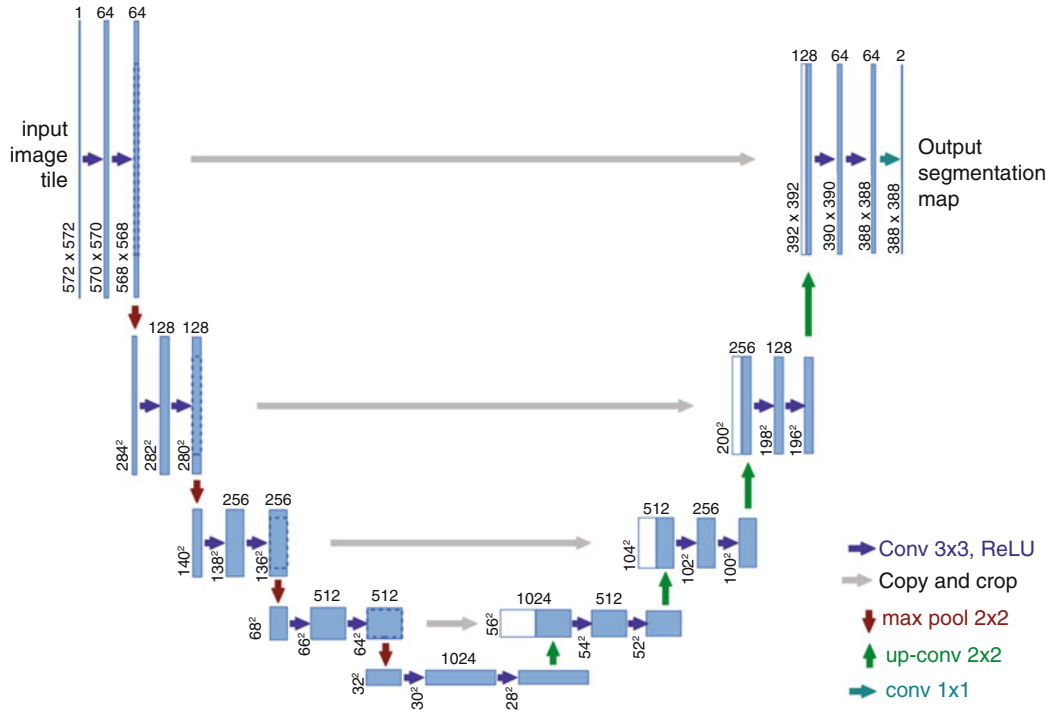

Fig. 1 U-Net architecture. Blue boxes are the feature maps. Channel numbers are denoted above each box, while the tensor sizes are denoted on the lower left. White boxes show the concatenations and arrows indicate various operations. ©2015 Springer Nature. Reprinted, with permission, from [1]

via skip connections at each level to preserve the low-level information. The cropping is needed to maintain the same size between feature maps for concatenation. Next, two repeated 3×3 conv and ReLU are applied. Lastly, a 1×1 conv is employed for converting the channel number to the desired number of classes C. In this configuration, the network takes a 2D image as input and produces a segmentation map with C classes. Later, a 3D U-Net [7] was introduced for volumetric segmentation that learns from volumetric images.

V-Net is another popular model for volumetric medical image segmentation. Based upon the overall structure of the U-Net, the V-Net [2] leverages the residual block [8] to replace the regular conv, and the convolution kernel size is enlarged to $5 \times 5 \times 5$. The residual blocks can be formulated as follows: (1) the input of a residual block is processed by conv layers and nonlinearities, and (2) the input is added to the output from the last conv layer or nonlinearity of the residual block. It consists of a fully convolutional neural network trained end-to-end.

Attention U-Net is a model based on U-Net with attention gates (AG) in the skip connections (Fig. 2). The attention gates can learn to focus on the segmentation target. The salient features are

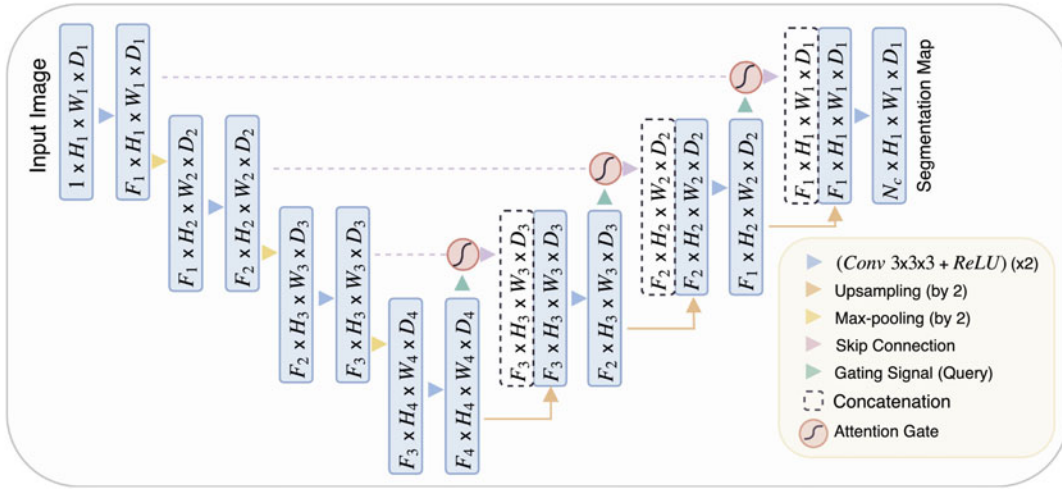

Fig. 2 Attention U-Net architecture. H_i, W_i, and D_i represent the height, width, and depth of the feature map at the i^{th} layer of the U-Net structure. F_i indicates the number of feature map channels. Replicated from [4] (CC BY 4.0)

emphasized with larger weights from the CNN during the training. This leads the model to achieve higher accuracy on target structures with various shapes and sizes. In addition, AGs are easy to integrate into the existing popular CNN architectures. The details of the attention mechanism and attention gates are discussed in Subheading 2.1.2. More details on attention can also be found in Chap. 6.

nnU-Net is a medical image segmentation pipeline that can achieve a self-configuring network architecture based on the different datasets and tasks it is given, without any manual intervention. According to the dataset and task, nnU-Net will generate one of (1) 2D U-Net, (2) 3D U-Net, and (3) cascaded 3D U-Net for the segmentation network. For cascaded 3D U-Net, the first network takes downsampled images as inputs, and the second network uses the image at full resolution as input to refine the segmentation accuracy. The nnU-Net is often used as a baseline method in many medical image segmentation challenges, because of its robust performance across various target structures and image properties. The details of nnU-Net can be found in [6].

2.1.2 *Attention Modules*

Although the U-Net architecture described in Subheading 2.1.1 has achieved remarkable success in medical image segmentation, the downsampling steps included in the encoder path can induce poor segmentation accuracy for small-scale anatomical structures (e.g., tumors and lesions). To tackle this issue, the attention modules are often applied so that the salient features are enhanced by higher weights, while the less important features are ignored. This subsection will introduce two types of attention mechanisms: additive attention and multiplicative attention.

Additive Attention As discussed in the previous section, U-Net is the most popular backbone for medical image analysis tasks. The downsampling enables it to work on features of different scales. Suppose we are working on a 3D segmentation problem. The output of the U-Net encoder at the lth level is then a tensor $\boldsymbol{X}^l$ of size $[F_l, H_l, W_l, D_l]$, where H_l, W_l, D_l denote the height, width, and depth of the feature map, respectively, and F_l represents the length of the feature vectors. We regard the tensor as a set of feature vectors $\boldsymbol{x}_i^l$:

$$X^l = \{\boldsymbol{x}_i^l\}_{i=1}^n, \quad \boldsymbol{x}_i^l \in \mathbb{R}^{F_l} \tag{1}$$

where $n = H_l \times W_l \times D_l$. The attention gate assigns a weight α_i to each vector $\boldsymbol{x}_i$ so that the model can concentrate on salient features. Ideally, important features are assigned higher weight that will not vanish when downsampling. The output of the attention gate will be a collection of weighted feature vectors:

$$\hat{X}^l = \{\alpha_i^l \cdot \boldsymbol{x}_i^l\}_{i=1}^n, \quad \alpha_i^l \in \mathbb{R} \tag{2}$$

These weights α_i, also known as gating coefficients, are determined by an attention mechanism that delineates the correlation between the feature vector $\boldsymbol{x}$ and a gating signal $\boldsymbol{g}$. As shown in Fig. 3, for all $\boldsymbol{x}_i^l \in X^l$, we compute an additive attention with regard to a corresponding $\boldsymbol{g}_i$ by

$$s_{att}^l = \boldsymbol{\psi}^\top \left[\sigma_1\left(\boldsymbol{W}_x^\top \boldsymbol{x}_i^l + \boldsymbol{W}_g^\top \boldsymbol{g}_i + \boldsymbol{b}_g\right)\right] + b_\psi \tag{3}$$

where $\boldsymbol{b}_g$ and b_ψ represent the bias and $\boldsymbol{W}_x$, $\boldsymbol{W}_g$, ψ are linear transformations. The output dimension of the linear transformation is $\mathbb{R}^{F_{int}}$ where F_{int} is a self-defined integer. Denote these

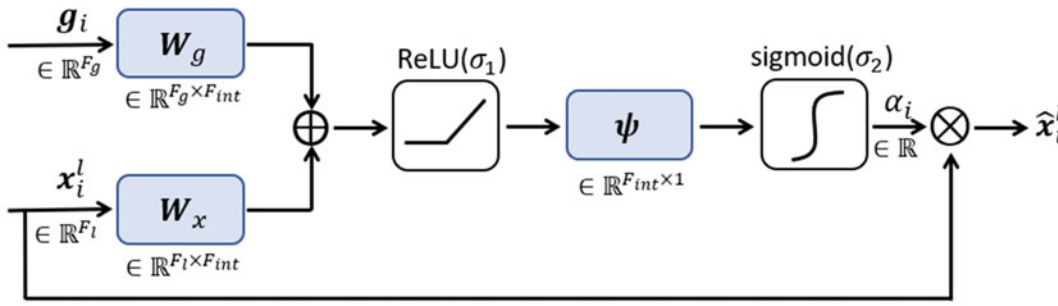

Fig. 3 The structure of the additive attention gate. x_i^l is the ith feature vector at the lth level of the U-Net structure and $\boldsymbol{g}_i$ is the corresponding gating signal. $\boldsymbol{W}_x$ and $\boldsymbol{W}_g$ are the linear transformation matrices applied to x_i^l and $\boldsymbol{g}_i$, respectively. The sum of the resultant vectors will be activated by ReLU and then its dot product with a vector $\boldsymbol{\psi}$ is computed. The sigmoid function is used to normalize the resulting scalar to [0, 1] range, which is the gating coefficient α_i. The weighted feature vector is denoted by $\hat{x}_i^l$. Adapted from [4] (CC BY 4.0)

learnable parameters by a set Θ_{att}. The coefficients s^l_{att} are normalized to [0, 1] by a sigmoid function σ_2:

$$\alpha^l_i = \sigma_2\left(s^l_{att}(\boldsymbol{x}^l_i, \boldsymbol{g}_i; \Theta_{att})\right) \tag{4}$$

Basically, the attention gate is thus a linear combination of the feature vector and the gating signal. In practical applications [3, 4, 9], the gating signal is chosen to be the coarser feature space as indicated in Fig. 2. In other words, for input feature $\boldsymbol{x}^l_i$, the corresponding gating signal is defined by

$$\boldsymbol{g}_i = \boldsymbol{x}^{l+1}_i \tag{5}$$

Note that an extra downsampling step should be applied on $\boldsymbol{X}^l$ so that it has the same shape as $\boldsymbol{X}^{l+1}$. In experiments to segment brain tumor on MRI datasets [9] and the pancreas on CT abdominal datasets [4], AG was shown to improve the segmentation performance for diverse types of model backbones including U-Net and Residual U-Net.

Multiplicative Attention Similar to additive attention, the multiplicative mechanism can also be leveraged to compute the importance of feature vectors. The basic idea of multiplicative attention was first introduced in machine translation [11]. Evolving from that, Vaswani et al. proposed a groundbreaking transformer architecture [10] which has been widely implemented in image processing [12, 13]. In recent research, transformers have been incorporated with the U-Net structure [14, 15] to improve medical image segmentation performance.

The attention function is described by matching a query vector $\boldsymbol{q}$ with a set of key vectors $\{\boldsymbol{k}_1, \boldsymbol{k}_2, ..., \boldsymbol{k}_n\}$ to obtain the weights of the corresponding values $\{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n\}$. Figure 4a shows an example for $n = 4$. Suppose the vectors $\boldsymbol{q}$, $\boldsymbol{k}_i$, and $\boldsymbol{v}_i$ have the same dimension $\mathbb{R}^d$. Then, the attention function is

$$s_i = \frac{\boldsymbol{q}^\top \boldsymbol{k}_i}{\sqrt{d}} \tag{6}$$

We note that the dot product can have large magnitude when d is large, which can cause gradient vanishing problem in the softmax function; s_i is normalized by the size of the vector to alleviate this. Equation 13.6 is a commonly used attention function in transformers. There are some other options including $s_i = \boldsymbol{q}^\top \boldsymbol{k}_i$ and $s_i = \boldsymbol{q}^\top \boldsymbol{W}\boldsymbol{k}_i$ where $\boldsymbol{W}$ is a learnable parameter. Generally, the attention value s_i is determined by the similarity between the query and the key. Similar to the additive attention gate, these attention values are normalized to [0, 1] by a softmax function σ_3:

$$\alpha_i = \sigma_3(s_1, ..., s_n) = \frac{e^{s_i}}{\sum_{j=1}^{n} e^{s_j}} \tag{7}$$

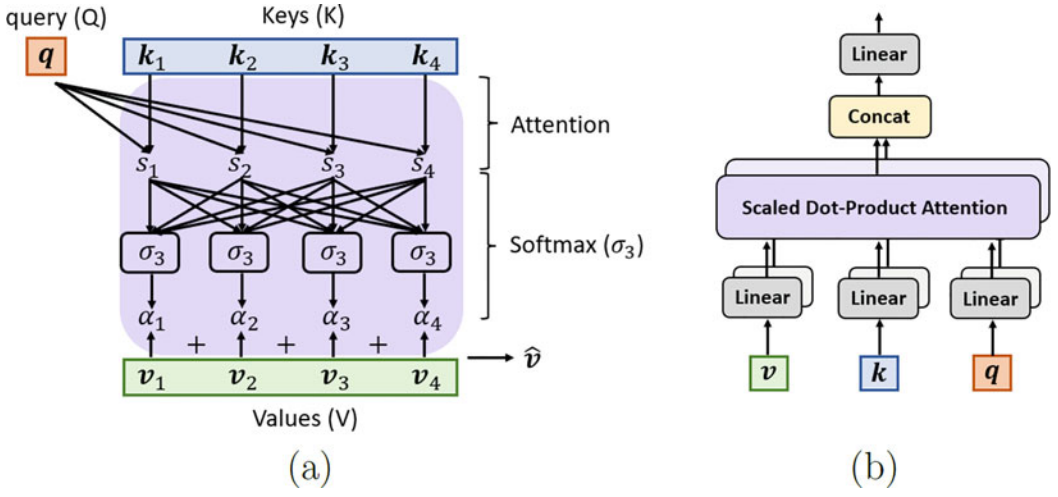

Fig. 4 (**a**) The dot-product attention gate. $\boldsymbol{k}_i$ are the keys and $\boldsymbol{q}$ is the query vector. s_i are the outputs of the attention function. By using the softmax σ_3, the attention coefficients α_i are normalized to [0, 1] range. The output will be the weighted sum of values v_i. (**b**) The multi-head attention is implemented in transformers. The input values, keys, and query are linearly projected to different spaces. Then the dot-product attention is applied on each space. The resultant vectors are concatenated by channel and passed through another linear transformation. Image (**b**) is adapted from [10]. Permission to reuse was kindly granted by the authors

The output of the attention gate will be $\hat{\boldsymbol{v}} = \sum_{i=1}^{n} \alpha_i \boldsymbol{v}_i$. In the transformer application, the values, keys, and queries are usually linearly projected into several different spaces, and then the attention gate is applied in each space as illustrated in Fig. 4b. This approach is called multi-head attention; it enables the model to jointly attend to information from different subspaces.

In practice, the value $\boldsymbol{v}_i$ is often defined by the same feature vector as the key $\boldsymbol{k}_i$. This is why the module is also called multi-head self-attention (MSA). Chen et al. proposed the TransUNet [15], which leverages this module in the bottleneck of a U-Net as shown in Fig. 5. They argue that such a combination of a U-Net and the transformer achieves superior performance in multi-organ segmentation tasks.

2.1.3 Loss Functions for Segmentation Tasks

This section summarizes some of the most widely used loss functions for medical image segmentation (Fig. 6) and describes their usage in different scenarios. A complementary reading material for an extensive list of loss functions can be found in [16, 17]. In the following, the predicted probability by the segmentation model and the ground truth at the ith pixel/voxel are denoted as p_i and g_i, respectively. N is the number of voxels in the image.

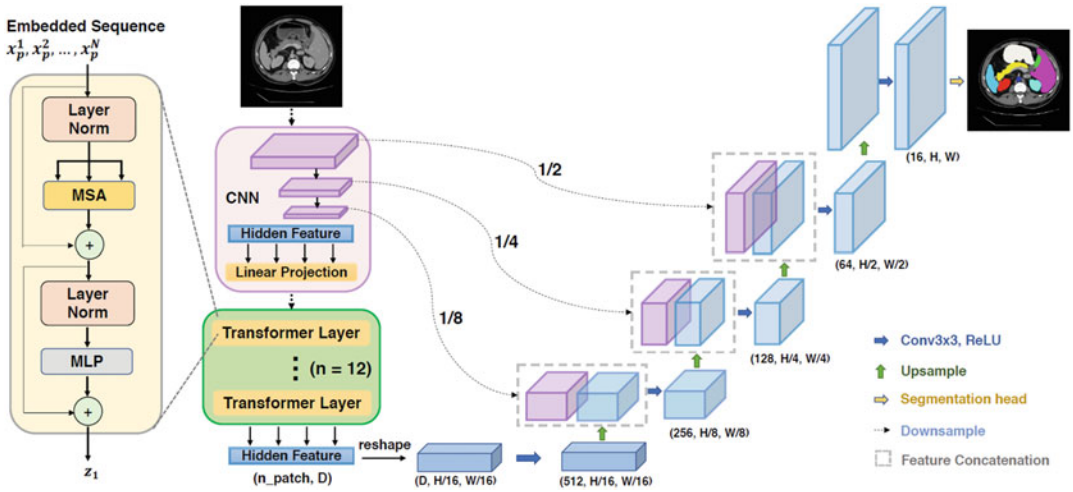

Fig. 5 The architecture of TransUNet. The transformer layer represented by the yellow box shows the application of multi-head attention (MSA). MLP represents the multilayer perceptron. In general, the feature vectors in the bottleneck of the U-Net are set as the input to the stack of *n* transformer layers. As these layers will not change the dimension of the features, they are easy to be implemented and will not affect other parts of the U-Net model. Replicated from [15] (CC BY 4.0)

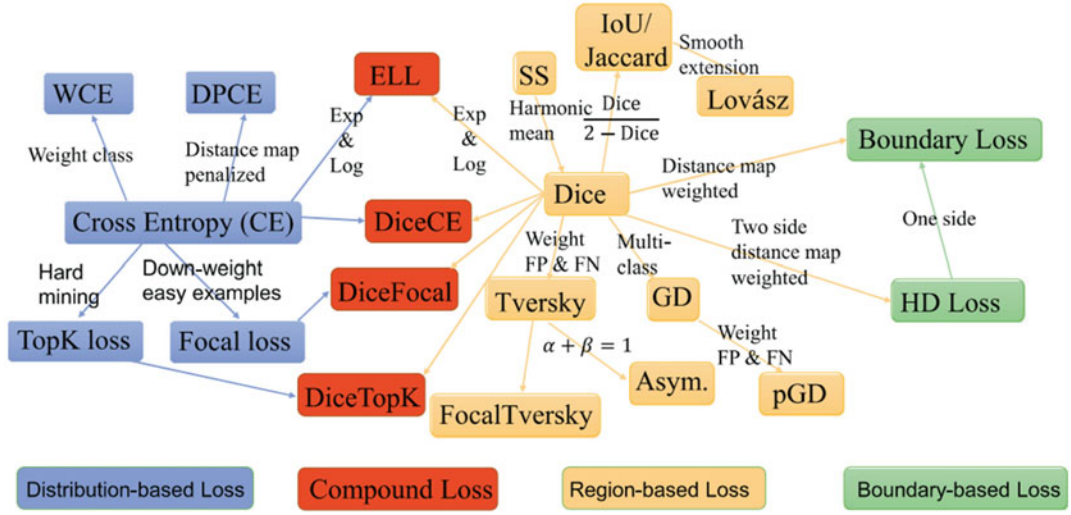

Fig. 6 Loss functions for medical image segmentation. WCE: weighted cross-entropy loss. DPCE: distance map penalized cross-entropy loss. ELL: exponential logarithmic loss. SS: sensitivity-specificity loss. GD: generalized Dice loss. pGD: penalty loss. Asym: asymmetric similarity loss. IoU: intersection over union loss. HD: Hausdorff distance loss. ©2021 Elsevier. Reprinted, with permission, from [16]

Cross-Entropy Loss Cross-entropy (CE) is defined as a measure of the difference between two probability distributions for a given random variable or set of events. This loss function is used for pixel-wise classification in segmentation tasks:

$$\ell_{CE} = -\sum_{i}^{N}\sum_{k}^{K} y_i^k \log(p_i^k) \tag{8}$$

where N is the number of voxels, K is the number of classes, y_i^k is a binary indicator that shows whether k is the correct class, and p_i^k is the predicted probability for voxel i to be in kth class.

Weighted Cross-Entropy Loss Weighted cross-entropy (WCE) loss is a variant of the cross-entropy loss to address the class imbalance issue. Specifically, class-specific coefficients are used to weigh each class differently, as follows:

$$\ell_{WCE} = -\sum_{i}^{N}\sum_{k}^{K} w_{y_k} y_i^k \log(p_i^k) \tag{9}$$

Here, w_{y_k} is the coefficient for the kth class. Suppose there are 5 positive samples and 12 negative samples in a binary classification training set. By setting $w_0 = 1$ and $w_1 = 2$, the loss would be as if there were ten positive samples.

Focal Loss Focal loss was proposed to apply a modulating term to the CE loss to focus on hard negative samples. It is a dynamically scaled CE loss, where the scaling factor decays to zero as confidence in the correct class increases. Intuitively, this scaling factor can automatically down-weight the contribution of easy examples during training and rapidly focus the model on hard examples:

$$\ell_{Focal} = -\sum_{i}^{N} \alpha_i (1 - p_i)^{\gamma} \log(p_i) \tag{10}$$

Here, α_i is the weighing factor to address the class imbalance and γ is a tunable focusing parameter ($\gamma > 0$).

Dice Loss The Dice coefficient is a widely used metric in the computer vision community to calculate the similarity between two binary segmentations. In 2016, this metric was adapted as a loss function for 3D medical image segmentation [2]:

$$\ell_{Dice} = 1 - \frac{2\sum_i^N p_i g_i + 1}{\sum_i^N (p_i + g_i) + 1} \tag{11}$$

Generalized Dice Loss Generalized Dice loss (GDL) [18] was proposed to reduce the well-known correlation between region size and Dice score:

$$L_{GDL} = 1 - 2\frac{\sum_{l=1}^{2} w_l \sum_i^N p_i g_i}{\sum_{l=1}^{2} w_l \sum_i^N p_i + g_i} \tag{12}$$

Here $w_l = \frac{1}{(\sum_i^N g_{li})^2}$ is used to provide invariance to different region sizes, i.e., the contribution of each region is corrected by the inverse of its volume.

Tversky Loss The Tversky loss [19] is a generalization of the Dice loss by adding two weighting factors α and β to the FP (false positive) and FN (false negative) terms. The Tversky loss is defined as

$$L_{Tversky} = 1 - \frac{\sum_i^N p_i g_i}{\sum_i^N p_i g_i + \alpha(1 - g_i)p_i + \beta(1 - p_i)g_i} \tag{13}$$

Recently, a comprehensive study [16] of loss functions on medical image segmentation tasks shows that using Dice-related compound loss functions, e.g., Dice loss + CE loss, is a better choice for new segmentation tasks, though none of losses can consistently achieve the best performance on multiple segmentation tasks. Therefore, for a new segmentation task, we recommend the readers to start with Dice + CE loss, which is also the default loss function in one of the most popular medical image segmentation frameworks, nnU-Net [6].

Finally, note that other loss functions have also been proposed to introduce prior knowledge about size, topology, or shape, for instance [20].

2.1.4 Early Stopping

Given a loss function, a simple strategy for training is to stop the training process once a predetermined maximum number of iterations are reached. However, too few iterations would lead to an under-fitting problem, while over-fitting may occur with too many iterations. "Early stopping" is a potential method to avoid such issues. The training set is split into training and validation sets when using the early stopping condition. The early stopping condition is based on the performance on the validation set. For example, if the validation performance (e.g., average Dice score) does not increase for a number of iterations, the early stopping condition is triggered. In this situation, the best model with the highest performance on the validation set is saved and used for inference. Of course, one should not report the validation performance for the validation of the model. Instead, one should use a separate test set which is kept unseen during training for an unbiased evaluation.

2.1.5 Evaluation Metrics for Segmentation Tasks

Various metrics can quantitatively evaluate different aspects of a segmentation algorithm. In a binary segmentation task, a true positive (TP) indicates that a pixel in the target object is correctly predicted as target. Similarly, a true negative (TN) represents a background pixel that is correctly identified as background. On the other hand, a false positive (FP) and a false negative (FN) refer to a wrong prediction for pixels in the target and background, respectively. Most of the evaluation metrics are based upon the number of pixels in these four categories.

Sensitivity measures the completeness of positive predictions with regard to the positive ground truth (TP + FN). It thus shows the model's ability to identify target pixels. It is also referred to as recall or true-positive rate (TPR). It is defined as

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{14}$$

As the negative counterpart of sensitivity, specificity describes the proportion of negative pixels that are correctly predicted. It is also referred to as true-negative rate (TNR). It is defined as

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{15}$$

Specificity can be difficult to interpret because TN is usually very large. It can even be misleading as TN can be made arbitrarily large by changing the field of view. This is due to the fact that the metric is computed over pixels and not over patients/controls like in classification tasks (the number of controls is fixed). In order to provide meaningful measures of specificity, it is preferable to define a background region that has an anatomical definition (for instance, the brain mask from which the target is subtracted) and does not include the full field of view of the image.

Positive predictive value (PPV), also known as precision, measures the correct rate among pixels that are predicted as positives:

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{16}$$

For clinical interpretation of segmentation, it is often useful to have a more direct estimation of false negatives. To that purpose, one can report the false discovery rate:

$$\text{FDR} = 1 - \text{PPV} = \frac{\text{FP}}{\text{TP} + \text{FP}} \tag{17}$$

which is redundant with PPV but may be more intuitive for clinicians in the context of segmentation.

Dice similarity coefficient (DSC) measures the proportion of spatial overlap between the ground truth (TP+FN) and the predicted positives (TP+FP). Dice similarity is the same as the F_1 score, which computes the harmonic mean of sensitivity and PPV:

$$\mathrm{DSC} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \tag{18}$$

Accuracy is the ratio of correct predictions:

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{19}$$

As was the case in specificity, we note that there are many segmentation tasks where the target anatomical structure is very small (e.g., subcortical structures); hence, the foreground and background have unbalanced number of pixels. In this case, accuracy can be misleading and display high values for poor segmentations. Moreover, as for the case of specificity, one needs to define a background region in order for TN, and thus accuracy, not to vary arbitrarily with the field of view.

The Jaccard index (JI), also known as the intersection over union (IoU), measures the percentage of overlap between the ground truth and positive prediction relative to the union of the two:

$$\mathrm{JI} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{20}$$

JI is closely related to the DSC. However, it is always lower than the DSC and tends to penalize more severely poor segmentations.

There are also distance measures of segmentation accuracy which are especially relevant when the accuracy of the boundary is critical. These include the average symmetric surface distance (ASSD) and the Hausdorff distance (HD). Suppose the surface of the ground truth and the predicted segmentation are $\mathcal{S}$ and $\mathcal{S}'$, respectively. For any point $\boldsymbol{p} \in \mathcal{S}$, the distance from $\boldsymbol{p}$ to surface $\mathcal{S}'$ is defined by the minimum Euclidean distance:

$$d(\boldsymbol{p}, \mathcal{S}') = \min_{\boldsymbol{p}' \in \mathcal{S}'} \|\boldsymbol{p} - \boldsymbol{p}'\|_2 \tag{21}$$

Then the average distance between $\mathcal{S}$ and $\mathcal{S}'$ is given by averaging over $\mathcal{S}$:

$$d(\mathcal{S}, \mathcal{S}') = \frac{1}{N_S} \sum_{i=1}^{N_S} d(\boldsymbol{p}_i, \mathcal{S}') \tag{22}$$

Note that $d(\mathcal{S}, \mathcal{S}') \neq d(\mathcal{S}', \mathcal{S})$. Therefore, both directions are included in ASSD so that the mean of the surface distance is symmetric:

$$\mathrm{ASSD} = \frac{1}{N_S + N_{S'}} \left[\sum_{i=1}^{N_S} d(\boldsymbol{p}_i, \mathcal{S}') + \sum_{j=1}^{N_{S'}} d(\boldsymbol{p}'_j, \mathcal{S}) \right] \tag{23}$$

The ASSD tends to obscure localized errors when the segmentation is decent at most of the points on the boundary. The Hausdorff distance (HD) can better represent the error, by, instead of

computing the average distance to a surface, computing the maximum distance. To that purpose, one defines

$$h(\mathcal{S}, \mathcal{S}') = \max_{\boldsymbol{p} \in \mathcal{S}} d(\boldsymbol{p}, \mathcal{S}') \tag{24}$$

Note that, again, $h(\mathcal{S}, \mathcal{S}') \neq h(\mathcal{S}', \mathcal{S})$. Therefore, both directions are included in HD so that the distance is symmetric:

$$\mathrm{HD} = \max\left(h(\mathcal{S}, \mathcal{S}'), h(\mathcal{S}', \mathcal{S})\right) \tag{25}$$

HD is more sensitive than ASSD to localized errors. However, it can be too sensitive to outliers. Hence, using the 95th percentile rather than the maximum value for computing $h(\mathcal{S}, \mathcal{S}')$ is a good option to alleviate the problem.

Moreover, there are some volume-based measurements that focus on correctly estimating the volume of the target structure, which is essential for clinicians since the size of the tissue is an important marker in many diseases. Denote the ground truth volume as V while the prediction volume as V'. There are a few expressions for the volume difference. (1) The unsigned volume difference: $|V' - V|$. (2) The normalized unsigned difference: $\frac{|V' - V|}{V}$. (3) The normalized signed difference: $\frac{V' - V}{V}$. (4) Pearson's correlation coefficient between the ground truth volumes and the predicted volumes: $\frac{\mathrm{Cov}(V, V')}{\sqrt{\mathrm{Var}(V)}\sqrt{\mathrm{Var}(V')}}$. Nevertheless, note that, while they are useful, these volume-based metrics can also be misleading (a segmentation could be wrongly placed while providing a reasonable volume estimate) when used in isolation. They thus need to be combined with overlap metrics such as Dice.

Finally, some recent guidelines on validation of different image analysis tasks, including segmentation, were published in [21].

2.1.6 Pre-processing for Segmentation Tasks

Image pre-processing is a set of sequential steps taken to improve the data and prepare it for subsequent analysis. Appropriate image pre-processing steps often significantly improve the quality of feature extraction and the downstream image analysis. For deep learning methods, they can also help the training process converge faster and achieve better model performance. The following sections will discuss some of the most widely used image pre-processing techniques.

Skull Stripping Many neuroimaging applications often require preliminary processing to isolate the brain from extracranial or non-brain tissues from MRI scans, commonly referred to as skull stripping. Skull stripping helps reduce the variability in datasets and is a critical step prior to many other image processing algorithms such as registration, segmentation, or cortical surface reconstruction. In literature, skull stripping methods are broadly classified into five categories: mathematical morphology-based methods

[22], intensity-based methods [23], deformable surface-based methods [24], atlas-based methods [25], and hybrid methods [26]. Recently, deep learning-based skull stripping methods have been proposed [27–32] to improve the accuracy and efficiency. A detailed discussion of the merits and limitations of various skull stripping techniques can be found in [33].

Bias Field Correction The bias field refers to a low-frequency and very smooth signal that corrupts MR images [34]. These artifacts, often described as shading or bias, can be generated by imperfections in the field coils or by magnetic susceptibility changes at the boundaries between anatomical tissue and air. This bias field can significantly degrade the performance of image processing algorithms that use the image intensity values. Therefore, a pre-processing step is usually required to remove the bias field. The N4 bias field correction algorithm [35] is one of the most widely used methods for this purpose, as it assumes a simple parametric model and does not require tissue classification.

Data Harmonization Another challenge of MRI data is that it suffers from significant intensity variability due to several factors such as variations in hardware, reconstruction algorithms, and acquisition settings. This is also due to the fact that most MR imaging sequences (e.g., T1-weighted, T2-weighted) are not quantitative (the voxel values can only be interpreted relative to each other). Such differences can often be pronounced in multisite studies, among others. This variability can be problematic because intensity-based models may not generalize well to such heterogeneous datasets. Any resulting data can suffer from significant biases caused by acquisition details rather than anatomical differences. It is thus desirable to have robust data harmonization methods to reduce unwanted variability across sites, scanners, and acquisition protocols. One of the popular MRI harmonization methods is a statistical approach named the combined association test (comBat). This method was shown to exhibit a good capacity to remove unwanted site biases while preserving the desired biological information [36]. Another popular method is a deep learning-based image-to-image translation model, CycleGAN [37]. The CycleGAN and its variants do not require paired data, and thus the training process is unsupervised in the context of data harmonization.

Intensity Normalization Intensity normalization is another important step to ensure comparability across images. In this section, we discuss common intensity normalization techniques. Readers can refer to the work [38] in which the author explores the impact of different intensity normalization techniques on MR image synthesis.

Z-Score Normalization The basic Z-score normalization on the entire image is also called the whole-brain normalization. Given the mean μ and standard deviation σ from all voxels in a brain mask B, Z-score normalization can be performed for all voxels in image I as follows:

$$I_{z-score}(x) = \frac{I(x) - \mu}{\sigma} \tag{26}$$

While straightforward to implement, whole-brain normalization is known to be sensitive to outliers.

White Stripe Normalization White stripe normalization [39] is based on the parameters obtained from a sample of normal-appearing white matter (NAWM) and is thus robust to local intensity outliers such as lesions. The NAWM is obtained by smoothing the histogram of the image I and selecting the mode of the distribution. For T1-weighted MRI, the "white stripe" is defined as the 10% of intensity values around the mean of NAWM μ. Let $F(x)$ be the CDF of the specific MR image $I(x)$ inside the brain mask B, and $\tau = 5\%$. The white stripe Ω_τ is defined as

$$\Omega_\tau = \{I(x) | F^{-1}(F(x) - \tau) < I(x) < F^{-1}(F(x) + \tau)\} \tag{27}$$

Then let σ_τ be the sample standard deviation associated with Ω_τ. The white stripe normalized image is

$$I_{ws}(x) = \frac{I(x) - \mu}{\sigma_\tau} \tag{28}$$

Compared to the whole-brain normalization, the white stripe normalization may work better and have better interpretation, especially for applications where intensity outliers such as lesions are expected.

Segmentation-Based Normalization Segmentation-based normalization uses a segmentation of a specified tissue, such as the cerebrospinal fluid (CSF), gray matter (GM), or white matter (WM), to normalize the entire image to the mean of the tissue. Let $T \subset B$ be the tissue mask for image I. The tissue mean can be calculated as $\mu = \frac{1}{|T|}\sum_{t \in T} I(t)$ and the segmentation-based normalized image is expressed as

$$I_{seg}(x) = \frac{cI(x)}{\mu} \tag{29}$$

where $c \in \mathbb{R}^+$ is a constant.

Kernel Density Estimate Normalization Kernel density estimate (KDE) normalization estimates the empirical probability density function of the intensities of the entire image I over the brain

mask B via kernel density estimation. The KDE of the probability density function for the image intensities can be expressed as

$$\hat{p}(x) = \frac{1}{\mathrm{HWD} \times \delta} \sum_{i=1}^{\mathrm{HWD}} K\left(\frac{x - x_i}{\delta}\right) \tag{30}$$

where H, W, D are the image sizes of I, x is an intensity value, K is the kernel, and δ is the bandwidth parameter which scales the kernel. With KDE normalization, the mode of WM can be selected more robustly via a smooth version of the histogram and thus is more suitable to be used in a segmentation-based normalization method.

Spatial Normalization Spatial normalization aims to register a subject's brain image to a common space (reference space) to allow comparisons across subjects. When the reference space is a standard space, such as the Montreal Neurological Institute (MNI) space [40] or the Talairach and Tournoux atlas (Talairach space), the registration also facilitates the sharing and interpretation of data across studies. It is also common practice to define a customized space from a dataset rather than using a standard space. For deep learning methods, it has been shown that training data with appropriate spatial normalization tend to yield better performances [41–43]. Rigid, affine, or deformable registration may be desirable for spatial normalization, depending on the application. Many registration methods are publicly available through software packages such as 3D Slicer, FreeSurfer [https://surfer.nmr.mgh.harvard.edu/], FMRIB Software Library (FSL) [https://fsl.fmrib.ox.ac.uk/fsl/fslwiki], and Advanced Normalization Tools (ANTs) [https://picsl.upenn.edu/software/ants/].

2.2 Supervision Settings

In the following three sections, we categorize the learning-based segmentation algorithms by their supervision setting. In the reverse order of the amount of annotation required, these include supervised, semi-supervised, and unsupervised methods (Fig. 7). For supervised methods, we mainly present some training strategies and model architectures that will help improve the segmentation performance. For the other two types of approaches, we classify the mainstream ideas and then provide application examples proposed in recent research.

2.3 Supervised Methods

2.3.1 Background

In supervised learning, a model is presented with the given dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ of inputs x and associated labels y. This y can take several forms, depending on the learning task. In particular, for fully convolutional neural network-based segmentation applications, y is a segmentation map. In supervised learning, the model can learn from labeled training data by minimizing the loss function

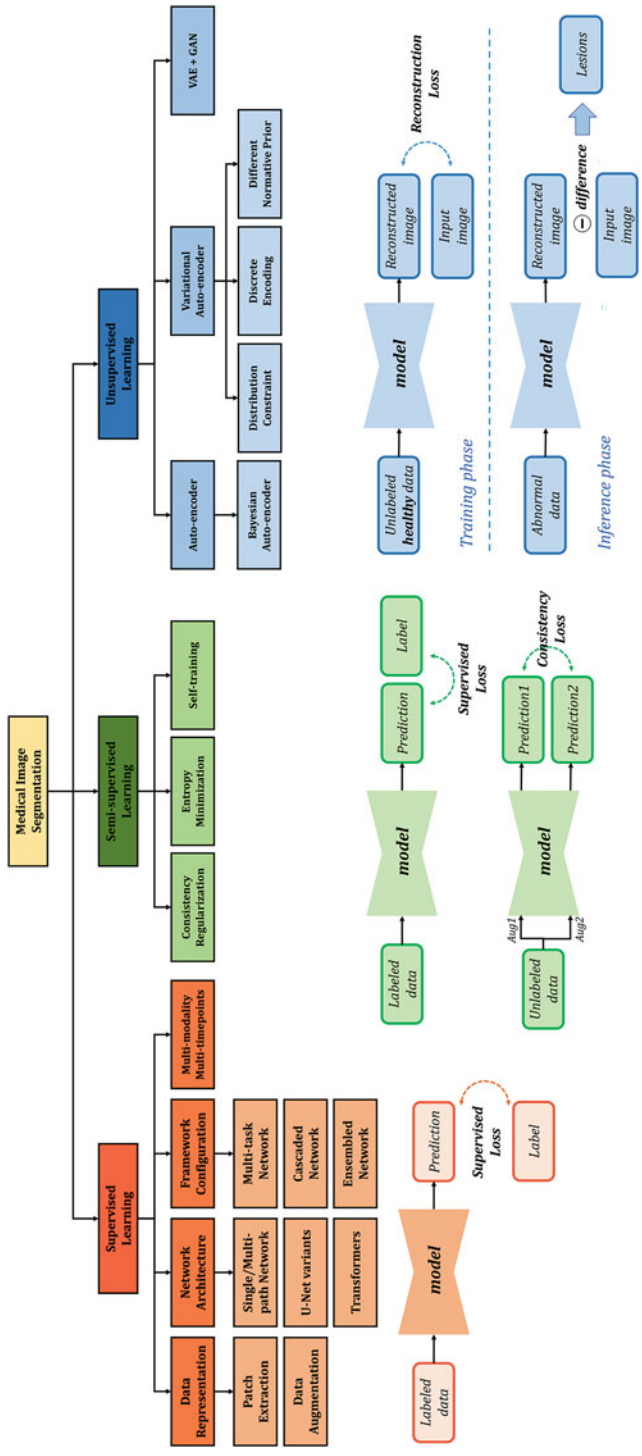

Fig. 7 Overview of the supervision settings for medical image segmentation. Best viewed in color

and apply what it has learned to make a prediction/segmentation in testing data. Supervised training thus aims to find model parameters θ that best predict the data based on a loss function $L(y, \hat{y})$. Here, $\hat{y}$ denotes the output of the model obtained by feeding a data point x to the function $f(x;\theta)$ that represents the model. Given sufficient training data, supervised methods can generally perform better than semi-supervised or unsupervised segmentation methods.

2.3.2 Data Representation

Data is an important part of supervised segmentation models, and the model performance relies on data representation. In addition to image pre-processing (Subheading 2.1.6), there are a few key steps for data preparation before being fed into the segmentation network.

Patch Formulation The inputs of CNN can be represented as image patches when the whole image is too large and would require too much GPU memory. The image patches could be 2D slices, 3D patches, and any format in between. The choice of patches would affect the performance of networks for a given dataset and task [44]. Compared to 3D patches, 2D slices have the advantage of lighter computational load during training. However, contextual information along the third axis is missing. In contrast, 3D patches leverage data from all three axes, but they require more computational resources. As a compromise between 2D and 3D patches, "2.5D" approaches have been proposed, by taking 2D slices in all three orthogonal views through the same voxel [45]. Those 2D slices could be trained in a single CNN or a separate CNN for each view. Furthermore, Zhang et al. [46] proposed 2.5D stacked slices to leverage the information from adjacent slices in each view.

Patch Extraction Due to the imbalance between foreground and background, various patch extraction strategies have been designed to obtain robust segmentation. Kamnitsas et al. [47], Dolz et al. [48], and Li et al. [49] pick a voxel within the foreground or background with 50% probability at every iteration during training and select the patch centered at that voxel. In [46], Zhang et al. extract 2.5D stacked patches if the central slice contains the foreground, even with only one voxel. In some models [50, 51], 3D patches with target structure are used as input instead of the whole image, which could reduce the effect of the background for segmenting target structures with smaller volume.

Data Augmentation To avoid the over-fitting problem and increase the generalizability of the model, data augmentation (DA) is widely used in medical image segmentation [52]. The common DA strategies could be classified into three categories:

(1) spatial augmentation, (2) image appearance augmentation, and (3) image quality augmentation. For spatial augmentation, random image flip, rotation, scale, and deformation are often used [4, 45, 53–55]. Random gamma correction, intensity scale, and intensity shift are the common forms for image appearance augmentation [51, 54, 56, 57]. Image quality augmentation includes random Gaussian blur, random noise addition, and image sharpening [51, 56]. Note that while we only list a few commonly used methods here, many others have been explored. TorchIO [58] is a widely used software package for data augmentation.

2.3.3 Network Architecture

Here, we classify the popular supervised segmentation networks into single/multipath networks and encoder-decoder networks.

Single/Multipath Networks As discussed above, patches are often used as input instead of the entire image, resulting in a lack of global context. This could produce noisy segmentations, such as undesired islands of false-positive voxels that need to be removed in post-processing [48]. To compensate for the missing global context, Li et al. [49] used spatial coordinates as additional channels of input patches. A multipath network is another feasible solution (Fig. 8). Multipath networks usually contain global and local paths [47, 59, 60] that extract different features at different scales. The global path uses convolutions with larger kernel size [60] or a larger receptive field [47] to learn global information [47]. In

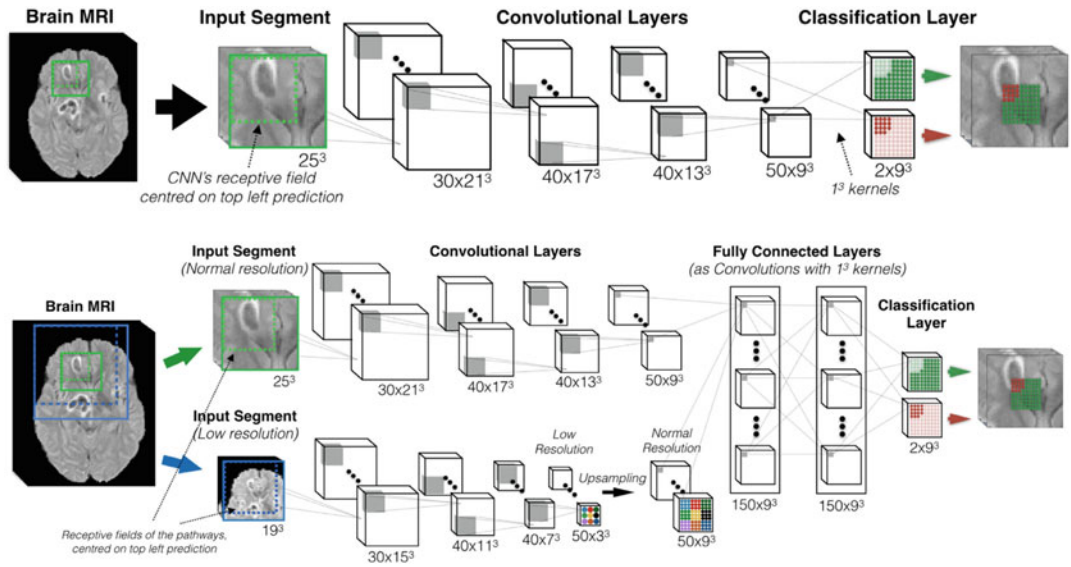

Fig. 8 Examples of single-path (top) and multipath (bottom) networks. In the multipath network, the inputs for the two pathways are centered at the same location. The top pathway is equivalent to the single-path network and takes the normal resolution image as input, while the bottom pathway takes a downsampled image with larger field of view as input. Replicated from [47] (CC BY 4.0)

contrast, local features are extracted in the local path. The global path thus extracts global features and tends to locate the position of the target structure. In contrast, the shape, size, texture, boundary, and other details of the target structure are identified by the local path. However, the performance of this type of network is easily affected by the size and design of input patches: for example, too small patches would not provide enough information, while too large patches would be computationally prohibitive.

U-Net and Its Variants To tackle the limitations of the single/multipath networks, many models use U-net variants with encoder-decoder paths [1, 61], which establishes end-to-end training from image to segmentation map. The encoder is similar to the single/multipath networks but with downsampling operations between the different scales of feature maps. The decoder leverages the extracted features from the encoder and produces a segmentation of the same size as the original image. Skip connections that pass the feature maps from the encoder directly to the decoder contribute to the performance of the U-net. The passed information could help to recover the details of segmentation.

The most common modification of the U-Net is the introduction of other convolutional modules, such as *residual blocks* [62], *dense blocks* [63], *attention modules* [3, 4], etc. These convolutional modules could replace regular convolution operations or be used in the skip connections of the U-Net. Residual blocks could mitigate the gradient vanishing problem during training by adding the input of the module to its output, which also contributes to the speed of convergence [62]. In this configuration, the network can be built deeper. The work of [53, 59, 64–66] used residual connections or residual blocks instead of regular convolutions in their network architecture for robust segmentation of various brain structures. Dense blocks could strengthen feature propagation and encourage feature reuse to improve segmentation accuracy. However, they require more computational resources during training. Zhang et al. [46, 56] employed the Tiramisu network [67], a densely U-shaped network, to produce superior multiple sclerosis (MS) lesion segmentation.

The attention module is another commonly used tool in segmentation to focus on salient features [4]. It can be categorized into spatial attention and channel attention modules. Li et al. [53] use spatial attention modules in the skip connections for extracting smaller subcortical structures. Similarly, attention modules are used between skip connections and in the decoder part in the work of [51, 68] for segmenting vestibular schwannoma and cochlea. In addition, Zhang et al. [69] proposed to use slice-wise attention networks in 3D CNNs for MS segmentation. Applying the slice-

wise attention in three different orientations improves the computational efficiency compared to the regular attention module. Hou et al. [70] proposed the cross-attention block, which combines channel attention and spatial attention. Moreover, in [71], a skip attention unit is used for brain tumor segmentation. Zhou et al. [72] build fusion blocks based on the attention module. Attention modules have also been used for brain tumor segmentation [73].

Transformers As discussed in Subheading 2.1.2, transformers have become popular in medical image segmentation [74–76]. Transformers leverage the long-range dependencies and can better capture low-level details. In practice, they can replace CNNs [77], be combined with CNNs [78, 79], or integrated into CNNs [80]. Some recent works [14, 15, 77] have shown that the implementation of transformer on U-Net architecture can achieve superior performance in medical image segmentation compared to their CNN counterparts.

2.3.4 Framework Configuration

The single network mainly focuses on a single task during training and may ignore other potentially useful information. To improve the segmentation accuracy, frameworks with multiple encoders and decoders have been proposed [53, 81, 82].

Multi-task Networks As the name suggests, multi-task networks attempt to simultaneously tackle a main task as well as auxiliary tasks, rather than focusing on a single segmentation task. These networks usually contain a shared encoder and multiple decoders for multiple tasks, which could help deal with class imbalance (Fig. 9). Compared to a single-task network, the learning ability of the encoder is increased from same domain tasks (e.g., multiple tasks of multiple decoders), which could improve segmentation performance. Simultaneously learning multiple tasks could also improve model generalizability. McKinley et al. [81] leverage the information of additional tissue types to increase the accuracy of MS lesion segmentation. Another common multi-task setting is to introduce an auxiliary reconstruction task [57].

Cascaded Networks A cascaded network is a series of connected networks such that the input of each downstream network is the output from an upstream network (Fig. 10). For example, a coarse-to-fine segmentation strategy can be used to reduce the high computational cost of training for 3D images [50, 53]. In this scenario, an upstream network could take downsampled images as input to roughly locate the target structures, allowing the images to be cropped to the region of interest for the downstream network. The downstream network could then produce high-quality segmentation in full resolution. Another advantage of this approach

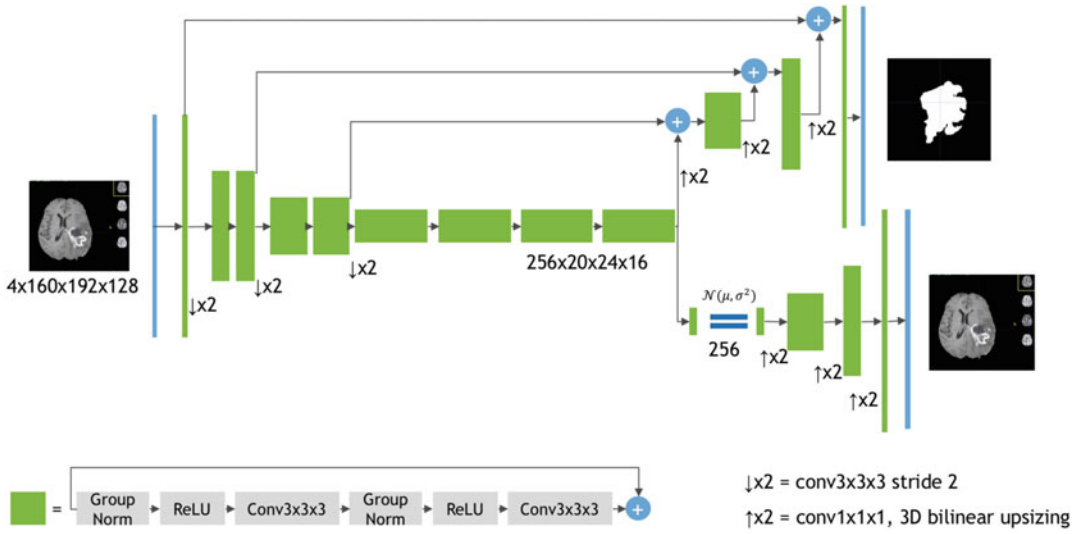

Fig. 9 Example of multi-task framework. The model takes four 3D MRI sequences (T1w, T1c, T2w, and FLAIR) as input. The U-Net structure (the top pathway with skip connection) serves as the segmentation network, and the output contains the segmentation maps of the three subregions (whole tumor (WT), tumor core (TC), and enhancing tumor (ET)). An auxiliary VAE branch (the bottom decoder) that reconstructs the input images is applied in the training stage to regularize the shared encoder. ©2019 Springer Nature. Reprinted, with permission, from [57]

is to reduce the impact of volume imbalance between foreground and background classes. However, the upstream network would determine the performance of the whole framework, and some global information is missing in the downstream networks.

Ensemble Networks To obtain a robust segmentation, a popular approach is to aggregate the output from multiple independent networks (i.e., no weights/parameters shared). Kanitsas et al. proposed the ensemble of multiple models and architectures (EMMA) [83] for brain tumor segmentation. Kao et al. [84] produce segmentation using 26 ensemble neural networks. Zhao et al. [85] proposed a framework for 3D segmentation with multiple 2D networks that take input from different views. Huo et al. [82] proposed the spatially localized atlas network tiles (SLANT) method to distribute multiple networks for 3D high-resolution whole-brain segmentation. Among their variants, SLANT-27 (Fig. 11), which ensembles 27 networks, produces the best result. Last but not least, many medical image segmentation challenge participants use model ensembling to achieve high performance.

2.3.5 Multiple Modalities and Timepoints

Many neuroimaging studies contain multiple modalities or multiple timepoints per subject. This additional information is clearly valuable and can be leveraged to improve segmentation performance.

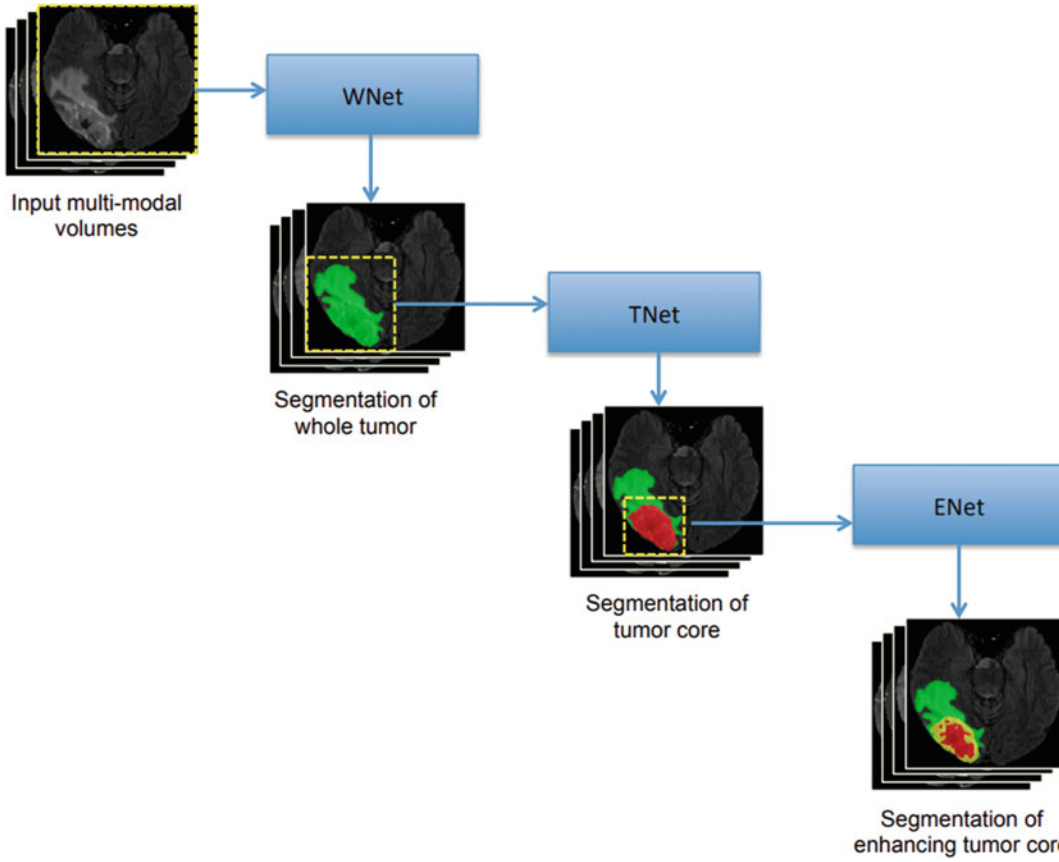

Fig. 10 Example of cascaded networks. WNet segments the whole tumor from the input multimodal 3D MRI. Then based upon the segmentation, a bounding box (yellow dash line) can be obtained and used to crop the input. The TNet takes the cropped image to segment the tumor core. Similarly, the ENet segments the enhancing tumor core by taking the cropped images determined by the segmentation from the previous stage. ©2018 Springer Nature. Reprinted, with permission, from [50]

Multiple Modalities Different imaging modalities offer different visualizations of various tissue types. Multi-modality datasets can be thus leveraged to improve segmentation accuracy. For example, Zhang et al. [86] proposed a framework with two independent networks that take two different modalities as inputs. Instead of combining single modality networks, Zhang et al. [46] concatenate multi-modality data as different channels of inputs. However, not all modalities are available in clinical practice: (1) the MRI sequences can vary between different imaging sites and (2) some modalities may be unusable due to poor image quality. This is known as the missing modality problem. To tackle this problem, Havaei et al. [87] proposed a deep learning method that is robust to missing modalities for brain tumor and MS segmentation, which contains an abstraction layer that transforms feature maps into statistics to help learning during training. In [88], the authors further improved modality dropout by introducing dynamic filters

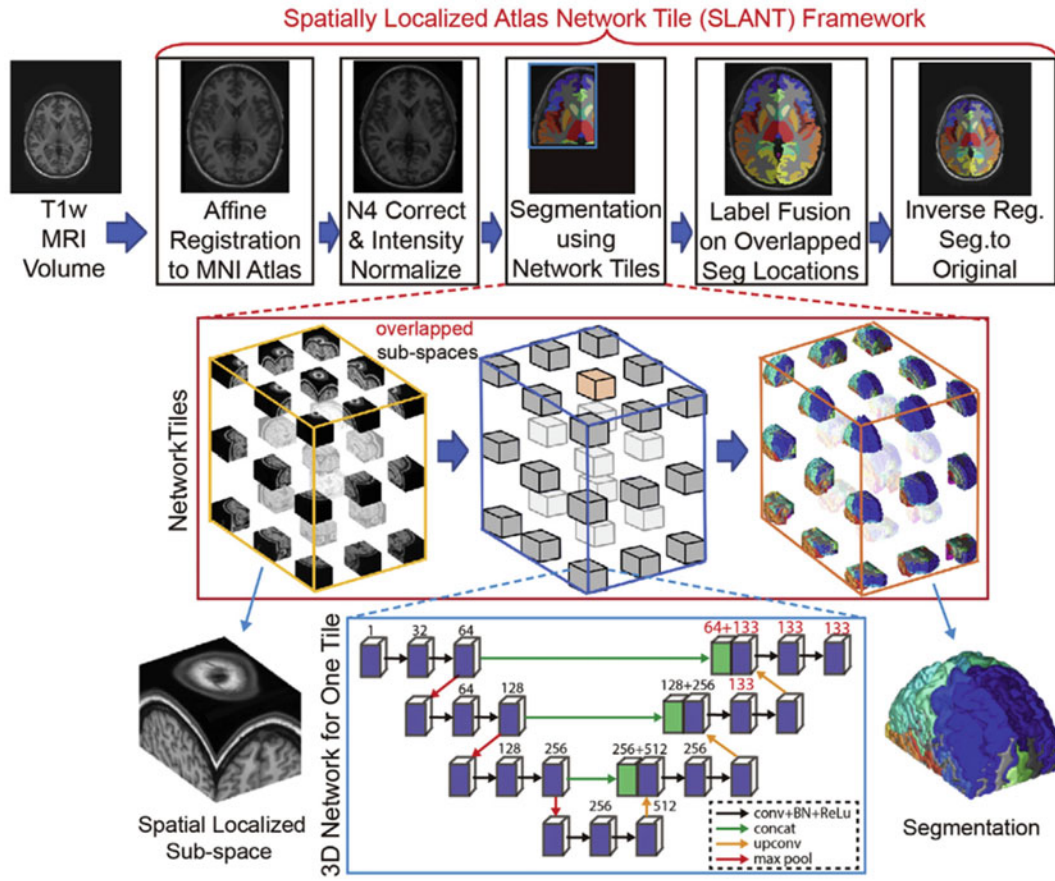

Fig. 11 SLANT-27: An example of ensemble networks. The whole brain is split into 27 overlapping subspaces with regard to their spatial locations (yellow cube). For each location, there is an independent 3D fully convolutional network (FCN) for segmentation (blue cube). The ensemble is achieved by label fusion on overlapping locations. ©2019 Elsevier. Reprinted, with permission, from [82]

and co-training strategy for MS lesion segmentation. In [89, 90], the authors used knowledge distillation scheme to transfer the knowledge from full-modality data to each missing condition with individual models.

Multiple Timepoints Data from multiple timepoints are important for tracking the longitudinal changes in a single subject. The additional timepoints can also be used as temporal context to improve the segmentation for each timepoint. In [45], longitudinal data are concatenated as a multichannel input to improve segmentation. In the work of [91], the stacked convolutional long short-term memory modules (C-LSTMs) are integrated into CNN for 4D medical image segmentation, which allows the model to learn the correlation and overall trends from longitudinal data. Li et al. [92] also proposed a framework with C-LSTM modules for segmenting longitudinal data jointly.

2.4 Semi-supervised Methods

2.4.1 Background

Given a considerable amount of labeled data, deep learning-based methods have achieved state-of-the-art performances in various medical image analysis applications. However, it is a laborious and time-consuming process to obtain dense pixel/voxel-level annotations for segmentation tasks. Since accurate annotations require expertise in medical domain, they are also expensive to collect. It is therefore desirable to leverage unlabeled data alongside the labeled data to improve model performance, an approach typically known as semi-supervised learning (SSL). Intuitively, these unlabeled data can provide critical information on the data distribution and thus can be used to improve model robustness by exploring this distribution.

Conceptually, SSL falls in between supervised learning (fully labeled data) and unsupervised learning (no labeled data). In SSL, we have access to both a labeled dataset $\mathcal{D}_L = \{(x_l^{(i)}, y_l^{(i)}) | i = 1, 2, \cdots, n_l\}$, where $y_l^{(i)}$ is the ith manually annotated ground truth mask in the context of segmentation task, and an unlabeled dataset $\mathcal{D}_U = \{(x_u^{(i)} | i = 1, 2, \cdots, n_u\}$. Typically, $n_u \gg n_l$. The main objective of SSL is to train a segmentation network X by leveraging both $\mathcal{D}_L$ and $\mathcal{D}_U$ to surpass the performances achieved by solely supervised learning with $\mathcal{D}_L$ or unsupervised learning with $\mathcal{D}_U$.

According to [93], there are mainly three underlying assumptions held by SSL: (1) smoothness assumption, (2) low-density assumption, and (3) cluster assumption. The smoothness assumption states that the data points that are close by in the input or latent space should have similar or identical labels. With this assumption, we can expect the labels of unlabeled data to be similar to those of labeled data when these samples are similar in input or latent space, i.e., the labels from the labeled dataset can be transferred to the unlabeled dataset. In the low-density assumption, we assume that the decision boundary of a classifier should ideally not pass through the high density of the marginal data distribution. Placing the decision boundary in a high-density region would violate the smoothness assumption because the labels would be more likely to be dissimilar for similar data points. Lastly, the cluster assumption states that each cluster of data points should belong to the same class. This assumption is necessary because if the data points from the unlabeled and labeled datasets cannot be meaningfully clustered, the unlabeled data cannot be used to improve the model performance trained from only the labeled data.

2.4.2 Overview of Semi-supervised Techniques

In the semi-supervised learning literature, most of the techniques are originally designed and validated in the context of classification tasks. However, these methods can be readily adapted to segmentation tasks since a segmentation task can be viewed as pixel-wise classification. In this chapter, we mainly categorize the SSL approaches into three techniques, namely, (1) consistency

Table 1
Summary of classic semi-supervised learning methods

Method	Consistency regularization	Entropy minimization	Self-training
Pseudo-label [94]	No	Yes	Yes
Π model [95]	Yes	No	Yes
Temporal ensembling [95]	Yes	No	Yes
Mean teacher [96]	Yes	No	No
UDA [97]	Yes	Yes	No
MixMatch [98]	Yes	Yes	No
FixMatch [99]	Yes	Yes	No

regularization, (2) entropy minimization, and (3) self-training. However, most existing SSL approaches often employ a combination of these techniques rather than a single one, as summarized in Table 1. In the following sections, we will discuss each approach in detail and introduce some of the most important SSL techniques alongside.

2.4.3 Consistency Regularization

In semi-supervised learning, consistency regularization has been widely used as a technique to make use of unlabeled data. The idea of consistency regularization is based on the smoothness assumption that the network outputs should remain the same even if the input data is perturbed slightly (i.e., do not vary dramatically in the input space). The consistency between the predictions of an unlabeled sample and its perturbed counterpart can be used as a supervision mechanism for training to leverage the unlabeled data. In such scenarios, we can formulate the semi-supervised training objective as follows:

$$\ell_{SSL} = \sum_{x_l, y_l \in D_L} L_S(x_l, y_l) + \alpha \sum_{x_u \in D_U} L_C(x_u, \tilde{x}_u) \tag{31}$$

where L_S is the supervised loss for labeled data. For segmentation tasks, L_S can be one of the segmentation losses we presented in Subheading 2.1.3. x_u and $\tilde{x}_u$ are the unlabeled data and its perturbed version, respectively. L_C is the consistency loss function. Mean squared error loss and KL divergence loss have been widely used as L_C in the SSL literature. α is a balancing term to weigh the impact of consistency loss from unlabeled data.

It is worth noting that the random permutations involved in consistency regularization can be implemented in different ways. For instance, the Π model [95] encourages consistent network outputs between two versions of the same input data, i.e., with different data augmentation and different network dropout

conditions. In this way, training can leverage the labeled data by optimizing the supervised segmentation loss and the unlabeled data by using this unsupervised consistency loss. In mean teacher [96], the authors propose to compute the consistency between the outputs of the student network and the teacher network (which uses the exponential moving average of the student network weights) from the same input data. In unsupervised data augmentation (UDA) [97], unlabeled data are augmented via different augmentation strategies such as RandAugment [100] and are fed to the same network to obtain two model predictions, which are used to compute the consistency loss. Similarly, in MixMatch [98], another very popular SSL method, an unlabeled image is augmented K times and the average of their outputs is sharpened, which is then used as the supervision signal to compute the consistency loss. Moreover, in FixMatch [99], the consistency loss is computed on the weakly and strongly augmented versions of the same input. In summary, consistency regularization has been widely used in various SSL techniques to leverage the unlabeled data.

Application: MTANS MTANS [101] is an SSL framework for brain lesion segmentation. As shown in Fig. 12, the MTANS framework is built upon the mean teacher model [96] where both the teacher and the student models are used to segment the brain lesions as well as the signed distance maps of the object surfaces. As a variant of the mean teacher model, MTANS incorporates **consistency regularization** in the training strategy. Specifically, the authors propose to compute the multi-scale feature consistency as consistency regularization, while the traditional mean teacher model only computes the consistency at the output level. Besides, a discriminator network is used to extract hierarchical features and differentiate the signed distance maps obtained by labeled and unlabeled data. In experiments, MTANS is evaluated on three public brain lesion datasets including ISBI 2015 (multiple sclerosis) [102], ISLES 2015 (ischemic stroke) [103], and BRATS 2018 (brain tumor) [104]. Experimental results show that MTANS can outperform the supervised baseline and other competing SSL methods when trained with the same amount of labeled data.

2.4.4 Entropy Minimization

Entropy minimization is another important SSL technique and is often used together with consistency training. Generally, entropy is the measure of the disorder or the uncertainty of a system. In the context of SSL, this term often refers to the uncertainty in the pseudo-label obtained by the unlabeled data. Entropy minimization, also known as minimum entropy regularization, aims to encourage the model to produce high-confidence predictions. The idea of entropy minimization is built upon the low-density assumption as it requires the network to output low-entropy

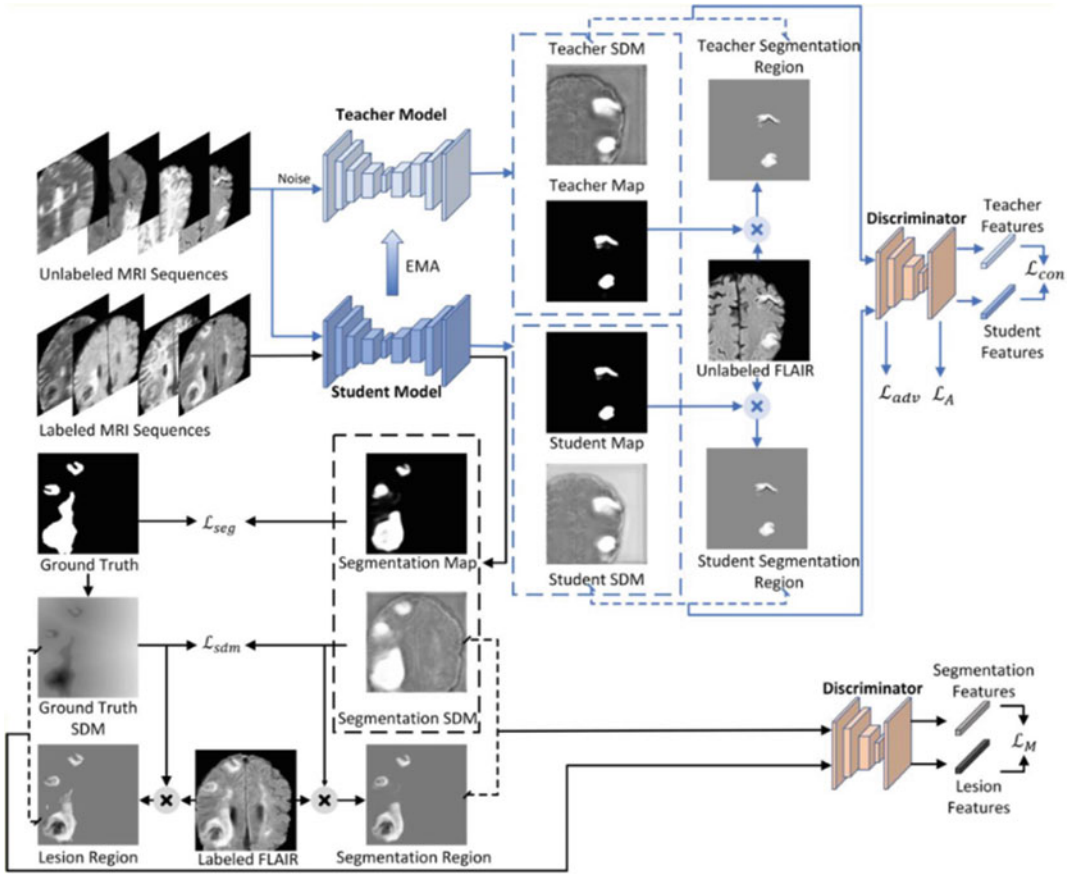

Fig. 12 An illustration of the MTANS framework. The blue solid lines indicate the path of unlabeled data, while the labeled data follows the black lines. The two segmentation models provide the segmentation map and the signed distance map (SDM). The discriminator is applied to check the consistency of the outputs from the teacher and student models. The parameters of the teacher model are updated according to the student model using the exponential moving average (EMA). ©2021 Elsevier. Reprinted, with permission, from [101]

predictions on unlabeled data. The high-confidence pseudo-labels have been found very effective when used as the supervision for unlabeled data. For example, in MixMatch, the pseudo-label of the unlabeled data, i.e., the average predictions of K augmented samples, is "sharpened" by adjusting the prediction distribution. This sharpening process is an implicit way to minimize the entropy on the unlabeled data distribution. In pseudo-label [94], the authors propose to construct the hard (one-hot) pseudo-labels from the high-confidence predictions of the unlabeled data, which is another form of entropy minimization. In addition, the UDA method proposes to compute the consistency loss only when the highest probability in the predicted class is above a pre-defined threshold. Similarly, in FixMatch, the predictions of the weakly augmented unlabeled data are first filtered by a pre-defined threshold and later converted to a one-hot pseudo-label.

2.4.5 Self-training

Self-training is an iterative training process where the network uses the high-confidence pseudo-labels of the unlabeled data from previous training steps. Interestingly, it has been shown that self-training is equivalent to a version of the classification EM algorithm [105]. The ideas of self-training and consistency regularization are very similar. Here, we differentiate these two concepts as follows: for consistency regularization, the supervision signals of the unlabeled data are generated online, i.e., from the current training epoch; in contrast, for self-training, the pseudo-labels of unlabeled data are generated offline, i.e., generated from the previous training epoch/epochs. Typically, in self-training, the pseudo-labels produced from previous epochs need to be carefully processed before being used as the supervision, as they are crucial to the effectiveness of the self-training methods. In the SSL literature, pseudo-label [94] is a representative method that uses self-training. In pseudo-label, the network is first trained on the labeled data only. Then the pseudo-labels of the unlabeled data are obtained by feeding them to the trained model. Next, the top K predictions on the unlabeled data are used as the pseudo-labels for the next epoch. The training objective function of pseudo-label is as follows:

$$L_{PL} = \sum_{x_l, y_l \in D_L} L_S(x_l, y_l) + \alpha(t) \sum_{x_u \in D_U} L_S(x_u, \tilde{y}_u) \tag{32}$$

where $\tilde{y}$ is the pseudo-label and $\alpha(t)$ is a balancing term to weigh the importance of pseudo-label training. Particularly, $\alpha(t)$ is designed to slowly increase to help the optimization process to avoid poor local minima [94]. Note that both labeled and unlabeled data are trained in a supervised manner with ground truth labels y_l and pseudo labels $\tilde{y}_u$.

Application: 4S In this study, the authors propose a sequential semi-supervised segmentation (4S) framework [106] for serial electron microscopy image segmentation. As shown in Fig. 13, 4S relies on the **self-training** strategy as it applies pseudo-labeling to all slices in the target continuous images, with only a small number of consecutive input slices. Specifically, a few labeled samples are used for the first round of training. The trained model is then used to generate pseudo-labels for the next sample. Afterward, the segmentation model is retrained using the pseudo-labels and produces new pseudo-labels for the next slices. This method was evaluated on the ISBI 2012 dataset (neural cell membranes) [107] and Japanese carpenter ant dataset (nestmate discriminant sensory elements) [108]. Results show that 4S has achieved better performance than the supervised learning-based method.

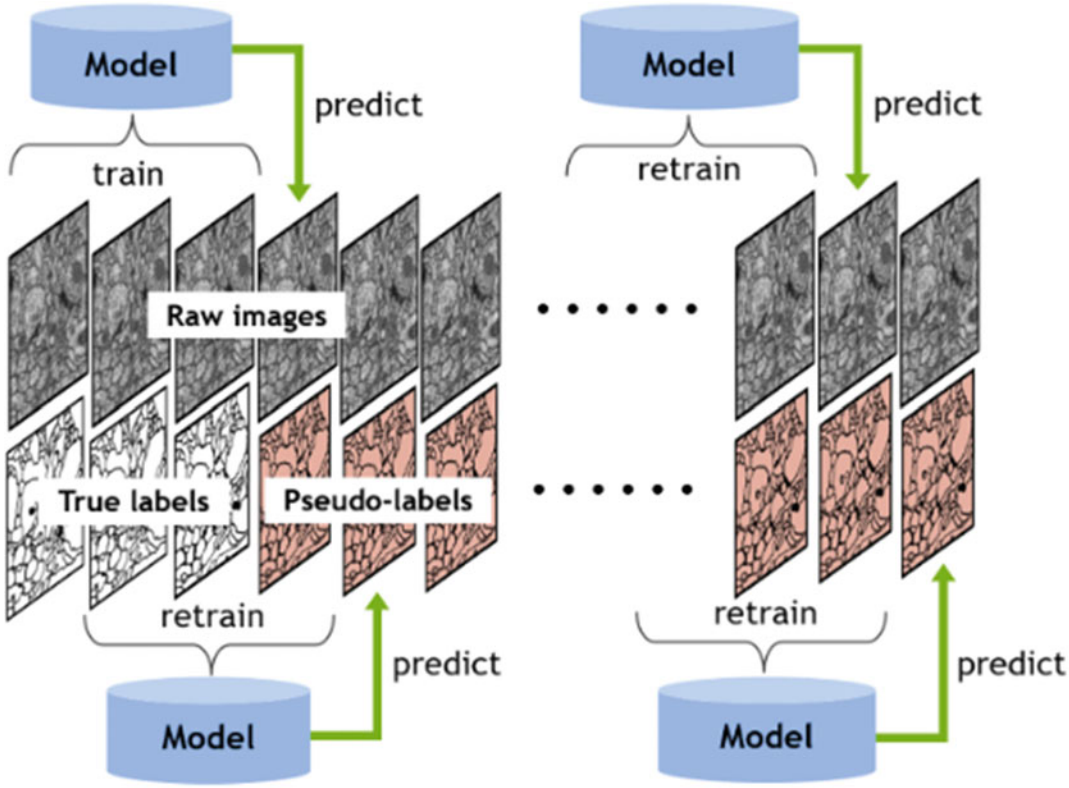

Fig. 13 The workflow of the 4S framework. Based on the assumption that consecutive images are strongly correlated, the manual annotations (true labels) are provided for the first few slices. These labeled data are used for the initial training. Then the model can provide the pseudo-labels for the next few slices which can be applied for retraining. Adapted from [106] (CC BY 4.0)

2.5 Unsupervised Methods

2.5.1 Background

As suggested in Subheadings 2.3 and 2.4, most deep segmentation models learn to map the input image x to the manually annotated ground truth y. Although semi-supervised approaches can drastically reduce the need for labels, low availability of ground truth is still a primary concern for the development of learning-based models. Another disadvantage of supervised learning approaches becomes evident when considering the anomaly detection/segmentation task: a model can only recognize anomalies that are similar to those in the training dataset and will likely fail with rare findings that may not appear in the training data [109].

Unsupervised anomaly detection (UAD) methods have been developed in recent years to tackle these problems. Since no ground truth labels are provided, the models are designed to capture the inherent discrepancy between healthy and pathological data distributions. *The general idea is to represent the distribution of normal brain anatomy by a deep model that is trained exclusively on healthy subjects* [109]. Consequently, the pathological subjects are out of

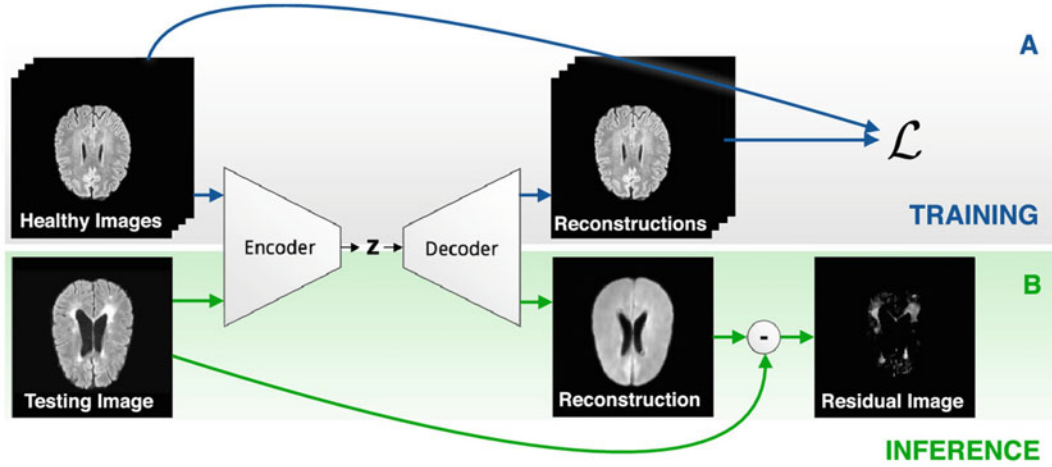

Fig. 14 The general idea of unsupervised anomaly detection (UAD) realized by an auto-encoder. (**a**) Train the model with only healthy subjects. (**b**) Test with pathological samples. The residual image depicts the anomalies. ©2021 Elsevier. Reprinted, with permission, from [109]

the distribution modeled by the network. Usually, this neural network has an encoder-decoder architecture such that the output will be a reconstruction of the input image. Since not well represented by the training data, the abnormal region cannot be fully reconstructed. Hence, the pixel-wise reconstruction error can be used as an estimate of the anomalous region. Figure 14 illustrates this process.

The auto-encoder (AE) and its variations (Fig. 15) are widely used in the UAD problem. All these models generate a low-dimensional representation of the input image termed latent vector $\boldsymbol{z}$ at the bottleneck. Most of the research concentrates on manipulating the distribution of $\boldsymbol{z}$ so that the abnormal region can be "cured" in the reconstruction. This process is often referred to as image restoration (or sometimes image inpainting) in the computer vision literature. The following sections will discuss some mainstream approaches categorized by the model structure implemented.

2.5.2 Auto-encoders

The auto-encoder (AE) (Fig. 15a) is the simplest encoder-decoder structure. Let an encoder f_θ and a decoder g_ϕ, where θ, ϕ are model parameters. Given a healthy input image $\boldsymbol{X}^h \in \mathbb{R}^{D \times H \times W}$, the encoder learns to project it to a lower-dimensional latent space $\boldsymbol{z} = f_\theta(\boldsymbol{X}^h)$, $\boldsymbol{z} \in \mathbb{R}^L$. Then the decoder recovers the original image from the latent vector as $\hat{\boldsymbol{X}}^h = g_\phi(\boldsymbol{z})$. The model is trained by minimizing the loss function $\mathcal{L}$ that delineates the difference between the input and the reconstructed image:

$$\underset{\theta,\phi}{argmin} \mathcal{L}_{\theta,\phi}(\boldsymbol{X}^h, \hat{\boldsymbol{X}}^h) = \|\boldsymbol{X}^h - \hat{\boldsymbol{X}}^h\|_n \tag{33}$$

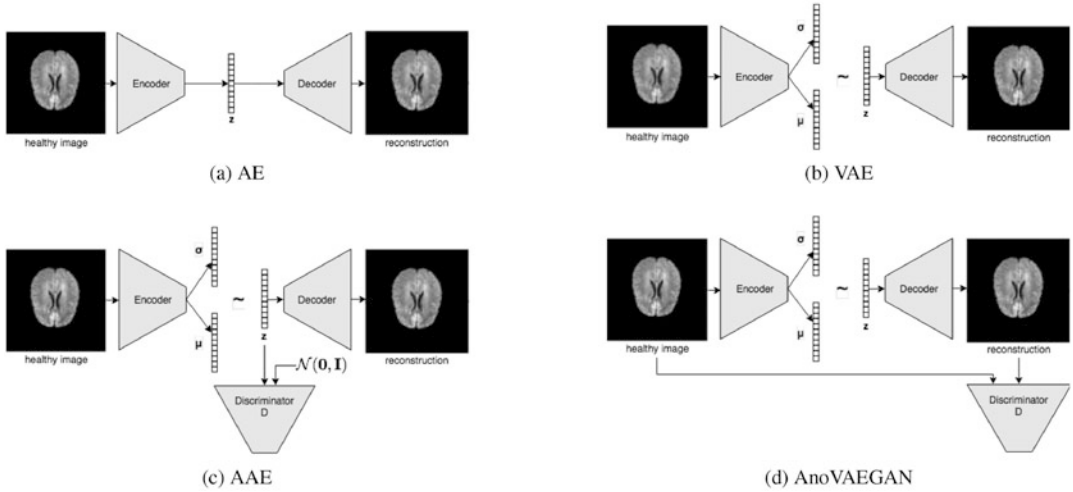

Fig. 15 Variations of auto-encoder. (**a**) The auto-encoder. (**b**) The variational auto-encoder. (**c**) The adversarial auto-encoder includes a discriminator that provides constraint on the distribution of the latent vector *z*. (**d**) Anomaly detection VAEGAN introduces a discriminator to check whether the reconstructed image lies in the same distribution as the healthy image. ©2021 Elsevier. Reprinted, with permission, from [109]

The ℓ_1-norm ($n=1$) and ℓ_2-norm (mean squared error) ($n=2$) are common choices for the loss function. The training stage is illustrated in Fig. 14a. When a sample with anomaly $\boldsymbol{X}^a$ is passed into the model, the abnormal region (e.g., lesion, tumor) cannot be well reconstructed in $\hat{\boldsymbol{X}}^a$ as the model has never seen the anomaly in the healthy training data. In other words, the AE-based methods leverage the models' dependence on training data to discern the region that is out of distribution. Figure 14b shows that the anomaly can be roughly represented by the reconstruction error $\hat{\boldsymbol{\Upsilon}} = |\boldsymbol{X}^a - \hat{\boldsymbol{X}}^a|$.

Bayesian Auto-encoder Pawlowski et al. [110] report a Bayesian convolutional auto-encoder to model the healthy data distribution. They introduce the model uncertainty and deem the reconstructed image as the Monte Carlo (MC) estimate. Let F_Θ be the auto-encoder model with weights Θ and $\mathcal{D}$ the training dataset. Then, the MC estimation can be expressed as

$$F_\Theta(\boldsymbol{X}) = \int P(\boldsymbol{X}|\Theta)P(\Theta|\mathcal{D})d\Theta \approx \frac{1}{N}\sum_{i=1}^{N} F_{\Theta_i}(\boldsymbol{X}) \tag{34}$$

where $\Theta_i \sim P(\Theta|\mathcal{D})$. In practice, the authors apply the MC-dropout to model the weight uncertainty. The segmentation is still obtained by setting a threshold on the reconstruction error, as in the vanilla auto-encoder.

2.5.3 Variational Auto-encoders

In some applications, instead of utilizing the lack of generalizability of the model, we want to modify the latent vector $\boldsymbol{z}$ to further guarantee that the reconstructed testing image $\hat{\boldsymbol{X}}^a$ looks closer to a healthy subject. Then again, the residual between $\boldsymbol{X}^a$ and $\hat{\boldsymbol{X}}^a$ is sufficient to highlight the anomalies in the image. Usually, such manipulation requires probabilistic modeling for the latent manifold. Hence, many applications use the variational auto-encoder (VAE) [111] as the backbone of the model (Fig. 15b).

As previously stated, we want the model to learn the distribution of healthy data $P(\mathbf{X}^h)$. In the encoder-decoder structure, we introduce a latent vector $\mathbf{z}$ at the bottleneck which follows a given distribution $P(\mathbf{z})$. Usually, $P(\mathbf{z})$ is assumed to follow a normal distribution $\mathcal{N}(0, \boldsymbol{I})$. The encoder and decoder are expressed by the conditional probabilities $Q_\theta(\mathbf{z}|\mathbf{X}^h)$ and $P_\phi(\mathbf{X}^h|\mathbf{z})$, respectively. Then the target distribution is given by

$$P(\mathbf{X}^h) = \int P_\phi(\mathbf{X}^h|\mathbf{z})P(\mathbf{z})d\mathbf{z}. \tag{35}$$

In addition to the reconstruction loss (e.g., ℓ_1/ℓ_2 norm), the Kullback-Leibler (KL) divergence $D_{KL}[Q_\theta(\mathbf{z}|\mathbf{X}^h)\|P(\mathbf{z})]$ that measures the distance of two distributions is another objective function to minimize. This term provides a constraint on the latent manifold such that the feature vector $\mathbf{z}$ can be stochastically sampled from a normal distribution. By modifying Eq. 13.35 and then applying Jensen's inequality, we get the evidence lower bound (ELBO) $\mathcal{L}$ for the log-likelihood of the healthy data:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{\mathbf{z}\sim Q_\theta(\mathbf{z}|\mathbf{X}^h)}[\log P_\phi(\mathbf{X}^h|\mathbf{z})] - D_{KL}[Q_\theta(\mathbf{z}|\mathbf{X}^h)\|P(\mathbf{z})] \tag{36}$$

It has been proved that maximizing the $\log P(\mathbf{X}^h)$ is equivalent to maximizing its ELBO, so $-\mathcal{L}$ serves as an objective function to optimize parameters θ and ϕ in the VAE model. By leveraging the same idea in the AE-based methods, the neural networks f_θ and g_ϕ model the normal brain anatomy if the training data contains only the healthy subjects. The approaches using VAE take one more step to guarantee the abnormal region cannot be recovered in the output, that is, modify the latent vector $\mathbf{z}^a$ of the anomalous input such that $\mathbf{z}^a \sim Q_\theta(\mathbf{z}|\mathbf{X}^h)$.

Given that healthy brains $\mathbf{X}^h$ and subjects with anomaly $\mathbf{X}^a$ are differently distributed, it is reasonable to assume that their latent manifolds $Q_\theta(\mathbf{z}|\mathbf{X}^h)$ and $Q_\theta(\mathbf{z}|\mathbf{X}^a)$ also vary. Suppose $\mathbf{z}^a = f_\theta(\mathbf{X}^a)$, then naturally, $\mathbf{z}^a \sim Q_\theta(\mathbf{z}|\mathbf{X}^a)$. If we can modify $\mathbf{z}^a$ so that $\mathbf{z}^a \sim Q_\theta(\mathbf{z}|\mathbf{X}^h)$, then after passing through the decoder $P_\phi(\mathbf{X}^h|\mathbf{z})$, the reconstruction output of the model $\hat{\mathbf{X}}^a$ would belong in $P(\mathbf{X}^h)$. That is to say, the modification in the latent manifold "cures" the anomaly. It is then easy to identify the anomaly as the residual between the input and output. The core part of the process is how to "cure" the latent representation of abnormal input. Some common ways are reported in the following examples.

Distribution Constraint A straightforward way to force $\mathbf{z}^a \sim Q_\theta(\mathbf{z}|\mathbf{X}^b)$ is adding a specific loss function at the bottleneck. Chen et al. [112] propose an adversarial auto-encoder (AAE) shown in Fig. 15c. The encoder works as a generator that produces samples in the latent space, and an additional discriminator is trained to judge whether the sample is drawn from the normal distribution. It emphasizes that all the latent representations should follow $\mathcal{N}(0, \boldsymbol{I})$, whether the input is healthy or not.

Discrete Encoding Another solution is proposed by Pinaya et al. [113]. They implement the vector-quantized variational auto-encoder (VQ-VAE) [114] to obtain a discrete representation of the latent tensor $\boldsymbol{z} \in \mathbb{R}^{n_z \times h \times w}$. It can be regarded as a $h \times w$ image which contains a vector $\boldsymbol{v}_i \in \mathbb{R}^{n_z}$ at each image location, where $i = 1, 2, \ldots, h \times w$. The quantization of $\boldsymbol{z}$ is realized by a pretrained embedding space ($\boldsymbol{e}_j \in \mathbb{R}^{n_z}$, where $j = 1, 2, \ldots, K$). It serves as a codebook from which we can always find a code $\boldsymbol{e}_j$ that is closest to the given $\boldsymbol{v}_i$. Then by simply replacing the vector $\boldsymbol{v}_i$ with the index of its closest counterpart in the codebook, a quantized latent image $\boldsymbol{z}_q \in \mathbb{R}^{h \times w}$ is obtained. Theoretically, the abnormal region is "cured" by using $\boldsymbol{e}_j$ to approximate $\boldsymbol{v}_i$ as the embedding space follows a fixed distribution. As usual, the residual between input and the reconstructed image $|\boldsymbol{X} - \hat{\boldsymbol{X}}|$ is used to find the anomaly.

Different Normative Prior Different from the vanilla VAE described above, Dilokthanakul et al. [115] propose a Gaussian mixture VAE (GMVAE) that replaces the unit multivariate Gaussian prior in the latent space with a Gaussian mixture model. GMVAE was used for brain UAD by You et al. [116]. Following the same idea of ruling out the anomaly in the latent space, they restore the image with anomaly using maximum a posteriori estimation given the Gaussian mixture model.

2.5.4 Variational Auto-encoders with Generative Adversarial Networks

A generative adversarial network (GAN) consists of two modules, a generator G and a discriminator D. Similar with the decoder in VAE, the generator G models the mapping from a latent vector to the image space $\mathbf{z} \mapsto \mathcal{X}$ where $\mathbf{z} \sim \mathcal{N}(0, \boldsymbol{I})$. The discriminator D can be deemed as a trainable loss function that judges whether the generated image $G(\mathbf{z})$ is in the image space $\mathcal{X}$. Combining the GAN discriminator and the VAE backbone has become a common idea in UAD problems. More details on GANs can be found in Chap. 5.

We note that D can be used as an additional loss in either latent or image space. In the adversarial auto-encoder (AAE) discussed above, the discriminator works to check whether the latent vector is drawn from the multivariate normal distribution. In contrast, Buar et al. [117] propose the AnoVAEGAN (Fig. 15d) model, in which the discriminator is applied in the image space to check whether the reconstructed image lies in the distribution of healthy data.

3 Medical Image Segmentation Challenges

Medical image segmentation is affected by different aspects of the specific task, such as image quality, visibility of tissue boundaries, and the variability of the target structures. Moreover, each organ, anatomical structure, or lesion type has its own specificities, and a given method may perform well for a given target and worse for another. Therefore, many public challenges are held that target specific problems in an attempt to create benchmarks and attract new researchers into an application field.

In this section, we briefly introduce some of the popular medical image segmentation challenges related to neuroimages. Then, we focus on brain tumor and multiple sclerosis (MS) segmentation challenges and summarize the most competitive methods for each challenge to highlight examples of the concepts discussed in this chapter.

3.1 Popular Segmentation Challenges

Medical image segmentation challenges aim to find better solutions to certain tasks, and it also provides researchers with benchmark or baseline methods for future development. Furthermore, the developments are driven by the need to clinical problems.

Medical Segmentation Decathlon There are ten different segmentation tasks in the medical segmentation decathlon (MSD), and each task focuses on certain organ/structure [118]. Specifically, liver tumors, brain tumors, hippocampus, lung tumors, prostate, cardiac, pancreas tumors, colon cancer, hepatic vessels, and spleen are the focused organ of each task. Each task usually involves a different modality. For example, multimodal multisite MRI data are used for brain tumors, while liver tumors are studied from portal venous-phase CT data. The Dice score (DSC) and normalized surface distance are used as evaluation metrics due their well-known behavior. Instead of finding the state-of-the-art performance for each task, MSD aims to find generalizable methods.

crossMoDA These years, domain adaptation techniques are a hot topic in medical image segmentation field, and a new challenge for unsupervised cross-modality domain adaptation is held for researchers which is named as cross-modality domain adaptation (crossMoDA) for medical image segmentation [119]. Furthermore, it is the first large and multi-class benchmark for unsupervised domain adaptation to segment vestibular schwannoma (VS) and cochleas. In a short summary, crossMoDA consists of labeled and unlabeled datasets of T1-weighted and T2-weighted MRIs (T1-w and T2-w images are unpaired). It aims to segment the corresponding regions of interest in unlabeled T2-weighted MRIs by leveraging the information from unpaired and labeled T1-weighted MRIs.

3.2 Brain Tumor Segmentation Challenge

Brain tumor segmentation (BraTS) challenge is an annual challenge held since 2012 [104, 120–123]. The participants are provided with a comprehensive dataset that includes annotated, multisite, and multi-parametric MR images. It is worth noting that the dataset has increased from 30 cases to 2000 between 2012 and 2021 [123].

Brain tumor segmentation is a difficult task for a variety of reasons [124], including morphological and location uncertainty of tumor, class imbalance between foreground and background, and low contrast of MR images and annotation bias. BraTS focuses on segmentations for the enhancing tumor (ET), tumor core (TC), and whole tumor (WT). The Dice score, 95% Hausdorff distance, sensitivity, and specificity are used as evaluation metrics.

BraTS 2021 There are two tasks in BraTS 2021 and one of them is segmentation of brain tumor subregions (task 1) [123].

Dataset The BraTS 2021 competition comprises 8000 multi-parametric MR images from 2000 patients. The data split is 1251 cases for training, 219 cases for the validation phase, and 530 cases for final ranking, and ground truth labels are only provided to participants for the training set. The validation phase aims to help the participants examine their algorithm, and the results are shown on the public leaderboard. The dataset contains four MRI modalities per subject (Fig. 16): T1-w, post-contrast T1-w (T1Gd), T2-w, and T2-fluid-attenuated inversion recovery (T2-FLAIR).

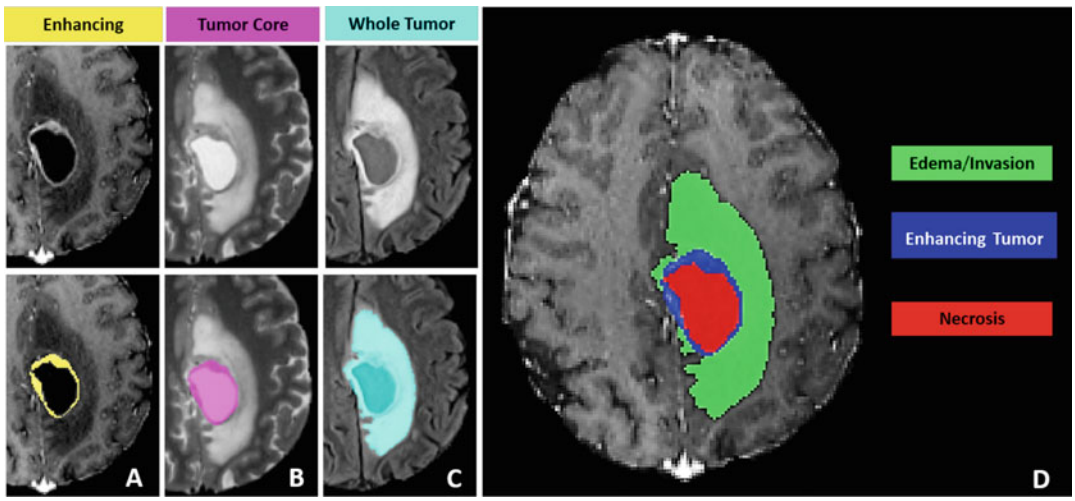

Fig. 16 BraTS 2021 dataset. The images and ground truth labels of enhancing tumor, tumor core, and whole tumor are shown in the panels A (T1w with gadolinium injection), B (T2w), and C (T2-FLAIR), respectively. Panel D shows the combined segmentations to generate the final tumor subregion labels. Replicated from [123] (CC BY 4.0)

The images were acquired at different institutions with different protocols and scanners. The pre-processing pipeline includes (1) co-registration to the same anatomical template, (2) resampling to isotropic $1mm^3$ resolution, and (3) skull stripping.

Winner Method Luu et al. contributed a novel method [125] that won the first place in the final ranking after being applied to unseen test data. Their work is based on the nnU-Net, the winner of BraTS 2020. Some contributions include using group normalization instead of batch normalization; employing axial attention modules [126, 127] in the decoder part, which is efficient for multidimensional data; and building a deeper network. In the training phase, the networks were trained with 5-fold cross-validation. "Online" data augmentations were applied, including random rotation and scaling, elastic deformation, additive brightness augmentation, and gamma correction. The sum of the cross-entropy and Dice losses was used as the loss function. Last but not least, before feeding the input, the volumes were cropped to nonzero voxels and normalized by their mean and standard deviation.

3.3 Multiple Sclerosis Segmentation Challenge

Multiple sclerosis (MS) lesion segmentation from MR images is challenging for both radiologists and automated algorithms. The difficulties of this task include the large variability of lesion appearance, boundary, shape, and location, as well as variations in image appearance caused by different scanners and acquisition protocols from different institutes [128].

MSSEG-2 Delineation of new MS lesions on T2/FLAIR images is of interest as a biomarker of the effectiveness of anti-inflammatory disease-modifying drugs. Building upon the MSSEG (multiple sclerosis segmentation) challenge, MSSEG-2 (https://portal.fli-iam.irisa.fr/msseg-2/) focuses on new MS lesion detection and segmentation. Here, we focus on the new lesion segmentation task.

Dataset The MSSEG-2 challenge dataset consists of 100 MS patients with 200 scans. Each subject has two FLAIR scans at different timepoints, with a time gap between 1 and 3 years. The images are acquired with 15 different 1.5T/3T scanners. Forty patients and their labels are used for training, and 120 scans of 60 patients are provided to test the performance.

Winner Method Zhang et al. proposed a novel method for segmentation of new MS lesions [56] that performed best for the Dice score evaluation. They adopted the model from [46], which is based on the U-Net and dense connections. The model inputs the concatenation of MR images from different timepoints and

outputs the new MS lesion segmentation for each patient. In addition, the 2.5D method, which stacks slices from three different orthogonal views (axial, sagittal, and coronal), is applied to each MR scan. In this way, both local and global information are provided to the model during training. Furthermore, to increase the generalizability of the model from the source domain to the target domain, three types of data augmentation are used that include image quality augmentation, image intensity augmentation, and spatial augmentation.

4 Conclusion

Image segmentation is a crucial task in medical image analysis. With the help of deep learning algorithms, one can achieve more precise segmentation on brain structures and lesions. In this chapter, we first introduced the fundamental components (Subheadings 2.1.1–2.1.6) needed to set up a complete deep neural network for a medical image segmentation task. Next, we provided a review of the rich literature on medical image segmentation methods categorized by supervision settings in Subheading 2.2–2.5. For each type of supervision, we explained the main ideas and provided example applications. Finally, we introduced some medical image segmentation challenges (Subheading 3) that have publicly available data, so that the readers can start their own projects.

References

1. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer, Berlin, pp 234–241
2. Milletari F, Navab N, Ahmadi SA (2016) V-net: fully convolutional neural networks for volumetric medical image segmentation. In: 2016 fourth international conference on 3D vision (3DV). IEEE, Piscataway, pp 565–571
3. Oktay O, Schlemper J, Folgoc LL, Lee M, Heinrich M, Misawa K, Mori K, McDonagh S, Hammerla NY, Kainz B, et al (2018) Attention u-net: learning where to look for the pancreas. Preprint. arXiv:180403999
4. Schlemper J, Oktay O, Schaap M, Heinrich M, Kainz B, Glocker B, Rueckert D (2019) Attention gated networks: Learning to leverage salient regions in medical images. Med Image Anal 53:197–207
5. Isensee F, Jäger PF, Kohl SA, Petersen J, Maier-Hein KH (2019) Automated design of deep learning methods for biomedical image segmentation. Preprint. arXiv:190408128
6. Isensee F, Jaeger PF, Kohl SA, Petersen J, Maier-Hein KH (2021) nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nat. Methods 18(2):203–211
7. Çiçek Ö, Abdulkadir A, Lienkamp SS, Brox T, Ronneberger O (2016) 3D U-Net: learning dense volumetric segmentation from sparse annotation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Berlin, pp 424–432
8. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
9. Zhang J, Jiang Z, Dong J, Hou Y, Liu B (2020) Attention gate resU-Net for automatic MRI brain tumor segmentation. IEEE Access 8:58533–58545

10. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in Neural Information Processing Systems, pp 5998–6008
11. Luong MT, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation. Preprint. arXiv:150804025
12. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, et al (2020) An image is worth 16×16 words: transformers for image recognition at scale. Preprint. arXiv:201011929
13. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. Preprint. arXiv:210314030
14. Hatamizadeh A, Tang Y, Nath V, Yang D, Myronenko A, Landman B, Roth HR, Xu D (2022) Unetr: transformers for 3D medical image segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp 574–584
15. Chen J, Lu Y, Yu Q, Luo X, Adeli E, Wang Y, Lu L, Yuille AL, Zhou Y (2021) Transunet: transformers make strong encoders for medical image segmentation. Preprint. arXiv:210204306
16. Ma J, Chen J, Ng M, Huang R, Li Y, Li C, Yang X, Martel AL (2021) Loss odyssey in medical image segmentation. Med Image Anal 71:102035
17. Jadon S (2020) A survey of loss functions for semantic segmentation. In: 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB). IEEE, Piscataway, pp 1–7
18. Sudre CH, Li W, Vercauteren T, Ourselin S, Jorge Cardoso M (2017) Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer, Berlin, pp 240–248
19. Salehi SSM, Erdogmus D, Gholipour A (2017) Tversky loss function for image segmentation using 3D fully convolutional deep networks. In: International Workshop on Machine Learning in Medical Imaging. Springer, Berlin, pp 379–387
20. El Jurdi R, Petitjean C, Honeine P, Cheplygina V, Abdallah F (2021) High-level prior-based loss functions for medical image segmentation: a survey. Comput Vis Image Underst 210:103248
21. Maier-Hein L, Reinke A, Christodoulou E, Glocker B, Godau P, Isensee F, Kleesiek J, Kozubek M, Reyes M, Riegler MA, et al (2022) Metrics reloaded: pitfalls and recommendations for image analysis validation. Preprint. arXiv:220601653
22. Shattuck DW, Sandor-Leahy SR, Schaper KA, Rottenberg DA, Leahy RM (2001) Magnetic resonance image tissue classification using a partial volume model. NeuroImage 13(5): 856–876
23. Hahn HK, Peitgen HO (2000) The skull stripping problem in mri solved by a single 3D watershed transform. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Berlin, pp 134–143
24. Smith SM (2002) Fast robust automated brain extraction. Hum Brain Mapp 17(3): 143–155
25. Leung KK, Barnes J, Modat M, Ridgway GR, Bartlett JW, Fox NC, Ourselin S, Initiative ADN, et al (2011) Brain maps: an automated, accurate and robust brain extraction technique using a template library. NeuroImage 55(3):1091–1108
26. Ségonne F, Dale AM, Busa E, Glessner M, Salat D, Hahn HK, Fischl B (2004) A hybrid approach to the skull stripping problem in MRI. NeuroImage 22(3):1060–1075
27. Kleesiek J, Urban G, Hubert A, Schwarz D, Maier-Hein K, Bendszus M, Biller A (2016) Deep MRI brain extraction: a 3D convolutional neural network for skull stripping. NeuroImage 129:460–469
28. Yogananda CGB, Wagner BC, Murugesan GK, Madhuranthakam A, Maldjian JA (2019) A deep learning pipeline for automatic skull stripping and brain segmentation. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). IEEE, Piscataway, pp 727–731
29. Zhang Q, Wang L, Zong X, Lin W, Li G, Shen D (2019) Frnet: Flattened residual network for infant MRI skull stripping. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). IEEE, Piscataway, pp 999–1002
30. Isensee F, Schell M, Pflueger I, Brugnara G, Bonekamp D, Neuberger U, Wick A, Schlemmer HP, Heiland S, Wick W, et al (2019) Automated brain extraction of multisequence MRI using artificial neural networks. Hum Brain Mapp 40(17):4952–4964
31. Gao Y, Li J, Xu H, Wang M, Liu C, Cheng Y, Li M, Yang J, Li X (2019) A multi-view pyramid network for skull stripping on neonatal

T1-weighted MRI. Magn Reson Imaging 63: 70–79

32. Li H, Zhu Q, Hu D, Gunnala MR, Johnson H, Sherbini O, Gavazzi F, D'Aiello R, Vanderver A, Long JD, et al (2022) Human brain extraction with deep learning. In: Medical Imaging 2022: Image Processing, vol 12032. SPIE, Bellingham, pp 369–375
33. Kalavathi P, Prasath VS (2016) Methods on skull stripping of MRI head scan images—a review. J Digit Imaging 29(3):365–379
34. Juntu J, Sijbers J, Van Dyck D, Gielen J (2005) Bias field correction for MRI images. In: Computer Recognition Systems. Springer, Berlin, pp 543–551
35. Tustison NJ, Avants BB, Cook PA, Zheng Y, Egan A, Yushkevich PA, Gee JC (2010) N4itk: improved N3 bias correction. IEEE Trans Med Imaging 29(6):1310–1320
36. Fortin JP, Parker D, Tunç B, Watanabe T, Elliott MA, Ruparel K, Roalf DR, Satterthwaite TD, Gur RC, Gur RE, et al (2017) Harmonization of multi-site diffusion tensor imaging data. NeuroImage 161:149–170
37. Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2223–2232
38. Reinhold JC, Dewey BE, Carass A, Prince JL (2019) Evaluating the impact of intensity normalization on MR image synthesis. In: Medical Imaging 2019: Image Processing, vol 10949. SPIE, Bellingham, pp 890–898
39. Shinohara RT, Sweeney EM, Goldsmith J, Shiee N, Mateen FJ, Calabresi PA, Jarso S, Pham DL, Reich DS, Crainiceanu CM, et al (2014) Statistical normalization techniques for magnetic resonance imaging. NeuroImage Clin 6:9–19
40. Brett M, Johnsrude IS, Owen AM (2002) The problem of functional localization in the human brain. Nat Rev Neurosci 3(3): 243–249
41. Shin H, Kim H, Kim S, Jun Y, Eo T, Hwang D (2022) COSMOS: cross-modality unsupervised domain adaptation for 3D medical image segmentation based on target-aware domain translation and iterative self-training. Preprint. arXiv:220316557
42. Dong H, Yu F, Zhao J, Dong B, Zhang L (2021) Unsupervised domain adaptation in semantic segmentation based on pixel alignment and self-training. Preprint. arXiv:210914219
43. Liu H, Fan Y, Cui C, Su D, McNeil A, Dawant BM (2022) Unsupervised domain adaptation for vestibular schwannoma and cochlea segmentation via semi-supervised learning and label fusion. Preprint. arXiv:220110647
44. Isensee F, Petersen J, Kohl SA, Jäger PF, Maier-Hein KH (2019) nnU-Net: breaking the spell on successful medical image segmentation. Preprint 1:1–8. arXiv:190408128
45. Birenbaum A, Greenspan H (2016) Longitudinal multiple sclerosis lesion segmentation using multi-view convolutional neural networks. In: Deep Learning and Data Labeling for Medical Applications. Springer, Berlin, pp 58–67
46. Zhang H, Valcarcel AM, Bakshi R, Chu R, Bagnato F, Shinohara RT, Hett K, Oguz I (2019) Multiple sclerosis lesion segmentation with tiramisu and 2.5 D stacked slices. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Berlin, pp 338–346
47. Kamnitsas K, Ledig C, Newcombe VF, Simpson JP, Kane AD, Menon DK, Rueckert D, Glocker B (2017) Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. Med Image Anal 36:61–78
48. Dolz J, Desrosiers C, Ayed IB (2018) 3D fully convolutional networks for subcortical segmentation in MRI: A large-scale study. NeuroImage 170:456–470
49. Li H, Zhang H, Hu D, Johnson H, Long JD, Paulsen JS, Oguz I (2020) Generalizing MRI subcortical segmentation to neurodegeneration. In: Machine Learning in Clinical Neuroimaging and Radiogenomics in Neuro-oncology. Springer, Berlin, pp 139–147
50. Wang G, Li W, Ourselin S, Vercauteren T (2017) Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. In: International MICCAI Brainlesion Workshop. Springer, Berlin, pp 178–190
51. Li H, Hu D, Zhu Q, Larson KE, Zhang H, Oguz I (2021) Unsupervised cross-modality domain adaptation for segmenting vestibular schwannoma and cochlea with data augmentation and model ensemble. Preprint. arXiv:210912169
52. Zhang L, Wang X, Yang D, Sanford T, Harmon S, Turkbey B, Wood BJ, Roth H, Myronenko A, Xu D, et al (2020) Generalizing deep learning for medical image segmentation to unseen domains via deep

stacked transformation. IEEE Trans Med Imaging 39(7):2531–2540

53. Li H, Zhang H, Johnson H, Long JD, Paulsen JS, Oguz I (2021) MRI subcortical segmentation in neurodegeneration with cascaded 3D CNNs. In: Medical Imaging 2021: Image Processing, International Society for Optics and Photonics, vol 11596, p 115960W
54. Dong H, Yang G, Liu F, Mo Y, Guo Y (2017) Automatic brain tumor detection and segmentation using U-Net based fully convolutional networks. In: Annual Conference on Medical Image Understanding and Analysis. Springer, Berlin, pp 506–517
55. Beers A, Chang K, Brown J, Sartor E, Mammen C, Gerstner E, Rosen B, Kalpathy-Cramer J (2017) Sequential 3D u-nets for biologically-informed brain tumor segmentation. Preprint. arXiv:170902967
56. Zhang H, Li H, Oguz I (2021) Segmentation of new MS lesions with tiramisu and 2.5 D stacked slices. In: MSSEG-2 Challenge Proceedings: Multiple Sclerosis New Lesions Segmentation Challenge Using a Data Management and Processing Infrastructure, p 61
57. Myronenko A (2018) 3D MRI brain tumor segmentation using autoencoder regularization. In: International MICCAI Brainlesion Workshop. Springer, Berlin, pp 311–320
58. Pérez-García F, Sparks R, Ourselin S (2021) Torchio: a python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning. Comput Methods Prog Biomed 208:106236
59. Kamnitsas K, Ferrante E, Parisot S, Ledig C, Nori AV, Criminisi A, Rueckert D, Glocker B (2016) Deepmedic for brain tumor segmentation. In: International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. Springer, Berlin, pp 138–149
60. Havaei M, Davy A, Warde-Farley D, Biard A, Courville A, Bengio Y, Pal C, Jodoin PM, Larochelle H (2017) Brain tumor segmentation with deep neural networks. Med Image Anal 35:18–31
61. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3431–3440
62. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 770–778
63. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4700–4708
64. Isensee F, Kickingereder P, Wick W, Bendszus M, Maier-Hein KH (2017) Brain tumor segmentation and radiomics survival prediction: contribution to the brats 2017 challenge. In: International MICCAI Brainlesion Workshop. Springer, Berlin, pp 287–297
65. Chang PD (2016) Fully convolutional deep residual neural networks for brain tumor segmentation. In: International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. Springer, Berlin, pp 108–118
66. Castillo LS, Daza LA, Rivera LC, Arbeláez P (2017) Volumetric multimodality neural network for brain tumor segmentation. In: 13th International Conference on Medical Information Processing and Analysis, vol 10572. International Society for Optics and Photonics, Bellingham, p 105720E
67. Jégou S, Drozdzal M, Vazquez D, Romero A, Bengio Y (2017) The one hundred layers tiramisu: fully convolutional densenets for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 11–19
68. Wang G, Shapey J, Li W, Dorent R, Demitriadis A, Bisdas S, Paddick I, Bradford R, Zhang S, Ourselin S, et al (2019) Automatic segmentation of vestibular schwannoma from T2-weighted MRI by deep spatial attention with hardness-weighted loss. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Berlin, pp 264–272
69. Zhang H, Zhang J, Zhang Q, Kim J, Zhang S, Gauthier SA, Spincemaille P, Nguyen TD, Sabuncu M, Wang Y (2019) RSANet: recurrent slice-wise attention network for multiple sclerosis lesion segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Berlin, pp 411–419
70. Hou B, Kang G, Xu X, Hu C (2019) Cross attention densely connected networks for multiple sclerosis lesion segmentation. In: 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, Piscataway, pp 2356–2361
71. Islam M, Vibashan V, Jose VJM, Wijethilake N, Utkarsh U, Ren H (2019) Brain tumor segmentation and survival prediction using 3D attention UNet. In:

International MICCAI Brainlesion Workshop. Springer, Berlin, pp 262–272

72. Zhou T, Ruan S, Guo Y, Canu S (2020) A multi-modality fusion network based on attention mechanism for brain tumor segmentation. In: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI). IEEE, Piscataway, pp 377–380
73. Sinha A, Dolz J (2020) Multi-scale self-guided attention for medical image segmentation. IEEE J Biomed Health Inform 25(1): 121–130
74. Shamshad F, Khan S, Zamir SW, Khan MH, Hayat M, Khan FS, Fu H (2022) Transformers in medical imaging: a survey. Preprint. arXiv:220109873
75. Hatamizadeh A, Nath V, Tang Y, Yang D, Roth H, Xu D (2022) Swin UNETR: Swin transformers for semantic segmentation of brain tumors in MRI images. Preprint. arXiv:220101266
76. Peiris H, Hayat M, Chen Z, Egan G, Harandi M (2021) A volumetric transformer for accurate 3D tumor segmentation. Preprint. arXiv:211113300
77. Cao H, Wang Y, Chen J, Jiang D, Zhang X, Tian Q, Wang M (2021) Swin-UNET: Unet-like pure transformer for medical image segmentation. Preprint. arXiv:210505537
78. Zhang Y, Liu H, Hu Q (2021) Transfuse: fusing transformers and CNNs for medical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Berlin, pp 14–24
79. Li H, Hu D, Liu H, Wang J, Oguz I (2022) Cats: complementary CNN and transformer encoders for segmentation. In: 2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI). IEEE, Piscataway, pp 1–5
80. Valanarasu JMJ, Oza P, Hacihaliloglu I, Patel VM (2021) Medical transformer: gated axial-attention for medical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Berlin, pp 36–46
81. McKinley R, Wepfer R, Aschwanden F, Grunder L, Muri R, Rummel C, Verma R, Weisstanner C, Reyes M, Salmen A, et al (2019) Simultaneous lesion and neuroanatomy segmentation in multiple sclerosis using deep neural networks. Preprint. arXiv:190107419
82. Huo Y, Xu Z, Xiong Y, Aboud K, Parvathaneni P, Bao S, Bermudez C, Resnick SM, Cutting LE, Landman BA (2019) 3D whole brain segmentation using spatially localized atlas network tiles. NeuroImage 194: 105–119
83. Kamnitsas K, Bai W, Ferrante E, McDonagh S, Sinclair M, Pawlowski N, Rajchl M, Lee M, Kainz B, Rueckert D, et al (2017) Ensembles of multiple models and architectures for robust brain tumour segmentation. In: International MICCAI Brainlesion Workshop. Springer, Berlin, pp 450–462
84. Kao PY, Ngo T, Zhang A, Chen JW, Manjunath B (2018) Brain tumor segmentation and tractographic feature extraction from structural MR images for overall survival prediction. In: International MICCAI Brainlesion Workshop. Springer, Berlin, pp 128–141
85. Zhao X, Wu Y, Song G, Li Z, Zhang Y, Fan Y (2017) 3D brain tumor segmentation through integrating multiple 2D FCNNs. In: International MICCAI Brainlesion Workshop. Springer, Berlin, pp 191–203
86. Zhang D, Huang G, Zhang Q, Han J, Han J, Yu Y (2021) Cross-modality deep feature learning for brain tumor segmentation. Pattern Recogn 110:107562
87. Havaei M, Guizard N, Chapados N, Bengio Y (2016) Hemis: hetero-modal image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Berlin, pp 469–477
88. Liu H, Fan Y, Li H, Wang J, Hu D, Cui C, Lee HZ Ho Hin, Oguz I (2022) Moddrop++: a dynamic filter network with intra-subject co-training for multiple sclerosis lesion segmentation with missing modalities. Preprint. arXiv:220304959
89. Wang Y, Zhang Y, Liu Y, Lin Z, Tian J, Zhong C, Shi Z, Fan J, He Z (2021) ACN: adversarial co-training network for brain tumor segmentation with missing modalities. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Berlin, pp 410–420
90. Azad R, Khosravi N, Merhof D (2022) SMU-Net: style matching U-Net for brain tumor segmentation with missing modalities. Preprint. arXiv:220402961
91. Gao Y, Phillips JM, Zheng Y, Min R, Fletcher PT, Gerig G (2018) Fully convolutional structured LSTM networks for joint 4D medical image segmentation. In: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018). IEEE, Piscataway, pp 1104–1108

92. Li H, Zhang H, Johnson H, Long JD, Paulsen JS, Oguz I (2021) Longitudinal subcortical segmentation with deep learning. In: Medical Imaging 2021: Image Processing, International Society for Optics and Photonics, vol 11596, p 115960D
93. Van Engelen JE, Hoos HH (2020) A survey on semi-supervised learning. Mach Learn 109(2):373–440
94. Lee DH, et al (2013) Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning, ICML, vol 3, p 896
95. Laine S, Aila T (2016) Temporal ensembling for semi-supervised learning. Preprint. arXiv:161002242
96. Tarvainen A, Valpola H (2017) Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results. Preprint. arXiv:170301780
97. Xie Q, Dai Z, Hovy E, Luong MT, Le QV (2019) Unsupervised data augmentation for consistency training. Preprint. arXiv:190412848
98. Berthelot D, Carlini N, Goodfellow I, Papernot N, Oliver A, Raffel C (2019) Mixmatch: a holistic approach to semi-supervised learning. Preprint. arXiv:190502249
99. Sohn K, Berthelot D, Li CL, Zhang Z, Carlini N, Cubuk ED, Kurakin A, Zhang H, Raffel C (2020) Fixmatch: simplifying semi-supervised learning with consistency and confidence. Preprint. arXiv:200107685
100. Cubuk ED, Zoph B, Shlens J, Le QV (2020) Randaugment: practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp 702–703
101. Chen G, Ru J, Zhou Y, Rekik I, Pan Z, Liu X, Lin Y, Lu B, Shi J (2021) MTANS: multi-scale mean teacher combined adversarial network with shape-aware embedding for semi-supervised brain lesion segmentation. NeuroImage 244:118568
102. Carass A, Roy S, Jog A, Cuzzocreo JL, Magrath E, Gherman A, Button J, Nguyen J, Prados F, Sudre CH, et al (2017) Longitudinal multiple sclerosis lesion segmentation: resource and challenge. NeuroImage 148:77–102
103. Maier O, Menze BH, von der Gablentz J, Häni L, Heinrich MP, Liebrand M, Winzeck S, Basit A, Bentley P, Chen L, et al (2017) ISLES 2015-a public evaluation benchmark for ischemic stroke lesion segmentation from multispectral MRI. Med Image Anal 35:250–269
104. Menze BH, Jakab A, Bauer S, Kalpathy-Cramer J, Farahani K, Kirby J, Burren Y, Porz N, Slotboom J, Wiest R, et al (2014) The multimodal brain tumor image segmentation benchmark (BraTS). IEEE Trans Med Imaging 34(10):1993–2024
105. Amini MR, Gallinari P (2002) Semi-supervised logistic regression. In: ECAI, vol 2, p 11
106. Takaya E, Takeichi Y, Ozaki M, Kurihara S (2021) Sequential semi-supervised segmentation for serial electron microscopy image with small number of labels. J Neurosci Methods 351:109066
107. Arganda-Carreras I, Turaga SC, Berger DR, Cireşan D, Giusti A, Gambardella LM, Schmidhuber J, Laptev D, Dwivedi S, Buhmann JM, et al (2015) Crowdsourcing the creation of image segmentation algorithms for connectomics. Front Neuroanat 9:142
108. Takeichi Y, Uebi T, Miyazaki N, Murata K, Yasuyama K, Inoue K, Suzaki T, Kubo H, Kajimura N, Takano J, et al (2018) Putative neural network within an olfactory sensory unit for nestmate and non-nestmate discrimination in the Japanese carpenter ant: the ultra-structures and mathematical simulation. Front Cell Neurosci 12:310
109. Baur C, Denner S, Wiestler B, Navab N, Albarqouni S (2021) Autoencoders for unsupervised anomaly segmentation in brain MR images: a comparative study. Med Image Anal, p 101952
110. Pawlowski N, Lee MC, Rajchl M, McDonagh S, Ferrante E, Kamnitsas K, Cooke S, Stevenson S, Khetani A, Newman T, et al (2018) Unsupervised lesion detection in brain CT using bayesian convolutional autoencoders. MIDL
111. Kingma DP, Welling M (2013) Auto-encoding variational bayes. Preprint. arXiv:13126114
112. Chen X, Konukoglu E (2018) Unsupervised detection of lesions in brain MRI using constrained adversarial auto-encoders. Preprint. arXiv:180604972
113. Pinaya WHL, Tudosiu PD, Gray R, Rees G, Nachev P, Ourselin S, Cardoso MJ (2021) Unsupervised brain anomaly detection and segmentation with transformers. Preprint. arXiv:210211650
114. Van Den Oord A, Vinyals O, et al (2017) Neural discrete representation learning. Adv Neural Inf Proces Syst 30

115. Dilokthanakul N, Mediano PA, Garnelo M, Lee MC, Salimbeni H, Arulkumaran K, Shanahan M (2016) Deep unsupervised clustering with gaussian mixture variational autoencoders. Preprint. arXiv:161102648
116. You S, Tezcan KC, Chen X, Konukoglu E (2019) Unsupervised lesion detection via image restoration with a normative prior. In: International Conference on Medical Imaging with Deep Learning, PMLR, pp 540–556
117. Baur C, Wiestler B, Albarqouni S, Navab N (2018) Deep autoencoding models for unsupervised anomaly segmentation in brain MR images. In: International MICCAI Brainlesion Workshop. Springer, Berlin, pp 161–169
118. Antonelli M, Reinke A, Bakas S, Farahani K, Kopp-Schneider A, Landman BA, Litjens G, Menze B, Ronneberger O, Summers RM, et al (2022) The medical segmentation decathlon. Nat Commun 13(1):1–13
119. Dorent R, Kujawa A, Ivory M, Bakas S, Rieke N, Joutard S, Glocker B, Cardoso J, Modat M, Batmanghelich K, et al (2022) Crossmoda 2021 challenge: benchmark of cross-modality domain adaptation techniques for vestibular schwnannoma and cochlea segmentation. Preprint. arXiv:220102831
120. Ghaffari M, Sowmya A, Oliver R (2019) Automated brain tumor segmentation using multimodal brain scans: a survey based on models submitted to the BraTS 2012–2018 challenges. IEEE Rev Biomed Eng 13:156–168
121. Bakas S, Reyes M, Jakab A, Bauer S, Rempfler M, Crimi A, Shinohara RT, Berger C, Ha SM, Rozycki M, et al (2018) Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BraTS challenge. Preprint. arXiv:181102629
122. Bakas S, Akbari H, Sotiras A, Bilello M, Rozycki M, Kirby J, Freymann J, Farahani K, Davatzikos C (2017) Segmentation labels and radiomic features for the pre-operative scans of the TCGA-LGG collection. Cancer Imaging Arch 286
123. Baid U, Ghodasara S, Mohan S, Bilello M, Calabrese E, Colak E, Farahani K, Kalpathy-Cramer J, Kitamura FC, Pati S, et al (2021) The RSNA-ASNR-MICCAI braTS 2021 benchmark on brain tumor segmentation and radiogenomic classification. Preprint. arXiv:210702314
124. Liu Z, Chen L, Tong L, Zhou F, Jiang Z, Zhang Q, Shan C, Wang Y, Zhang X, Li L, et al (2020) Deep learning based brain tumor segmentation: a survey. Preprint. arXiv:200709479
125. Luu HM, Park SH (2021) Extending nn-Unet for brain tumor segmentation. Preprint. arXiv:211204653
126. Wang H, Zhu Y, Green B, Adam H, Yuille A, Chen LC (2020) Axial-deeplab: stand-alone axial-attention for panoptic segmentation. In: European Conference on Computer Vision. Springer, Berlin, pp 108–126
127. Ho J, Kalchbrenner N, Weissenborn D, Salimans T (2019) Axial attention in multidimensional transformers. Preprint. arXiv:191212180
128. Zhang H, Oguz I (2020) Multiple sclerosis lesion segmentation–a survey of supervised CNN-based methods. Preprint. arXiv:201208317