Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review

Jose Bernal*, Kaisar Kushibar, Daniel S. Asfaw, Sergi Valverde, Arnau Oliver, Robert Martí, Xavier Lladó

Computer Vision and Robotics Institute, Dept. of Computer Architecture and Technology, University of Girona, Ed. P-IV, Av. Lluis Santaló s/n, 17003 Girona, Spain

ARTICLE INFO

ABSTRACT

In recent years, deep convolutional neural networks (CNNs) have shown record-shattering performance in a variety of computer vision problems, such as visual object recognition, detection and segmentation. These methods have also been utilised in medical image analysis domain for lesion segmentation, anatomical segmentation and classification. We present an extensive literature review of CNN techniques applied in brain magnetic resonance imaging (MRI) analysis, focusing on the architectures, pre-processing, data-preparation and post-processing strategies available in these works. The aim of this study is three-fold. Our primary goal is to report how different CNN architectures have evolved, discuss state-of-the-art strategies, condense their results obtained using public datasets and examine their pros and cons. Second, this paper is intended to be a detailed reference of the research activity in deep CNN for brain MRI analysis. Finally, we present a perspective on the future of CNNs in which we hint some of the research directions in subsequent years.

1. Introduction

Convolutional neural networks (CNNs), an outstanding branch of deep learning applications to visual purposes, have earned major attention in the last years due to its breakthrough performances in varied computer vision applications, such as in object recognition, detection and segmentation challenges [1–3], in which they have achieved astonishing performances [4–8]. Their success has not been limited to reach top positions in different tasks, but also to achieve competent personnel levels in visual object recognition tasks [9] and, perhaps more importantly, sensitive medical applications [10].

CNNs have been used in medical imaging applications since the 1990s in areas such as lung structure and nodule detection [11,12] and breast tissue classification [13]. However, due to the lack of labelled training data and computational power limitations by that time, it was not possible to train deep CNNs without over-fitting. As a result, proposals in this regard were discontinued in the field for some years. With the time, large annotated training datasets and more powerful graphics processing units (GPUs) have been created, enabling researchers to continue working in the area. This trend can be observed in Fig. 1 in which the number of papers using these strategies has increased year after year from 2010. In brain image analysis, the pioneering work

appeared in multiple sclerosis (MS) by Maleki et al. [14], after Krizhevsky et al. [4] rekindled research on CNNs in 2012. Nowadays, deep CNN architectures are widely used in brain MRI for preprocessing data [15], detecting and segmenting lesions [16–21] and segmenting tumours [22–24], whole tissue [25–27] and sub-cortical structures [28,29].

Although in medical image classification tasks, there are usually fewer classes compared to large-scale semantic image recognition, two significant difficulties hindering achieving similar accuracy to human raters. First, there is a lack of sufficiently labelled training data. Essentially, generating highly accurate labels and finding sufficient preprocessed and representative data require a considerable amount of time. Second, medical image annotation is carried out by experts which is subjective and error-prone [30,31]. Learning a model from a less accurate representation of training samples degenerates the algorithm accuracy.

Brain MR image analysis has traditionally been an important area of research, attracting researchers to work on different tasks, such as lesion detection and segmentation, tissue segmentation and brain parcellation on neonatal, infant and adult subjects [32–35]. Several public brain MR image datasets are available to the community, especially those organised by the Medical Image Computing and Computer-

* Corresponding author.
  E-mail addresses: jose.bernal@udg.edu (J. Bernal), kaisar.kushibar@udg.edu (K. Kushibar), daniel.asfaw@udg.edu (D.S. Asfaw), svalverde@eia.udg.edu (S. Valverde), aoliver@eia.udg.edu (A. Oliver), marly@eia.udg.edu (R. Martí), llado@eia.udg.edu (X. Lladó).
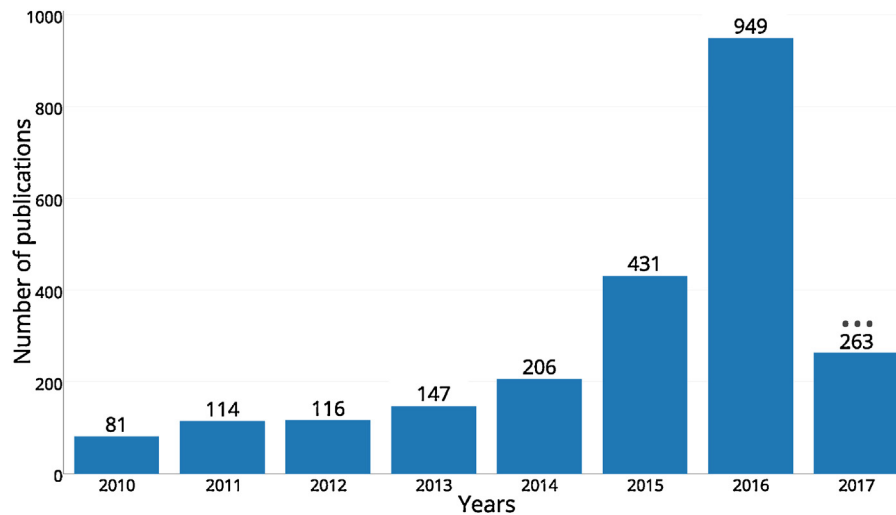
Fig. 1. Number of publications per year in IEEE-Xplore containing "deep learning" and "medical imaging" keywords from 2010 to 2017. (Queried: June 6th, 2017).

Assisted Intervention (MICCAI[1]) society, actively encouraging research and publications in the field. These standard evaluation frameworks have been proposed for quantitatively comparing brain segmentation algorithms under the same directives and conditions. Consequently, the number of publications using CNNs in brain MRI has been increasing: of 23 works collected from PubMed, Scholar, IEEE Xplore and Scopus databases for this review work, 20 of them were published in the period from 2015 to May 2017.

Deep learning methods have been extensively reviewed in recent years [36–40] and discussed for particular applications [41–44]. Developments in the field of medical image analysis have been covered in [45,46]. To the best of our knowledge, this work is the first to present a detailed review of deep CNN techniques applied to brain MRI analysis. We intend to comprise all publications in this field published from 2012 to June 2017. Along with [45,46], we aim to provide a good reading basis for newcomers to the topic.

The rest of this review is structured as follows. The general concepts in CNNs are given in Section 2. Note that an experienced reader could skip this section. Afterwards, the different methods published in the literature on brain MRI are surveyed and analysed regarding advantages and disadvantages in Section 3. In Section 4, evaluations and comparisons of the works are performed, based on the reported numerical results. We conclude this work with a discussion, indicating future trends in the field.

2. Deep convolutional neural networks

For years, conventional supervised machine-learning techniques were built using automatic learning techniques and well-engineered algorithms. The approach consisted in taking the raw data, describing its content with low-dimensional feature vectors – using specific prior knowledge of the addressed scenario – and inputting the vectors to a trainable classifier. While the classifier was indeed useful for other purposes, the features were not necessarily generic. Indeed, the overall accuracy of the method would depend on how appropriately designed were the heuristics [47].

Representation learning appears as an alternative to this drawback: discover automatically suitable detection and classification representations from the input data. One of the first successful attempts using this strategy took place in 1998 when LeCun et al. [48] presented a five-layer fully-adaptive architecture for addressing handwritten digit recognition. Despite its accuracy results (1% error rate and 9% reject

rate from a dataset of around 1000 samples), the authors were able to apply neural networks to a real-world task. However, these authors were not able to scale a large number of hidden layers and larger images, mainly due to computational resource constraints and the "vanishing gradient" problem. This latter situation being the case when the gradients of the network's output with regard to the parameters in early layers becomes negligible [49].

Two initial solutions were proposed in the literature to address the "vanishing gradient" problem. The first one was introduced by Hinton et al. [50], in which training was conducted in two stages. The first stage was training each layer in an unsupervised, greedy manner, essentially initialising the parameters with better values than random or uniform values. In the second stage, the network was fine-tuned using labelled data. This work rekindled interest in current deep learning strategies. The second solution appeared with the introduction of the ReLU activation function [51], which showed successful results in large-scale image recognition tasks [4]. Along with the availability of faster GPUs, these strategies have dramatically increased the research in deep neural networks.

One of the most widely adopted approaches of deep neural networks is the convolutional neural networks which can process array-like data [40], such as images or video sequences. From a high-level perspective, the idea behind CNN is to identify compositional hierarchy features which object from the real world exhibit: low-level features (e.g. edges) form patterns, and these specific patterns form more intrinsic structures (e.g. shapes, textures). Further information about the building blocks of CNN is provided in following sections.

2.1. Building blocks of CNN

2.1.1. Convolutional layer

CNNs learn the relationships among the pixels of input images by extracting representative features using convolution and pooling operations. The features detected at each layer using learnt kernels vary concerning their complexity, with the first layer extracting simple features, such as edges, and the later layers extracting more complex and high-level features. The convolution operation in CNNs has three main advantages. First, the weight-sharing mechanism helps to deal with high dimensional data, either 2D images or 3D data, such as videos and volumetric images. Second, local connectivity of the input topology can be exploited using 2D or 3D kernels. Finally, slight shift invariance is achieved using pooling layer. The typical architecture of a 2D CNN is shown in Fig. 2.

Very deep CNN architectures were recently proposed to replace the conventional convolutional layer with modules with more powerful
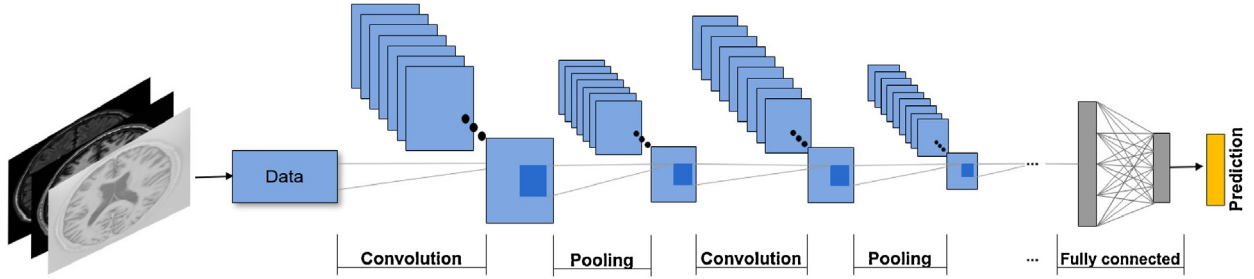
Fig. 2. Generic architecture of convolutional neural networks. The output of each convolution operation at each layer is activated using activation functions before applying pooling operations. The convolution operation produces different numbers of feature maps, depending on the numbers of filters used. The pooling operations reduce the spatial dimensions of each feature map. After convolution and pooling the layers, the future maps are flattened in the fully connected layer before a prediction is made using linear classifiers.

representation while using less computational resources. For instance, Szegedy et al. [52] introduced inception modules that could extract multi-scale features from the input feature maps and could efficiently reduce the number of parameters. This module was further improved in [9,53]. The recent version of the inception module in [6] was created to have a more uniform simplified architecture than previous versions and thus could achieve top performance in large-scale image classification tasks.

2.1.2. Non-linearity layer

The above convolutional layer is usually followed by non-linearity operations. Non-linearity is achieved using a specific family of functions called activation functions. These activation functions ensure that the representation in the input space is mapped to a sparse one and, hence achieving (i) a certain invariance to data variability and (ii) a computationally efficient representation [51]. The former situation refers to the fact that sparse representations are more resilient to slight modifications than dense ones. In the past, sigmoid and hyperbolic tangent functions were commonly used for this purpose. However, for large-scale image recognition, novel activation functions are being continuously proposed. We categorise the commonly used activation functions into three broad families.

Rectified linear units (ReLUs) and variants: which are expressed in a general form as

$$f(z_{lk}^{xy}) = \begin{cases} \max(z_{lk}^{xy}, \ 0) & \text{if } z_{lk}^{xy} > 0, \\ \min(\alpha \cdot z_{lk}^{xy}, \ 0) & \text{if } z_{lk}^{xy} \leq 0, \end{cases} \tag{1}$$

where z_{lk}^{xy} is the input value at position (x, y) on the kth feature map at the lth layer and α is the slope of the negative linear function. There are five special cases distinguished depending on the conditions over α. First, if $\alpha = 0$, the expression results in the so-called ReLU [4,54] which is one of the most commonly used activation functions [40]. Despite its computationally efficient implementation, this method presents some drawbacks due to its gradient discontinuity at the origin in terms of gradient update and empirical performance [6,55]. Second, if α is a small constant, the variant is referred as Leaky ReLU (LReLU) [56]. This approximation enables to cope with the problem of zero gradient. Third, if α is tuned up in the training process along with other parameters using back-propagation, the approach is referred as Parametric ReLU (PReLU) [55]. Fourth, in [57], the parameter α is sampled from a uniform distribution for each example, and this approach is called Randomised ReLU (RReLU). Although using a slope parameter for the negative part in ReLU showed improvements in performance, RReLU showed better performance than the other ReLU variants, based on the evaluation in [57] on image classification tasks. Fifth, recently, Jin et al. [58] proposed a new type of activation function, called S-shaped ReLU (SReLU), in which the essential idea is to consider a piecewise function composed of three linear functions, i.e.

$$f(z_{lk}^{xy}) = \begin{cases} t_r + a_r \cdot (z_{lk}^{xy} - t_r) & \text{if } z_{lk}^{xy} \geq t_r, \\ z_{lk}^{xy} & \text{if } t_r > z_{lk}^{xy} > t_l, \\ t_l + a_l \cdot (z_{lk}^{xy} - t_l) & \text{if } z_{lk}^{xy} \leq t_l, \end{cases} \tag{2}$$

where a_r, a_l, t_r and t_l are learnable parameters. According to the experiments in the paper, the SReLU is able to learn both convex and non-convex functions. This activation improved the performance of well-known CNN architectures in MNIST [59] and ImageNet [60] datasets, compared to ReLU, LReLU and PReLU.

Maxout and variants: Maxout [61] was proposed in particular to improve the optimisation and model averaging performance of dropout training. This activation function is also a generalisation of ReLU. It is computed by calculating the maximum across K affine feature maps, i.e.

$$f(z_{lk}^{xy}) = \max_{k \in [1,K]}(z_{lk}^{xy}). \tag{3}$$

One major drawback of this technique is that the number of weights to be learned in each layer is increased by a factor of K. A workaround to this situation was proposed by Springenberg and Reidmiller [62] in which a probabilistic sampling procedure to compute the maximum across feature maps, generalising maxout was considered. This activation function called Probout empirically matched or improved the performance of maxout [62].

Exponential Linear Units (ELU) and variants: ELU [63] are similar to the extensions of ReLU as they employ an identity for positive inputs. Unlike ReLU variants, they provide saturated output for negative inputs. The saturation in the negative regions of the function is reported to be beneficial for expediting the learning and improving the performance of very deep CNNs. It is defined as

$$f(z_{lk}^{xy}) = \max(z_{lk}^{xy}, \ 0) + \alpha \cdot \min(e^{z_{lk}^{xy}} - 1, \ 0). \tag{4}$$

Trottier et al. [64] defined parameters controlling different aspects of the ELU function and proposed learning them with gradient descent during training. This parametric ELU (PELU) further improved the speed and performance of training deep networks. Using off-the-shelf ResNet [5], PELU performed better than ELU and ReLU in image classification tasks on MNIST, CIFAR-10/100 and ImageNet datasets [64].

2.1.3. Pooling and sub-sampling layer

Typical convolutional layers consist of three steps. First, the layer performs several convolutions to produce feature maps. Second, non-linear activation functions are used on the resulting maps. Third, the output is modified by the pooling layer before reaching the next convolutional layer. The idea of a pooling function is to extract a summary statistics of non-overlapping neighbourhoods – usually – to (i) reduce the number of parameters in the following layers, (ii) control over-fitting, and (iii) achieve slight translation invariance [65].

Among several pooling methods, max pooling and average pooling are widely used types. Their operations are similar except that the former uses the maximum of the activations, and the latter uses the average of them. Max pooling is the most common due to its empirical

performance [66,67]. Apart from these two approaches, other pooling methods are proposed to achieve higher performance.

Stochastic pooling: Zeiler et al. [68] proposed stochastic pooling to regularise the convolutional layers and, hence, overcome the overfitting problem of average pooling and max pooling. In this method, the output of the activation from each pooling region is selected first by computing the probability distribution p by normalising the activations; then, the output activation is chosen by sampling from the multinomial distribution based on p. The authors showed that this pooling strategy improved the accuracy on MNIST, CIFAR-10/100 and street view house numbers image recognition (SVHN) [69] datasets.

Spatial pyramid pooling: Another pooling method called spatial pyramid pooling [70], is proposed to work with any input image size. It is added after the last pooling layer, immediately before the fully connected layer, to generate fixed length representation regardless of the image size or scale. By simple modification of the off-the-shelf CNN architectures with a spatial pyramid pooling layer, this pooling scheme was reported to improve the performance of CNN models.

Deformation pooling: Despite max pooling and average pooling being useful in handling deformations, they are incapable of learning deformation constraints or geometric models from object parts. Deformation pooling (referred as def-pooling) was explicitly designed to overcome this drawback for object detection applications [71].

Combination of max and average pooling: Lee et al. [72] proposed a generalisation of max and average pooling operations that allowed them to be combined and adjusted in the training process. The combination operation was performed using mixed max-average and gated max-average pooling. The latter approach improved the characteristics of the region being pooled. The learning is performed during a combined pooling operation using a binary tree in which each leaf of the tree is associated with a learnt pooling filter. This strategy is called tree pooling. An experiment in [72] reported that this method boosted the performance of AlexNet [4] and GoogLeNet [52] models on MNIST, CIFAR-10 and SVHN datasets.

2.1.4. Fully connected layer

Unlike convolutional layer, the Fully Connected (FC) layer has a full connection to all of the units in the previous layer, as shown in Fig. 2. This layer changes the previous layers 2D structure features into a predefined one-dimensional feature vector. Essentially, the main task of the FC layer is to mine the incoming features to extract information about the content of the input image. The process usually consists in flattening the feature maps coming from convolutional layers, to achieve a one dimension feature vector representation, and, then, inputting it into the FC layer. The output of this layer could either be the predict class labels [4] or an intermediate layer [73] (consecutive FC layers can be stacked together).

Implementing FC layers usually require a large number of parameters – compared to other layers – as each cell of a feature map is fully connected to all elements in the previous layer. Besides, there are two drawbacks of these kinds of layers: (i) a single output is produced – if it is used as output layer – and (ii) accepts fixed-size inputs. The former issue means that a single input image receives a single output label. This situation is computationally inconvenient if the CNN is intended to be used in segmentation tasks rather than classification ones. The second issue relates to problems on extending the network: either the input images are scaled to fit the requirements of the network or the network is re-factorised to be able to process the new data. A solution to these problems lies in the fact that FC layers can be converted to convolutional layers of 1×1 kernels [74] ($1 \times 1 \times 1$ kernels in case of 3D). In this way, the model keeps the fully connected functionality while accepting arbitrary input size image and making dense predictions. These types of architectures, called fully convolutional networks (FCNN), have been gradually introduced in the literature [16,21,23,24,28,74].

2.1.5. Loss function

CNNs are well-known for their ability to extract discriminative features using learned weights in each layer. The learning process is reinforced by employing appropriate loss functions. Loss functions are designed to encourage intra-class similarity and inter-class separability.

In image classification tasks, most CNNs employ softmax loss, which is a combination of the softmax function and cross-entropy loss, mainly because of simplicity and the probabilistic interpretation of softmax classifiers. Hinge loss is another type of loss function, which in conjunction with margin-based classifiers was reported to perform better than cross-entropy loss in a standard image classification challenge [75]. More recently, the research in [76] proposed a Large-Margin Softmax Loss (L-softmax), which is a modification of the softmax loss with a distance margin constraint. This loss was reported to boost the performance of deeply learned features in visual classification and verification tasks.

One of the main issues during the training phase is the disproportion among class observations (i.e. the number of available samples per class varies dramatically) as the resulting classifier may be biased towards the majority class. For instance, the region of interest to be analysed occupies only a small part of the scan. This problem can be addressed in the training phase by (i) undersampling the majority class [23,25], (ii) merging or subdividing classes [27] and (iii) penalising sample misclassification based on the reciprocal frequency [77]; or throughout the loss function. Brosch et al. [78] considered an approach for brain lesion segmentation in which the weighted sum of the mean squared difference of the lesion voxels (sensitivity) and non-lesion voxels (specificity) was used. Also, Milletari et al. [79] proposed an objective function based on Dice Similarity Coefficient (DSC) for a two-class problem. According to the authors, this loss function yielded higher performance than the same architecture using softmax.

Other loss functions are task-specific, such as contrastive losses, which are mostly used to measure similarities between two data points. Su et al. [80] used a combination of softmax and contrastive losses for face recognition. The purpose of the softmax loss was to classify faces into different classes, thus encouraging inter-class separation. Then, the contrastive losses were applied for face verification to enforce intra-class similarity. A distance constraint was introduced to the contrastive loss in [81] to obtain more discrimination of features.

2.1.6. Regularisations

We observed the appealing performance of deep CNN methods in different domains, although they use enormous numbers of parameters. Unless trained on a large, labelled training dataset, proper regularisation should be employed to mitigate over-fitting. There are several regularisation methods widely used in the community, such as L_1 or L_2 regularisation approaches encouraging sparsity and small weight magnitude; early stopping [65] forcing the training to stop when there is a sign of over-fitting (also widely used to select hyper-parameters of the model); batch normalisation [9] in which each batch is preprocessed to achieve mean equal to zero and standard deviation equal to one; and dropout [82,83] in which some feature map units are skipped. This last approach being the dominant as (i) it is computationally inexpensive and (ii) prevents co-adaptation among feature map units by encouraging independent contributions of each of the units to the final prediction.

Different improvements in dropout have been proposed. For instance, Wang et al. [84] proposed a method to expedite training using dropout by sampling from a Gaussian approximation, instead of repeatedly sampling a random subset of input features. Another research in [85] proposed Dropconnect, which drops a subset of weights within the network instead of subset of activation within each layer. This approach showed better generalisation than the original dropout in standard image recognition tasks.

Data augmentation is another approach to improve the generalisation of CNNs by increasing the training dataset by artificially

generating data. Although different image transformations [4,52,55,86] and colour perturbation [87] have been used and proved useful in recognition tasks, one limitation with these methods is that no theoretical background indicates the group of transformations helping to improve the generalisation capability of a model.

2.1.7. Optimisation

Deep CNNs are learnt by searching appropriate values for model parameters optimising the loss function. Gradient descent methods, in which the parameter update is performed using a back-propagation algorithm, are widely used for minimisation [88]. This parameter update is undertaken by computing the loss function for a single, small subset, or the whole training set. Each of these cases is referred in the literature as stochastic, mini-batch and batch gradient descent, respectively. Updating values using the whole training set can be computationally expensive and, hence, mini-batch gradient descent is ubiquitously used in the community, leading to smoother parameter updating and more stable convergence.

The primary difficulty when optimising deep CNNs using gradient descent-based algorithms is the non-convex nature of the loss function. Non-convex functions have several local minima, in which gradient descent methods could get easily trapped [89]. Dauphin et al. [90] argued that saddle points, where the gradient vanishes at non-local optimum places, are much more of an issue than local optima in optimising non-convex loss functions. The momentum variable was introduced to the stochastic gradient descent (SGD) update method to avoid oscillation in local optima. Further, momentum-based SGD was improved to have some prescience about the next update direction, referred to as Nesterov accelerated gradient descent [91,92]. Recent works have also been proposed to escape saddle points during the optimisation process [93,94].

One major drawback of the momentum-based SGD methods is to select an adequate learning rate, a parameter determining how substantial a change in the update should be made. This parameter is commonly set globally to be equal for all settings. Much work has been carried out on tuning the global learning rate adaptively based on the gradient of each parameter. One method in [95], called adaptive gradient algorithm (Adagrad), scales the learning rate of each parameter according to the sum of the previous gradients. The amount of updating differs based on the sparsity of the parameters in each gradient update. Adadelta [96] and Root Mean Square Propagation (RMSProp) [97] improved the drawbacks of Adagrad unstable gradient updating by adopting a more stable learning rate scheme. Kingma and Ba [98] proposed Adam, which improved the gradient computation of both Adadelta and RMSProp. They also empirically showed improved performance in non-convex optimisation problems in different machine learning tasks.

2.1.8. Weight initialisation

Weight initialisation is crucial for non-convex optimisation algorithms. Initialising all the parameters to the same small value results in the undesired case in which all weights are updated the same during back-propagation. Another option that could break the symmetry is to initialise the weights with small values taken from a Gaussian distribution [4]. Glorot and Bengio [99] proposed a method known as

"Xavier" initialisation that normalises the variance of each neuron's output to one. There are two beneficial outcomes from this approach: (i) this avoids the variance in each layer and (ii) keeps the weights from exploding to large values or vanishing. This approach was later improved in [55] to account for the non-linearity of ReLU. This allows training very deep networks with better convergence than the "Xavier" initialisation. Another recent work from He et al. [55] proposed an initialisation by explicitly modelling the non-linearity of rectifiers (ReLU/PReLU), which helped the convergence of extremely deep models.

2.1.9. Normalisation

It is a common practice to use mean-centred training datasets by applying contrast normalisation to train deep CNNs [4,22–24,100]. This simple preprocessing technique improves the convergence speed of SGD algorithms. Ioffe et al. [9] argued that training deep networks can be slow since the distribution of parameters across hidden units changes dynamically during training, which is a phenomenon called the internal covariate shift. Their proposal consisted in normalising the data input in each layer, with a technique known as batch normalisation. This method provides any layer in a network with inputs that come from a unit Gaussian distribution and also enables the use of a high learning rate with a less careful weight initialisation choice. More recently, in [101], the authors introduced weight normalisation. With simple weight re-parametrising at each layer, they were able to show improved conditioning of the optimisation problem and faster convergence of SGD methods.

3. CNN methods for brain image analysis

Automatic segmentation in brain MR has been widely investigated throughout the years to support medical tasks, such as diagnosis and patient monitoring. In the literature, most of the deep learning-based papers for brain MRI analysis have focused on lesion, tumour, tissue and whole brain and sub-cortical structure segmentation. This trend is primarily attributed to the different MICCAI challenges. Each year, the number of participants in these challenges employing deep learning approaches has been increasing. In this section, the proposed methods in this domain are discussed in more detail, based on their CNN architectures. Additionally, the description comprises the considered pipeline steps: pre-processing, data preparation, classification and post-processing techniques. There is no specific CNN architecture that is only suitable for a particular application; rather, a CNN model that is proposed for tumour segmentation could work for structure segmentation, and vice versa, with little or no modification.

The segmentation methods proposed in the literature could be seen from a top-level perspective as presented in Fig. 3. The overall pipeline consists of four stages: preprocessing, data preparation, classification, and post-processing. In the preprocessing stage the different pipelines consider noise filtering techniques, inter and intra-patient normalisation and skull-stripping methods – when necessary. Then, the data is prepared to be processed by the classifier. For instance, data preparation could contemplate augmenting the data or, in patch-based strategies, extracting patches from the input volumes. After that, classification takes place. Finally, once the segmentation results are obtained,
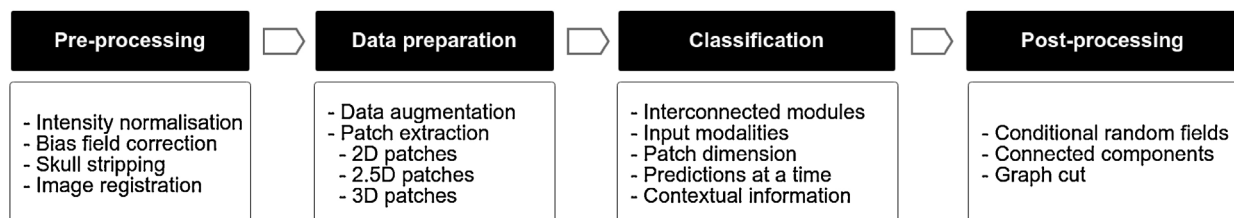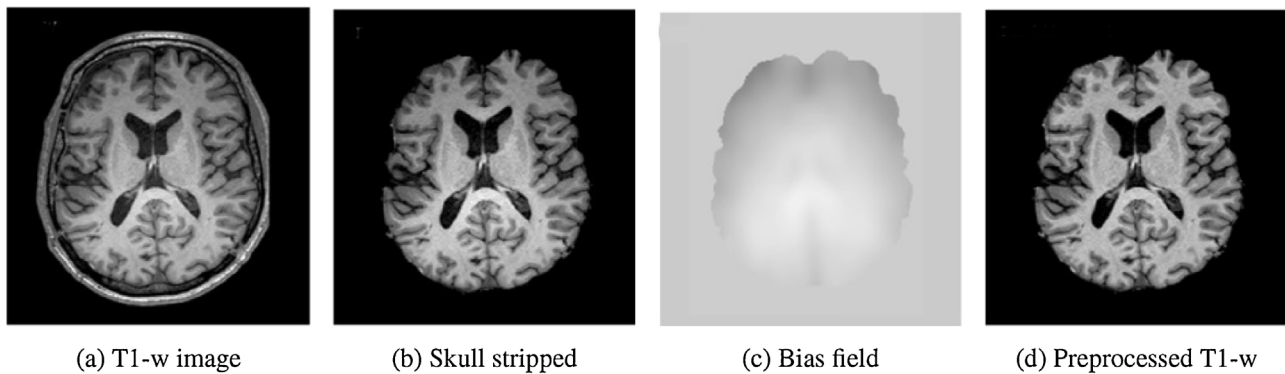
Pre-processing	Data preparation	Classification	Post-processing

- Intensity normalisation
- Bias field correction
- Skull stripping
- Image registration
	- Data augmentation
	- Patch extraction
	  - 2D patches
	  - 2.5D patches
	  - 3D patches
		- Interconnected modules
		- Input modalities
		- Patch dimension
		- Predictions at a time
		- Contextual information
			- Conditional random fields
			- Connected components
			- Graph cut

Fig. 3. A general block diagram of the CNN-based image analysis pipeline.

(a) T1-w image   (b) Skull stripped   (c) Bias field   (d) Preprocessed T1-w

Fig. 4. An example of two preprocessing methods: skull stripping and bias field correction [114]. In the figure, a T1-w slice is displayed in (a), brain tissue after removing non-brain areas in (b), estimated bias field in (c), and preprocessed brain tissue in (d).

they could be refined by removing small isolated areas by selecting the biggest groups only or smoothing regions. It is important to note that some works do not specify any preprocessing or post-processing methods. In the following sections, we will discuss each of these blocks emphasising on the CNN strategies.

3.1. Pre-processing

Pre-processing in MRI is an essential step for subsequent segmentation task. In training supervised models such as CNNs, the input training data hugely influences the performance of the model, so having preprocessed and well-annotated data is a crucial step in achieving good performance.

Acquired brain MRI volumes incorporate non-brain tissue parts of the head, such as eyes, fat, spinal cord or skull. The process of extracting the brain tissue from non-brain one is referred in the literature as skull stripping. An example of an original volume and its corresponding skull stripped output is presented in Fig. 4a and b, respectively. This step has direct consequences on the performance of automated methods, as the inclusion of skull or eyes as brain tissue may lead to unexpected results in classification [102,103], while unintended removal of the cortical surface may result in underestimation of the cortical thickness [104]. Among the different methods proposed in the literature for skull-stripping [105,102,106], methods such as BET [107,108], BSE [109], ROBEX [110] and BEaST [111] are commonly used. In the literature, the methods used in clinical trail datasets employed BET [21,25,112] and ROBEX [23]. Zhang et al. [26] applied a paediatric brain skull stripping algorithm known as LABEL [113]. The public dataset images from the Brain Tumor Image Segmentation Challenge (BRATS) 2013,[2] 2014,[3] and 2015[4] are preprocessed in this regard beforehand.

Inherent characteristics of the MRI acquisition process such as differences in the magnetic field, bandwidth filtering of the data or eddy currents driven by field gradients usually result in image artefacts that may also have a negative impact on the performance of the methods [115]. There is the need to remove spurious intensity variations caused by inhomogeneity of the magnetic fields and coils. In these cases, intensity correction of the MRI images is performed either before tissue segmentation, or as an integrated part of the tissue segmentation pipeline. A common technique to address this problem is to use bias-field correction [116]. The estimated bias field and the corrected version of Fig. 4b are depicted in Fig. 4c and d, respectively. Among the available strategies [117,118], the non-parametric non-uniform intensity normalisation (N3) [119] and N4ITK [120] methods are currently the most

widely used. Zhang et al. [26] and Yoo et al. [112] employed N3 algorithms on their clinical dataset. Similarly, Pereira et al. [22] used them in both BRATS 2013 and 2015 Challenges, Lyksborg et al. [121] in BRATS 2014, and Zikic et al. [122] in BRATS 2013.

Brain MRI datasets might have volumes acquired from different scanner vendors and also from the same scanner but with different protocols. As a result, the volumes may exhibit non-uniform intensity representation for the same tissue types, i.e. interclass variability. To correct this problem, image normalisation algorithms are utilised. According to the literature, this intensity normalisation can be driven in two ways: (i) histogram matching [123–125,15,22] and (ii) normalise data to achieve zero mean and unit variance [22,24,27,28,126]. In the former case, Urban et al. [124] and Kleesiek et al. [15] considered matching the histogram of all volumes to a subject in the training set, which may result in fused grey levels, while Pereira et al. [22] – based on the normalisation method proposed by Nyul et al. [127] – considered mapping to a virtual grey scale learnt directly from the data, so the undesired fusion of grey levels is avoided. Naturally, both normalisation strategies can be used one after the other one to improve the segmentation results. According to the results reported by Pereira et al. [22], the preprocessing step improved their result, obtaining a mean gain of 4.6%.

In addition to the above discussed pre-processing methods, image registration between different MRI modalities is important depending on the dataset analysed. Image registration transforms different modalities of MRI into a common coordinate space. The authors of [14,21,23,78][5] applied image registration algorithms on their clinic trial dataset. For instance, Brosch et al. [21] applied a six degree-of-freedom intra-subject registration using one of the 3 mm scans as the target image to align the different modalities. Additionally, Kamnitsas et al. [23] applied affine atlas-based registration.

3.2. Data preparation

By data preparation, we refer to all of the operations performed before feeding the data into the network, such as data augmentation and patch extraction. Although this stage in most pipelines is considered to be pre-processing, we include it as a separate step since (i) the preprocessing steps are generic while these are particular for the CNN approaches and also (i) to provide more details.

Data augmentation is mainly employed to increase the training samples to mitigate over-fitting as discussed in Section 2.1.6. It is a common practice to use data augmentation in computer vision tasks in which (i) the CNN architectures are very deep, and (ii) obtaining enormous amounts of labelled training data is difficult. Moreover, unless

---

[2] http://martinos.org/qtim/miccai2013/
[3] https://sites.google.com/site/miccaibrats2014/
[4] https://www.smir.ch/BRATS/Start2015

[5] The authors in [14,78] reported using registration during their pre-processing stage but details were scarce in this regard.

the dataset is large enough to correctly train the network (which is not the common case [24,121,125]), the high disproportion between the cardinality of the majority and the minority classes may lead to biased classifiers. Data augmentation could be used in these cases to alleviate and improve the overall performance of the model. Pereira et al. [22] used data augmentation for dual purposes: (i) to increase the training data by applying rotation at different angles and (ii) to introduce class balance by adding more data from the minority class. According to their experiments, by augmenting using rotation, they achieved better delineation of the complete tumour types, as well as of the intra-tumoural structures, reporting a mean gain of up to 2.6%. In addition, by oversampling the minority class, they stated a mean gain of 1.9%. In contrast, Havaei et al. [24] employed data augmentation by flipping the training data but had no success in improving the accuracy of the model. Furthermore, Chen et al. [27] used data augmentation as input, in addition to the given multi-modal images. The augmentation was carried out by subtracting a Gaussian smoothed version and applying histogram equalisation using the Contrast-Limited Adaptive Histogram Equalisation (CLAHE) [128] for enhancing local contrast [27,129]. Afterwards, they used the generated and original volumes to train their network. Nevertheless, the actual effect of this type of augmentation was not reported in their work.

The patch extraction can be performed on a single plane (referred as 2D architectures), from three anatomical planes (referred as 2.5D architectures) or directly from 3D. Also, the patches can be acquired from different imaging modalities – if available. The advantages of one approach or the other one are discussed in the following section. After extracting the patches of training images, the data is normalised by subtracting the mean intensity and dividing by the standard deviation [21,22,26,63,112,130]. The resulting zero mean and unit variance training data helps to expedite the convergence of SGD. It is also a recommended practice to shuffle the samples randomly, especially when training using a mini-batch gradient decent algorithm [131].

3.3. CNN architectures

The classification of CNN architectures in medical imaging can be grouped around five aspects: (i) number of interconnected operating modules, (ii) number of input modalities, (iii) input patch dimension, (iv) number of predictions at a time and (v) implicit and explicit contextual information. All these variants, their advantages and disadvantages are described in the following sections. A summary of the reviewed methods in the literature concerning pre-processing, post-processing and target of the segmentation are described in Table 1.

3.3.1. Interconnected operating modules

According to the number of interconnected operating modules, the strategies in the literature can be classified into single-path and multi-path architectures. The single-path architectures correspond to the cases in which there is a unique flow of information: the input data is processed by convolutional, pooling and non-linear rectifier layers; the feature maps are then mined in the FC layers, and afterwards used for predicting the label in the output layer. An example of 2D single-path architecture is shown in Fig. 2. This category was popular in the literature [14,16,22,15,29,78,122,130] perhaps due to its fast computation and simplicity compared to multi-path architectures. In contrary to single-path networks, the multi-path architectures are a type of CNN in which independent operative networks are integrated into a single model to capture a more varied set of features. The modules can be arranged either in parallel or series. In the former approach, the idea is to achieve a consented label between different sources of information while in the latter approach the key is to reprocess the output of one network using another one.

The parallel multi-path architectures are composed of different CNNs designed to operate in parallel to capture more comprehensive features. Each network uses different versions of the same target area.

There might be two possible variants of this type of architecture. The first variant aggregates features extracted from separate networks before making predictions, which can be understood as a process of information fusion that could occur at two levels: first, at a data level, where feature maps are merged to form a larger feature map; and second, at a feature level, where the extracted final features are concatenated before making the prediction. Fig. 5a graphically depicts a multi-path architecture in which different features are processed using separate networks and are concatenated before making the prediction. This concatenation could also be performed at the convolutional layers, where feature maps from different models form a larger feature map in its third dimension. The second variant of multi-path architectures consists of learning ensembles of more than two networks to improve prediction. Predictions made by each module for a given target voxel are merged probabilistically.

There are several variants of the parallel multi-path category within the literature (i) multi-scale/multi-resolution [23,25,28,124,136] – in which different views of the same ROI are used as input – (ii) 2.5D [121] – in which the data is extracted from the three anatomical planes – and (iii) 2.5D + 3D [135,139] – in which the different modules process 2.5D and 3D patches and operate simultaneously to provide consented labels. The fusion of various information sources allows the network to obtain contrast, local detailed and implicit contextual information.

The authors in [136] and [25] used multi-scale (i.e., different views of the same target area) 2D CNNs using three separate networks and combining the features immediately before the prediction step. According to the authors, the multi-scale design allowed the network to obtain fine-grained local and general information – presumably spatial context as well – to produce the classification of the target pixel. Similarly, Lyksborg et al. [121] trained three networks separately, using 2D slices from axial, sagittal and coronal planes. Unlike in [136] and [25], the ensemble of the three networks was performed after the prediction step, and it was based on majority vote. The ensemble, in this case, was used twice: to segment the tumour area and to perform a within-tumour segmentation.

Another work, presented by Kamnitsas et al. [23], proposed a network called DeepMedic that used a small $3 \times 3 \times 3$ kernel to increase the depth of the network to 11 layers, and they used two scales to train two separate networks for brain lesion segmentation. The output of each network was fused on the fully connected layer. The 11 layer 3D CNN produced a soft segmentation map. They extended a conditional random field [140] to a fully connected 3D CRF and used it as postprocessing to impose a regularisation constraint and produce the final hard segmentation labels.

Brebisson and Montana [135] were able to train eight networks arranged in parallel, with each network having a seven-layer CNN architecture for whole brain anatomical segmentation. The inputs of the network captured information at different scales around the voxel of interest: 3D and orthogonal 2D intensity patches captured a local spatial context, while downscaled, large 2D orthogonal patches and the distances to the regional centroids enforced global spatial consistency. They reported over-fitting, mainly due to the enormous number of parameters of the overall model and the few training images provided in the MICCAI 2012 challenge, which was a multi-atlas labelling challenge.

The second variant of multi-path architectures is the series arrangement. In this variant, different CNN networks are cascaded in series, with information from the previous network used as input for the latter, as shown in Fig. 5b. It also renders the overall CNN deeper and, as a result, increases the expressing power of the model.

Havaei et al. [24] trained a cascaded architecture, starting by training a two-path CNN. Then, they fixed the parameters of these two-path networks and used the already trained models in the cascaded architecture. They evaluated the performance of concatenating feature maps from the last layer of the first network as feature maps for the second network at three different layers. From their three architectures

Table 1
Summary of deep learning approaches for MRI brain analysis with regard to the pre-processing, architecture, post-processing and target application.

Architecture		Article	Pre-processing	Post-processing	Target
Single CNN	2D CNN	Maleki et al. 2012 [14]	Normalisation Registration Skull-stripping	–	MS lesions
	2D CNN	Zikic et al. 2014 [122]	N4ITK	–	Tumour
	2D CNN	Dvorak &Menze 2015 [130]	Normalisation N4ITK	–	Tumour
	2D CNN	Chen et al. 2015 [63]	Normalisation	–	Cerebral micro-bleed
	3D CNN	Lai 2015 [132]	Masking in normalised coordinates	Connected component analysis	Hippocampus
	2D CNN	Zhang et al. 2015 [26]	N3 Skull stripping (LABEL)	–	Tissue
	3D CNN	Milletari et al. 2016 [133]	–	–	Deep brain structure
	3D CNN	Dou et al. 2016 [16]	Normalisation	Connected component analysis	Cerebral micro-bleed
	3D CNN	Urban et al. 2016 [124]	Normalisation	Connected component analysis	Tumour
	3D CNN	Brosch et al. 2016 [21]	Normalisation Registration Skull-stripping	–	MS lesions
	2D CNN	Pereira et al. 2016 [22]	Normalisation N4ITK	Connected component analysis	Tumour
	3D CNN	Kleesiek et al. 2016 [15]	Bias field correction Resampling Intensity scaling	Connected component analysis	Skull stripping
	3D CNN	Wachinger et al. 2017 [29]	–	3D CRF	Sub-cortical structure
	3D CNN	Li et al. 2017 [134]	Normalisation	–	Deep brain structure
Multi-path CNN	2.5D CNN	Lyksborg et al. 2015 [121]	N4ITK Intensity scaling	Grow cut	Tumour
	2.5D CNN	Birenbaum and Greenspan et al. 2016 [123]	Histogram matching	–	MS lesions
	2D &3D CNN	Brebisson et al. 2015 [135]	–	–	Deep brain structure
	3D RNN	Stollenga et al. [129]	Normalisation	–	Tissue
	2D CNN	Havaei et al. 2016 [24]	Outlier removal Normalisation N4ITK	Connected component analysis	Tumour
	2D CNN	Zhao and Jia 2016 [136]	-	-	Tumour
	2D CNN	Moeskops et al. 2016 [25]	Intensity scaling Bias field correction ([137]) Skull stripping (BET)	-	Tissue
	3D FCNN	Kamnitsas et al. 2016 [23]	Normalisation Registration Skull-stripping	Fully connected 3D CRF	Tumour
	2D FCNN	Shakeri et al. 2016 [138]	–	2D CRF	Sub-cortical structure
	3D CNN	Chen et al. 2017 [27]	Intensity normalisation	-	Tissue
	2D &3D CNN	Mehta et al. 2017 [139]	N4ITK Skull stripping (BET)	–	Sub-cortical structure
	3D FCNN	Dolz et al. 2017 [28]	Bias field correction Normalisation Skull stripping (BET)	Connected component analysis	Sub-cortical structure
	3D CNN	Valverde et al. 2017 [126]	Normalisation	–	MS lesions

submitted to the BRATS 2013 brain tumour challenge, they reported the best performance by concatenating the last layer feature maps of the first network, with the actual inputs used as input for the second network.

Valverde et al. [126] presented a cascaded 3D CNN approach for addressing MS lesion segmentation. Due to the conditions of the scenario (i.e. lack of a large number of samples and data imbalance), the authors opted for performing the segmentation in two steps. In the first step, the network is trained using all the positive samples and the same number of negative samples (i.e. the negative class is under-sampled). Once the classification is performed, the probabilities for each voxel to belong to the positive class are obtained. In the second step, the network is trained with all the positive voxels and the same number of misclassified negative samples. In the paper, no details regarding the improvement of the cascade approach over the single-path architecture are quantitatively discussed. Nevertheless, with this cascaded architecture, the authors were able to outperform the rest of participants on the MICCAI 2008 challenge.

Chen et al. [27] proposed a 3D CNN, VoxResNet, for brain tissue segmentation. The network is a 3D extension of the ResNet network [5] with a shallower depth. The authors also integrated segmentation maps from VoxResNet (used as the context information) together with the

original volumes (i.e. appearance information) to train a new classifier called Auto-context VoxResNet. In this sense, possibly misclassified voxels are reprocessed and probably refined. It is important to understand that more than an "auto-context" technique, the procedure corresponds to a cascaded CNN similar to the ones described in [24,126]. Although this classifier was intended to refine the segmentation iteratively, according to the authors, the improvement was marginal and, hence, a single iteration was considered.

In general, training multiple networks, rather than a single network, could lead to better performance. The networks can be integrated in parallel or series. The former approach permits the incorporation of different pieces of information, which help in labelling the central pixel. Different networks, which extract various useful features, could be aggregated to perform the desired segmentation. Each network arranged in parallel might need different input data. Training all of the networks arranged in parallel at the same time might demand much memory. Also, managing individual networks, training processes, selecting the best hyper-parameters and avoiding over-fitting also require a significant amount of time. The latter approximation, models arranged in series, allows refining the output of one network using another one, which needs a careful design and training process. This type of architecture performs better than single networks [24].

J. Bernal et al.

(a) Multi-path architecture

(b) Cascade architecture
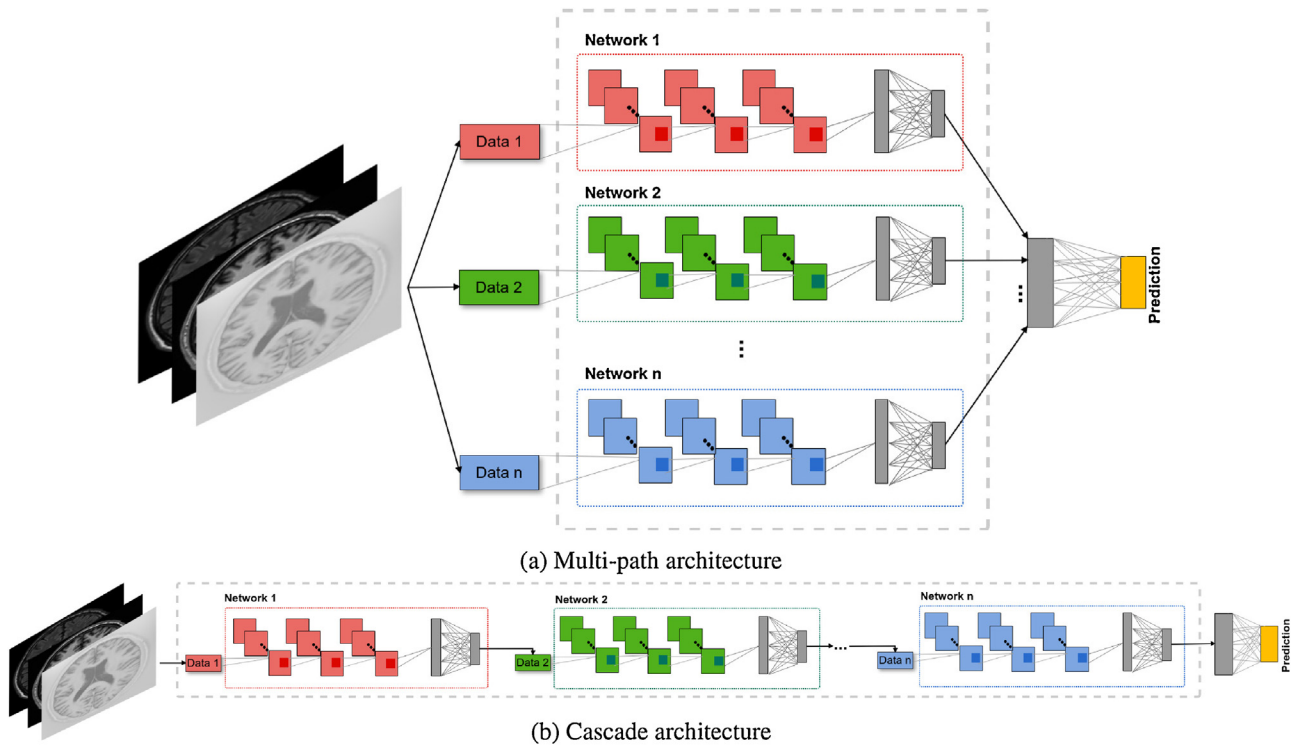
Fig. 5. Architecture of multi-path CNNs. The diagram in (a) represents a multi-path CNN architecture: different operating CNNs process in parallel input data and output a consented label. The diagram in (b) corresponds to a cascade architecture of CNN networks: the output of one network is provided as input to another one to refine the segmentation.

3.3.2. Input modalities

The strategies in the literature can also be grouped according to the number of modalities that are processed at the same time. The categories are two: single- and multi-modality. The former case is certainly more adaptable to different scenarios as a single modality is commonly provided in datasets for tissue and sub-cortical structure segmentation (mainly T1). The latter case contemplates processing different sources of information. As stated by Zhang et al. [26], it is important to consider fusing information from different imaging modalities – if possible – since it has been proven beneficial to discriminate when lacking contrast among tissues. The imaging sequences can be processed in parallel using multi-path configurations (e.g. one branch processes T1 and another one, FLAIR) or can be appended altogether to have multiple channels [26,27,121]. The difference between these two approaches is that specific kernels process each modality in the former case but, at the same time, the number of parameters increases.

3.3.3. Patch dimension

As mentioned previously, the different architectures can also be classified according to the dimension of the input patches. The categories, in this case, are three: 2D, 2.5D and 3D.

Two-dimensional architectures consider patches from a single plane (i.e. axial, sagittal or coronal). This architecture is widely used in the literature since it is adaptable to different image domains and segmentation tasks. In semantic computer vision, these types of networks are widespread. Compared to the earlier proposal from AlexNet [4], the depth of the recently proposed networks has shown a significant increase [5,6,52,141]. In MRI brain analysis, the architecture remains very similar despite being shallower, mainly because of limited available training data. It also seems unnecessary to design a very deep network if it does not improve the performance and, perhaps more importantly, it can be detrimental, due to potential over-fitting if too many network weights are required to be adjusted.

Diagnosing abnormalities employing a binary classifier is mostly the first step toward the localisation and segmentation of regions of interest. To the best of our knowledge, pioneering work using deep learning for brain image analysis was first reported in the detection of MS lesions by Maleki et al. [14]. They used a conventional CNN architecture to extract features from 2D MR FLAIR images and a three-layer neural network as a classifier. A similar classification algorithm principle could be extended to patch-wise image segmentation algorithms. Usually incorporating multi-modal images during segmentation as a stack of 2D images improves performance. Zikic et al. [122] used 2D slices from T1, T1c, T2 and FLAIR images to classify small 19×19 patches into five classes in the BRATS 2013 dataset. Their simplified CNN architecture consisted of 2D convolution, ReLU, max-pooling and softmax classifier layers.

Pixel-based labelling schemes that predict the label for a single voxel at a time have been a popular choice for training MR images using CNNs. Despite the success of this approach, scanning each voxel using a sliding window renders it computationally slow. To overcome this limitation, Dvorak and Menze [130] fed their network several versions of the whole 2D slice input image, shifted on the *X* and *Y* axes, and they merged the outputs accordingly. They used images from multi-modal volumes to train a CNN for the prediction of extended label patches. As mentioned earlier, in the context of brain MRI, building a very deep network is a rare practice because of the difficulty of obtaining sufficient representative and accurate data for training. In spite of the performance advantages, deeper networks usually have a larger number of parameters, and they may tend to suffer from over-fitting. Training on large datasets with proper regularisation algorithms is a common practice in the community. Pereira et al. [22] proposed a deep architecture, eleven layers, with 2D patches extracted from T1, T1c, T2 and FLAIR for gliomas segmentation. They made use of small 3×3 kernels which allowed designing a deeper architecture [141]. A CNN built using those small kernels has fewer parameters, and therefore reduces the over-fitting problem. Moreover, they also took advantage of data augmentation using rotation to increase their original training dataset.
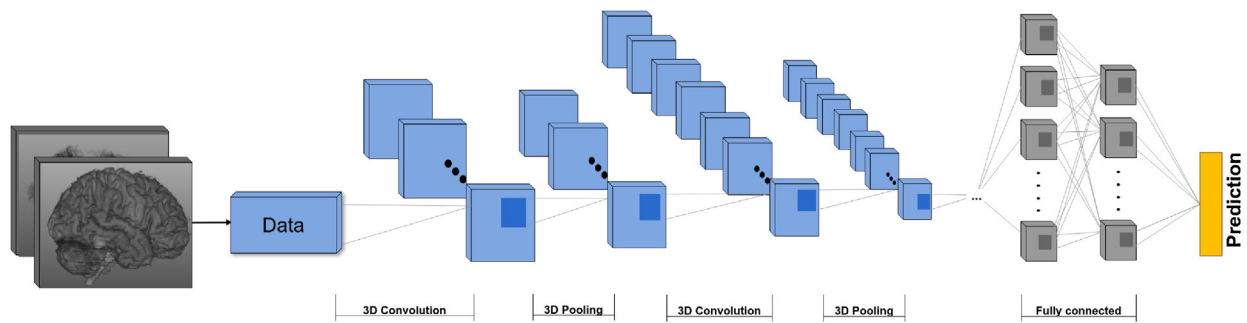
Fig. 6. Architecture of 3D CNNs. The convolution operation is performed by employing 3D filters. The pooling operation is also performed in three dimensions.

Their 2D CNN consisted of multiple 2D convolutions, followed by LeakyReLU and max-pooling layers.

In the reviewed literature, we observed that the conventional CNN models are popular choices of MRI analysis researchers, perhaps because of the authors' interest in utilising the already developed 2D CNN architectures, in addition to reducing computational complexity. This approach is in contrast to the actual 3D nature of the volume.

Tri-planar or 2.5D architectures are provided with patches from the three anatomical planes (i.e. axial, sagittal and coronal), commonly using a multi-path design. As explained by Lyksborg et al. [121], the 2.5D information provides a better understanding of the 3D scenario than 2D-based networks since it exploits the 3D nature of MR images and, consequently, brings up contextual information. Moreover, although 2.5D architectures are more computationally expensive than 2D variants, these networks still consider 2D convolutions and, hence, are expected to be less costly than 3D alternatives [130,132].

Lyksborg et al. [121] presented a 2.5D multi-path approach in which the patches from different planes were processed independently by similar operative modules. By using probabilistic methods, the knowledge coming from the three branches was combined after the prediction layer. Although the approach is reliable, it somehow assumes that the contribution among the three branches is the same, which may not be the case at some point (i.e. due to the anisotropic nature of the data and/or to the fact that regions of interest may be more visible in specific planes). A workaround to this situation is to perform the merging step before the prediction layer and, hence, the weight of the features to be merged are specifically fixed for the case of study since the network determines the best configuration based on their contribution to the final segmentation.

Three-dimension architectures consist in taking 3D segments directly from the MRI volume. This architecture utilises 3D convolution kernels, which seems to be a more appropriate solution for fully exploiting the spatial contextual information in volumetric data. The main constraint of a 3D CNN approach lies in its expensive computational cost, memory requirements and computational time. Utilising the whole volume at a time is computationally very expensive; therefore, small, three-dimensional patches are extracted for training and testing. For instance, Urban et al. [124] used four (multi-modal) 9^3 voxels as input to segment a tumour in the BRATS 2013 dataset. Fig. 6 shows a general 3D CNN architecture, in which the input is volumetric data, a 3D brain MR image in this case, and 3D kernels are used to perform the convolution.

Lai and Rueckert [132] investigated the use of three different convolutional network architectures for patch-based segmentation in ADNI hippocampus MRI dataset. They evaluated stacked 2D patches (one around the voxel in question, one in parallel and above it, and one in parallel and below it), tri-planar patches and 3D patches. According to their experiments, 3D convolutional architectures performed the best despite being the most computationally expensive.

To summarise, most of the algorithms, 10 of the 27, reviewed in the literature used a 2D CNN architecture. They were based on the two-dimensional convolutions used extensively in computer vision applications on natural images. This fact eases the model development process for practitioners by customising the popular CNN architectures and exploiting the available frameworks. Although patch-wise implementation of these types of architectures is very slow, converting the last output layer into a convolutional layer helps to expedite the inference time [24]. An implementation using a single network might not result in good performance, demonstrating that deeper networks that use smaller kernels achieve better performance, compared to using larger kernels and shallower architecture [22]. Two-dimensional CNNs disallow for exploiting of the actual 3D nature of the MRI data. In this sense, 2.5D and 3D CNNs were introduced mainly to overcome this limitation. Although both 2.5D and 3D architectures require more processing than 2D ones, the former type is less computationally expensive than the second one: 2D convolutional layers are used in 2.5D while 3D convolutional are used in 3D. Indeed, the main impediments to working with 3D CNNs are the number of parameters, memory requirements, expensive computational costs and slow speed. However, with the rapid development of GPUs, this situation may not represent a drawback in the coming years.

3.3.4. Predictions at a time

The approaches can also be grouped according to the number of predictions they perform at a time into CNN and FCNN [28,23,138]. The former case corresponds to the traditional approach in which a single patch is processed by the network, and a single output is returned. Since the idea at the end is to obtain segmented areas, this approach can be slow in practice. The latter case corresponds to the architectures for which the fully connected layers are replaced by 1×1-kernel (or $1 \times 1 \times 1$-kernel) convolutional layers to obtain a dense prediction. Additionally, some approaches consider upsampling layers to increase the number of predictions at a time. Naturally, the main drawback of these approaches is that a larger number of parameters are required to be set up (hence, more training samples to train the network properly are needed) in comparison to CNN strategies.

Dolz et al. [28] implemented a 3D FCNN architecture which was used to segment sub-cortical structures. Apart from the FCNN implementation, the authors considered multi-resolution information by extracting feature maps from high-resolution layers and merging them with low-resolution information, coming from the main information flow, before the $1 \times 1 \times 1$-kernel convolutional layers. As explained by Bharath [142], this kind of connections allowed the network to learn from semantic – coming from deeper layers – as well as fine-grained localisation information – coming from shallow layers.

Ronneberger et al. [143] proposed a multi-path 2D FCNN architecture called the U-Net. The network was successfully tested on the 2D electron microscopy [144] and ISBI 2015 cell tracking[6] challenges, achieving the first position in both of them. There were two key

---

[6] The challenge web page is http://www.codesolorzano.com/Challenges/CTC/Welcome.html

components on this FCNN. First, up-sampling layers were used to increase the resolution of the output. Thus, the volumes could be rapidly processed. In the paper, the authors were able to produce an output map of 388 × 388 from an input image of size 572 × 572. Second, localisation and context were retained in the network by merging features maps from high-resolution layers with the up-sampled ones. On the other hand, it is important to highlight that this type of network, as stated by its authors, requires setting up a large number of parameters. Indeed, the original implementation required data augmentation techniques (for example, non-linear deformations generated on-the-fly during training) to achieve desired segmentation results.

Several approaches in different scenarios have been inspired by the U-Net architecture. Milletari et al. [79] and Yu et al. [145] considered a 3D extension of the U-Net along with skip and/or residual connections for addressing the MICCAI Prostate MR Image Segmentation (PROMISE12). According to their experiments, the use of links improved the accuracy of the methods and also proved the effectiveness of information propagation. Also, the 3D variant has been implemented for confocal microscopic data [146] and liver and lesion segmentation in CT images [147]. Although none of these papers is directly applied to brain MRI, we consider that they could achieve relevance for brain MRI segmentation as well.

Brosch et al. [21] used 3D convolutional encoder networks for MS lesion segmentation. In addition to learning deconvolution and unpooling, they introduced short-cut connections from the first to the last deconvolution layers inspired by the U-Net architecture [143]. The network in the convolutional encoder path was initialised using weights pre-trained on a stack of convolutional restricted Boltzmann machines. With this approach, they achieved top performance in public MS lesion segmentation challenge datasets.

3.3.5. Contextual information

The fact that the discussed architectures process patches separately and not the whole volume has some implications. For instance, the spatial distribution of brain structures is not directly encoded. Although some implicit contextual information is encoded in 2D, 2.5D and 3D patches their information is limited to the size of the patches. The bigger the size of the patch, the more information the network can take into account to produce the prediction; but the more the parameters to be trained. When considering the architecture of Havaei et al. [24] and the work of Kamnitsas et al. [23], it seems to show that the context is important, but to preserve spatial correspondence of the multi-scale features is crucial. This could explain why a simple approach as [22] could outperform [24].

Recently, Wachinger et al. [29] introduced explicit within-brain location information through Cartesian and spectral coordinates [148,149] to help the classifier to discriminate one class from another. The Cartesian coordinates were obtained by taking the XYZ-coordinates of each of the voxels while the spectral coordinates came from the eigenvectors of the Laplacian matrix of the graph representing the brain mask. As explained by the authors in [150], the explicit spatial information is useful to (i) discriminate between patches from different brain hemispheres since patch-based strategies loss spatial context and (ii) help in the segmentation of tissues exhibiting low contrast. In this latter approach, it is essential to have in mind that (i) considering XYZ-coordinates on the image plane makes sense only if the volumes are registered and (ii) the distribution of the eigenvectors depends highly on the shape of the brain mask, i.e. these features may not be reliable when the shape of the region of interest differs considerably from one volume to another.

3.3.6. Summary of pros and cons of the different strategies

Table 2 summarises the main advantages and disadvantages of the approaches discussed previously in this section.

3.4. Post-processing

Post-processing is the step to refine or improve the results of the segmentation. The purpose of post-processing could be either to create hard segmentation labels [23] or to remove false positives [16,26,22,24,124,132] from segmentation results, using different algorithms such as connected component analysis and Conditional Random Fields (CRF). The former strategy consists in keeping only the *k* largest areas from the segmentation to remove spurious outputs. Nevertheless, in sub-cortical structure segmentation these spurious areas may have a larger volume than the small structures of interest and, hence, it becomes impractical. The latter strategy, CRF model, considers not only the given voxel but also its neighbours to produce a refined label, especially around the edges. Although CRF-based methods require learning parameters, they could bring regularisation properties to the segmentation since they are based on the minimisation of an energy equation. This stage helps to improve the overall performance of the model at the expense of adding computational complexity.

Contrast-driven post-processing techniques may lead to unexpected/unsatisfactory smoothing results. For instance, sub-cortical structures and some parts of the brainstem could be shrunk in this step due to low contrast with surrounding areas as shown in Fig. 7. However, this does not signify that the method is not capable of refining the segmentation since both the intensity features and the prior probabilities take part in the process.

In addition to CRF and connected component analysis, Lyksborg et al. [121] utilised an algorithm called Grow cut [151]. It was initially proposed as a continuous state cellular automata method for automated segmentation based on user labelled seed voxels. The authors used it to refine the initial tumour segmentation results. The post-processing algorithm was applied iteratively until a stable segmentation was obtained.

4. Evaluation

Brain MRI volume segmentation approaches are usually evaluated using different quantitative measurements. The described CNN models were tested in public and clinical research trial datasets. Regarding the works reviewed, both the most common datasets and the standard measurements used for the evaluation are described in this section. Moreover, the results obtained by the analysed methods on the standard datasets are presented.

### 4.1. Public evaluation frameworks and datasets

Several tissue segmentation algorithms have been proposed during the last years in the literature. However, determining which of them achieves the best performance can be complicated in principle since not all the algorithms are publicly available. A way to address this situation is by creating standard evaluation frameworks, such as the grand challenges taking place on relevant international conferences. Thus, before delving into a detailed discussion about the methods, we describe some of the most widely used public datasets for brain lesions, structures and anatomical segmentation.

Some of the most commonly used public datasets in brain image MRI analysis are summarised in Table 3 . The reviewed works primarily used MICCAI datasets for MS lesions, tumours, tissue and structure segmentation. For MS lesion segmentation, the challenge was created in conjunction with MICCAI 2008.[7] The challenge was part of the 3D segmentation in the Clinic Grand Challenge II and consisted of 54 brain MRI images, 20 of which are available for training with their corresponding ground truth. For brain tumour segmentation, the series of multi-modal brain tumour segmentation (BRATS) challenges are

---

[7] http://www.ia.unc.edu/MSseg/index.html

J. Bernal et al.

Table 2
Pros and cons of the segmentation strategies presented.

Strategy	Advantage	Disadvantage
Interconnected operating modules		
Single path	In principle, fast computation	Single flow of information
Multi-path – parallel	Extract more varied features	More parameters than single-path architectures
	Verdict is consented among interconnected modules	More computationally demanding (data preparation and processing)
	Incorporated information may provide the network with contextual information (e.g. multi-resolution architectures)	
Multi-path – series	Extract more varied features	Requires careful design of the network
	Allows refining information at different stages	May require training different networks
		The improvement could be marginal
Input modalities		
Single modality	Adaptable to different scenarios	Single source of information
	Easily extensible to multiple modalities (channels or multi-path)	
Multi-modality	Gain valuable contrast information	Requires more parameters than single modality
Patch dimension		
2D	Easily scalable to complex network architectures	Heavily dependent on the network design tricks to obtain good results
	Flexibility and adaptability	Disallow 3D nature of MRI
	Fast computation	
2.5D	Faster computation than 3D	Computationally more expensive than 2D
	Exploits the 3D nature of MRI	
	Acquires implicit contextual information	
3D CNN	Exploits 3D nature of the MRI volume directly	Computationally expensive
	Usually leads to better performance than 2D	Scaling to larger patch size may be computationally demanding
	Acquires implicit contextual information	Due to the number of parameters, it may require large training data
Predictions at a time		
CNN	In principle, less number of parameters than FCNN	Single classification for a single patch
FCNN	Faster segmentation (some architectures can classify a single volume in one shot)	Larger number of parameters to be set up
		Generally, requires more training samples (data augmentation is commonly adopted in this sense)
Contextual information		
Implicit	No additional calculations are required	Context depends on the size of the patch
Explicit	Discriminative features are provided to the network	Subject to registration accuracy or shape of the region of interest
	Sense of within-brain positioning	
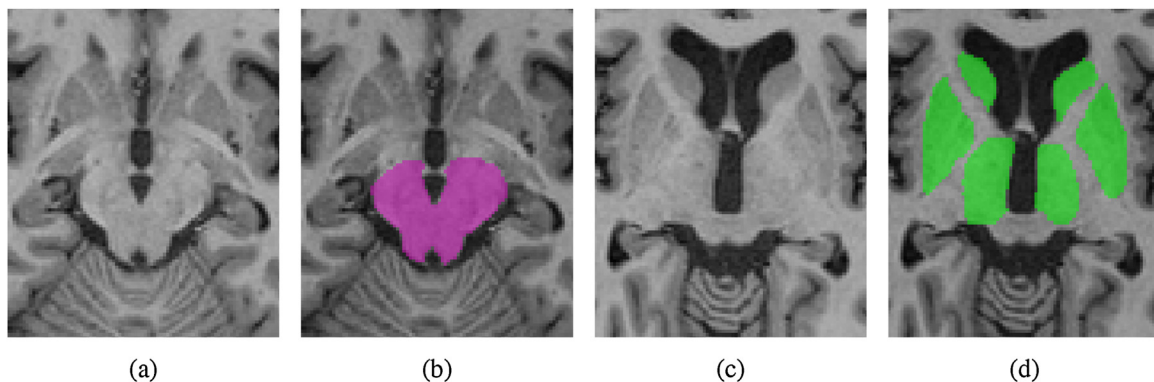
Fig. 7. Two input slices in which low contrast between the region of interest and the surrounding areas is exhibited. In the image (a) the focus is on the mesencephalon which is highlighted by the purple area in (b). In (c) the focus is on caudate, putamen, pallidum and thalamus structures presented in green in (d).

commonly used. These challenges started in 2012 with 80 cases of real and synthetic data and, every year, the size of the training and testing data has been enlarged (up to 300+ cases in total). For sub-cortical structure segmentation, the MICCAI 2012[8] challenge in multi-atlas labelling and the Internet Brain Segmentation Repository (IBSR)[9] are widely used. It is important to note that MICCAI 2012 challenge is initially intended to evaluate algorithms on whole-brain structure segmentation, but naturally, the labels can be ignored and merged to obtain the sub-cortical structures only. Additionally, the IBSR dataset can be used for both sub-cortical structures and brain tissue segmentation.

Other datasets widely adopted to evaluate algorithms for tissue segmentation are two the MICCAI Grand Challenges MRBrainS13.[10] and NeoBrainS12[11] The difference between the two datasets is that in the former case, the algorithms are evaluated using adults exhibiting WM lesions and, in the latter case, the data comes from neonatal subjects.

The main evaluation measures for the challenges mentioned previously are DSC, specificity, sensitivity, positive predictive value (precision), average surface distance (ASD), average volumetric difference (AVD) and modified Hausdorff distance (MHD).

---

[8] https://masi.vuse.vanderbilt.edu/workshop2012/index.php/Challenge_Details
[9] Available at http://www.nitrc.org/projects/ibsr
[10] http://mrbrains13.isi.uu.nl/index.php
[11] http://neobrains12.isi.uu.nl

Table 3

Summary of commonly used public datasets in MS, tumour, tissue and structure segmentation. The table is structured as follows. The different datasets are listed in the first column and their corresponding information: number of scans, offered modalities, acquisition scanner, segmentation tasks in which it is used, and image information such as reconstruction matrix and pixel spacing are detailed from column two to seven.

Name	Number of scans		Modality	Scanner	Segmentation tasks	Reconstruction matrix	Pixel spacing (mm)
	Train	Test					
MICCAI 2008 [152]	20	24	T1, T2, and FLAIR	3T Siemens Allegra	MS lesion	$512 \times 512 \times 512$	$0.5 \times 0.5 \times 0.5$
MICCAI 2012	15	20	T1	–	Whole brain, tissue and sub-cortical structure	$512 \times 512 \times 512$	$0.5 \times 0.5 \times 0.5$
NeoBrainS12 [153]	20	5	T1 and T2	3T Philips Achieva	Tissue	$384 \times 384 \times 50$ $512 \times 512 \times 110$ $512 \times 512 \times 50$	$0.34 \times 0.34 \times 2.0$ $0.35 \times 0.35 \times 1.2$ $0.35 \times 0.35 \times 2.0$
IBSR18	18	–	T1	–	Tissue and sub-cortical structure	$256 \times 128 \times 256$	$0.84 \times 0.84 \times 1.50$ $0.94 \times 0.94 \times 1.50$ $1.00 \times 1.00 \times 1.50$
MRBrainS13 [154]	5	15	T1,T1_1mm, T1_IR and FLAIR	3T Philips Achieva	Tissue	$256 \times 256 \times 192$ $240 \times 240 \times 48$	$0.958 \times 0.958 \times 3.0$ T1_1mm has $1 \times 1 \times 1$
BRATS 2012 [155]	80	30	T1, T1c, T2, and FLAIR	–	Tumour	$160 \times 216 \times 176$ $176 \times 176 \times 216$	$1 \times 1 \times 1$
BRATS 2013 [155]	30	25	T1, T1c, T2, and FLAIR	–	Tumour	$160 \times 216 \times 176$ $176 \times 176 \times 216$	$1 \times 1 \times 1$
BRATS 2014 [155]	309	–	T1, T1c, T2, and FLAIR	–	Tumour	$160 \times 216 \times 176$ $176 \times 176 \times 216$	$1 \times 1 \times 1$
BRATS 2015 [155]	274	110	T1, T1c, T2, and FLAIR	–	Tumour	$240 \times 240 \times 150$	$1 \times 1 \times 1$
ISBI 2015 [156]	21	61	T1, T2, PD, and FLAIR	3T MRI	Longitudinal MS lesion	$181 \times 217 \times 181$	$1 \times 1 \times 1$
ISLES 2015 [157]	28	36	FLAIR, T2 TSE, T1 TFE/ TSE, DWI	3T Philips	Ischemic stroke lesion	$230 \times 230 \times 154$	$1 \times 1 \times 1$
	30	20	T1c, T2, DWI, CBF, CBV, TTP, Tmax	1.5T Siemens Magnetom Avanto 3T Siemens Magnetom Trio		–	$2 \times 2 \times 2$
MSSEG 2016 [158]	15	38	T1, T1 GADO, FLAIR, DP/T2	3T GE Discovery 3T Philips Ingenia 1.5T Siemens Aera 3T Siemens Verio	MS lesion	$176 \times 256 \times 256$ $256 \times 256 \times 176$ $200 \times 336 \times 336$ $210 \times 336 \times 336$	$1 \times 1 \times 1$ $1.08 \times 1.08 \times 0.9$ $0.85 \times 0.74 \times 0.74$
LBPA40	40	–	T1	1.5T GE	Brain structure	$181 \times 217 \times 181$	$1 \times 1 \times 1$
Hammers95n30	30	–	T1	1T Philips HPQ	Brain structure	Varied	$0.94 \times 0.94 \times 0.94$
Hammers83n30	30	–	T1	1T Philips HPQ	Brain structure	Varied	$0.94 \times 0.94 \times 0.94$
Hammers67n20	20	–	T1	1T Philips HPQ	Brain structure	Varied	$0.94 \times 0.94 \times 0.94$

4.2. Evaluation measurements

Regarding the application, we found that the CNN models were applied for MS lesion segmentation, brain tumour segmentation and structure segmentation. Although there are public datasets available for these applications, researchers still prefer to work on their databases. One main reason might be the limitation of the number of training samples available in the public datasets. As stated before, working with a small dataset affects the performance of deep CNN models enormously and imposes restrictions on practitioners from fully exploiting the capacity of those algorithms.

The evaluation measurements compare the output of segmentation algorithms with ground truth in either a pixel-wise or a volume-wise basis. For a more robust assessment, more than one expert might be involved in generating the ground truth volumes to avoid inter-observer variability. In the reviewed literature, the results were evaluated using different evaluation metrics. In cases of MS and tumour segmentation, the most common evaluation measurements are DSC, precision, recall, the true positive rate (TPR) and the positive predictive value (PPV). Also, absolute volume difference, lesion-wise true positive rate (LTPR) and the lesion-wise false positive rate (LFPR) are also employed. Similarly, for brain tumour and structure segmentation applications, the DSC score, precision, and recall are among the widely used measurements. One main reason for this choice is that the challenge organisers primarily rely on these results. Table 4 summarises the types of databases, numbers of samples, modalities considered, evaluation measurements applied and corresponding results reported of the surveyed works.

5. Discussion and future directions

Deep CNNs have made a large impact in a broad range of application domains. Today, they are the first choice to solve many problems in computer vision, speech recognition and natural language processing. Taking computational advantage of working with small 2D and 3D patches, rather than the entire slice or volume, researchers in brain MR image analysis can train deep CNNs to obtain accurate segmentation algorithms. This success has received overwhelming acceptance by the community, in which shallow architectures were predominantly used. Currently, most CNN architectures have many layers, including additional normalisation layers, such as batch normalisation. Furthermore, each architecture is becoming increasingly more sophisticated, employing ideas from optimisation and probabilistic models.

About the architectures and their performances, the majority of the proposed works in Table 1 used 2D networks. Although working with each model has its computational advantages and drawbacks, obtaining good generalisation requires an architecture with optimised layers, considering the class imbalance and selecting the best hyper-parameters and advanced training procedures. From the results shown in Table 4, methods with 2D CNN architecture with sufficient depth [22], cascade [24], "short-cut connections" [21] and parallel networks [25,135] showed top performance in their respective applications. After all, there is no universal architecture, and the ongoing research contributes to obtaining a model that can learn to provide a good representation of the underlying input image without suffering from significant over-fitting.

In the reviewed works, we observed that most of the proposed methods emphasised the shortcomings of working with deep CNN models. First, there is the computational requirement. Analysing,

Table 4

Summary of results in the reviewed papers. The acronyms for the sequences stand for: susceptibility-weighted imaging (SWI), quantitative susceptibility mapping (QSM), proton density (PD), fractional anisotropy (FA), magnetisation prepared rapid gradient echo (MPRAGE) and gradient echo (GE). The acronyms in the measurements stand for: modified Hausdorff distance (MHD), positive predictive (PP) and average symmetric surface distance (ASSD). The acronyms in results are: grey matter (GM), white matter (WM), cerebrospinal fluid (CSF), dataset from University of North Carolina (UNC) and dataset from Children's Hospital Boston (CHB).

Article	Dataset	Number of samples			Modality	Classes	Evaluation measures	Result
		Train	Val	Test				
Lesion								
Maleki et al. [14]	Clinical trial	152	–	–	FLAIR	2	Sensitivity	96.1%
							Specificity	97.5%
							Accuracy	92.9%
Brosch et al. [21]	MICCAI 2008	20	-	23	T1, T2, PD and FLAIR	2	VD	UNC 63.5%, CHB 52.0%
							TPR	UNC 47.1%, CHB 56.0%
							FPR	UNC 52.7%, CHB 49.8%
	ISBI 2015	20	1	61	T1, T2, PD and FLAIR	2	DSC	68.3%
							LTPR	78.3%
							LFPR	64.5%
	Clinical trial	250	50	77	T1, T2 and FLAIR	2	DSC	63.8%
							LTPR	62.5%
							LFPR	36.2%
							VD	32.9%
Chen et al. [63]	Clinic data	55	–	55	–	2	Sensitivity	89.0%
							Precision	56.0%
							F1-score	0.69
							Average FP	6.4
Dou et al. [16]	Clinical data	924	108	117	SWI	2	Sensitivity	93.16%
							Precision	44.13%
							Mean FP	2.74
Birenbaum and Greenspan [123]	ISBI 2015	21	–	61	T1, T2, PD and FLAIR	2	DSC	62.7%
Valverde et al. [126]	MICCAI 2008	20	–	23	T1, T2, PD and FLAIR	2	VD	UNC 62.5%, CHB 48.8%
							TPR	UNC 55.5%, CHB 68.7%
							FPR	UNC 46.8%, CHB 46.0%
	Clinical trial 1	35	–	–	T1, T2 and FLAIR	2	DSC	53.5%
							VD	30.8%
							TPR	77.0%
							FPR	30.5%
							PPV	70.3%
	Clinical trial 2	25	–	–	T1, T2 and FLAIR	2	DSC	56.0%
							VD	27.5%
							TPR	68.2%
							FPR	33.6%
							PPV	66.1%
Tumour								
Zikic et al. [122]	BRATS 2013	20	–	–	–	5	DSC	Complete 83.7%, Core 73.6%, Enhanced 69.0%
Urban et al. [124]	BRATS 2013	30	–	15	T1, T1c, T2 and FLAIR	5	DSC	Complete 87.0%, Core 77.0%, Active 73.0%
Dvorak &Menze [130]	BRATS 2014	166	25	66	T1,T1c,T2 and FLAIR		DSC	Complete 83%, Core 75%, Active 77%
Lyksborg et al. [121]	BRATS 2014	91	-	187	T1, T2, PD, and FLAIR	2	DSC	Complete 79.9%, Core 63.1%, Enhancing 62.5%
							Positive predictive	Complete 0.78, Core 0.63, Enhancing 0.58
							Sensitivity	Complete 0.86, Core 0.74 Enhancing 0.78
Pereira et al. [22]	BRATS 2013	30	–	25	T1, T1c, T2 and FLAIR	5	DSC	Complete 88%, Core 83%, Enhanced 77%
							PPV	Complete 0.85, Core 0.82, Enhanced 0.60
							Sensitivity	Complete 0.86, Core 0.76, Enhanced 0.68
	BRATS 2015	30	–	25	T1, T1c, T2 and FLAIR		DSC	Complete 78.0%, Core 65.0%, Enhanced 75.0%
Kamnitsas et al. [23]	Clinical data	46	–	15	MPRAGE, FLAIR, T2, PD and GE	2	DSC	64.5%
							Precision	69.8%
							Sensitivity	63.9%
							ASSD	3.72
							HD	52.38
	BRATS 2015	274	-	110	T1, T1c, T2 and FLAIR	5	DSC	Complete 84.9%, Core 66.7%, Enhanced 63.4%
							Precision	Complete 85.3%, Core 86.1%, Enhanced 63.4%
							Sensitivity	Complete 87.7%, Core 60.0%, Enhanced 67.4%
	ISLES 2015	28	–	36	T1, T1c, FLAIR and DWI	2	DSC	59.0%
							Precision	68.0%
							Sensitivity	60.0%
							ASSD	7.87
							HD	39.61

14

Table 4 (*continued*)

Article	Dataset	Number of samples			Modality	Classes	Evaluation measures	Result
		Train	Val	Test				
Havaei et al. [24]	BRATS 2013	30	–	25	T1, T1c,T2 and FLAIR	5	DSC Sensitivity Specificity	Complete 84.0%, Core 71.0%, Enhancing 57.0% Complete 0.88, Core 0.79, Enhancing 0.54 Complete 0.84, Core 0.72, Enhancing 0.68
Zhao and Jia [136]	BRATS 2013	80	–	30	T1, T1c,T2 and FLAIR	5	DSC	81.0% ± 9.5%
Tissue								
Zhang et al. [26]	Clinic data	7	1	–	T1, T2 and FA	3	DSC MHD	85.03% ± 2.27% 0.32 ± 0.12
Stollenga [129]	MRBrainS13	5	–	15	T1, FLAIR and T1_IR	5	DSC MHD AVD	WM 88.33%, GM 84.82%, CSF 83.72% WM 2.07, GM 1.69, CSF 2.14 WM 7.05, GM 6.77, CSF 7.10
Moeskops et al. [25]	NeoBrainS12	-	-	-	T2	8	DSC	Cor. 30 wks 82.7%, Ax. 40 wks 80.5%, Cor. 40 wks 81.9%
	MRBrainS13	5	–	15	T1	7	Mean DSC Mean ASD	85.86% 0.93
	MICCAI 2012	15	–	20	T1	6	Mean DSC Mean ASD	90.66% 0.50
Chen et al. [27]	MRBrainS13	5	-	15	T1, T1_IR and FLAIR	4	DSC MHD AVD	WM 89.39%, GM 86.12%, CSF 83.96% WM 1.94, GM 1.47, CSF 2.28 WM 5.84, GM 6.42, CSF 7.44
Structure								
Lai [132]	ADNI	120	40	40	–	3	Error rate	7.21%
Brebisson et al. [135]	MICCAI 2012	15	-	20	T1	134	Mean DSC	72.5%
Milletari et al. [133]	Clinical data	45	–	10	QSM	26	Mean DSC	77.0%
Shakeri et al. [138]	IBSR18	18	–	–	T1	5	Mean DSC	82.40%
Mehta et al. [139]	MICCAI 2012	15	–	20	T1	134	Mean DSC	74.30%
	IBSR18	9	–	9	T1	32	Mean DSC	84.4%
	LONI-LPBA40	20	–	20	T1	54	Mean DSC	82.4%
	Hammers67n20	10	–	10	T1	67	Mean DSC	84.0%
	Hammers83n30	15	–	15	T1	83	Mean DSC	80.8%
Wachinger et al. [29]	MICCAI 2012	15	–	15	T1	25	Mean DSC	90.6%
Dolz et al. [28]	IBSR18	12	3	3	T1	4	Mean DSC	90.0%
	ABIDE	150	15	947	T1	4	Mean DSC	90.0%
Li *et al.* [134]	ADNI	443	50	50	T1	155	DSC	84.34 ± 1.89
Skull stripping								
Kleesiek et al. [15]	IBSR	18	–	–	T1	2	DSC	95.77 ± 0.01
	LPBA40	40	–	–	T1	2	Sensitivity	94.25 ± 0.03
	OASIS	77	–	–	T1	2	Specificity	99.36 ± 0.003
	Clinical data	53	–	–	T1, T1c, T2, FLAIR	2	DSC Sensitivity Specificity	95.19 ± 0.01 96.25 ± 0.02 99.24 ± 0.003

manipulating and processing each voxel in a volume is expensive computationally. The enormous amount of memory needed to store the extracted patches and a large amount of time required to process them constitute a difficulty. The deep learning software libraries used to implement layers of deep CNNs have either parallel or distributed frameworks, which help researchers to train their models in multi-core architectures or GPUs. Second, training CNN models for brain image analysis is hindered by the data imbalance problem, especially with small lesion or structure segmentation. For instance, in tumour or MS lesion segmentation, obtaining good generalisation is a challenge as most of the lesions are smaller than the entire volume. Two-phase training [24], careful patch selection [22,25,132] and loss functions [21,159,160] are among the proposed strategies to overcome this problem.

Although we observed the success of CNNs; their full capacity has not yet been fully leveraged in brain MRI analysis. Networks can perform exceptionally when trained and tested on images with similar acquisition characteristics (e.g. resolution, intensity range), but when tested on sets with slight variations concerning the training set, they tend to fall behind traditional methods like FAST [161] and SPM [162,163]. This is the leading cause why traditional approaches are still being used in medical centres due to their robustness and adaptability [164]. Even though there is progress in domain adaptation techniques, more research in this sense is needed before being able to use resilient CNN-based software on medical centres.

It is becoming a widespread practice in the computer vision community to release source codes to the public. This practice helps to expedite the research in the field. The most commonly used deep learning libraries for MRI segmentation are Tensorflow,[12] Theano,[13] Caffe,[14] Keras[15] and PyTorch.[16] Also, open-source frameworks with implementation of major CNN approaches can be found online, such as NiftyNet[17] and DLTK.[18] Another recommended practice is validating the model on different datasets. Few works [15,21–23,25,139] have reported their results on three or more different public datasets. This practice opens the door to design a robust model that can be applied to datasets of similar applications but with different types of MRI scanners, imaging modalities and numbers of training cases.

With the lack of training data, the poor spatial resolution of MR images and the need for a short prediction time, it has been impossible to train considerably deep CNNs. To train such networks, a considerable

---

[12] http://www.tensorflow.org/
[13] http://deeplearning.net/software/theano/
[14] http://caffe.berkeleyvision.org/
[15] https://keras.io/
[16] http://pytorch.org/
[17] http://www.niftynet.io/
[18] http://dltk.github.io/

effort is needed in designing faster methods to perform convolutions. FFT algorithms [21] and quicker matrix multiplication methods [165] have been used to improve the computation speed of CNNs, but there is yet room for improvement in the training algorithms of deep CNNs using variants of SGD [166] and their parallelised and distributed implementations. The new algorithms to come are expected not only to improve the performance of deep CNNs but also to be highly optimised, with less or no hyper-parameters, which constitute one of the major bottlenecks for most users to tune.

Acknowledgements

References

[1] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. Imagenet large scale visual recognition challenge. Int J Comput Vis 2015;115(3):211–52.

[2] Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft coco: common objects in context. European conference on computer vision. 2014. p. 740–55.

[3] Everingham M, Eslami SA, Van Gool L, Williams CK, Winn J, Zisserman A. The Pascal visual object classes challenge: a retrospective. Int J Comput Vis 2015;111(1):98–136.

[4] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 2012:1097–105.

[5] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition 2016:770–8.

[6] Szegedy C, Ioffe S, Vanhoucke V, Alemi AA. Inception-v4, inception-resnet and the impact of residual connections on learning. Proceedings of the thirty-first AAAI conference on artificial intelligence. 2017. p. 4278–84.

[7] Noh H, Hong S, Han B. Learning deconvolution network for semantic segmentation. Proceedings of the IEEE international conference on computer vision 2015:1520–8.

[8] Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL. Semantic image segmentation with deep convolutional nets and fully connected CRFs. International conference on learning representations 2015 http://arxiv.org/abs/1412.7062.

[9] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd international conference on machine learning 2015:448–56.

[10] Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, et al. Dermatologist-level classification of skin cancer with deep neural networks. Nature 2017;542(7639):115–8.

[11] Hasegawa A, Lo S-CB, Freedman MT, Mun SK. Convolution neural-network-based detection of lung structures. Medical imaging 1994, International Society for Optics and Photonics 1994:654–62.

[12] Lo S-C, Lou S-L, Lin J-S, Freedman MT, Chien MV, Mun SK. Artificial convolution neural network techniques and applications for lung nodule detection. IEEE Trans Med Imaging 1995;14(4):711–8.

[13] Sahiner B, Chan H-P, Petrick N, Wei D, Helvie MA, Adler DD, et al. Classification of mass and normal breast tissue: a convolution neural network classifier with spatial domain and texture images. IEEE Trans Med Imaging 1996;15(5):598–610.

[14] Maleki M, Teshnehlab M, Nabavi M. Diagnosis of multiple sclerosis (MS) using convolutional neural network (CNN) from MRIs. Glob J Med Plant Res 2012;1(1):50–4.

[15] Kleesiek J, Urban G, Hubert A, Schwarz D, Maier-Hein K, Bendszus M, et al. Deep MRI brain extraction: a 3D convolutional neural network for skull stripping. Neuroimage 2016;129:460–9.

[16] Dou Q, Chen H, Yu L, Zhao L, Qin J, Wang D, et al. Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks. IEEE Trans Med Imaging 2016;35(5):1182–95.

[17] Roth HR, Lu L, Seff A, Cherry KM, Hoffman J, Wang S, et al. A new 2.5 D representation for lymph node detection using random sets of deep convolutional neural network observations. International conference on medical image computing and computer-assisted intervention. 2014. p. 520–7.

[18] Sirinukunwattana K, Raza SEA, Tsang Y-W, Snead DR, Cree IA, Rajpoot NM. Locality sensitive deep learning for detection and classification of nuclei in routine

[19] Anthimopoulos M, Christodoulidis S, Ebner L, Christe A, Mougiakakou S. Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. IEEE Trans Med Imaging 2016;35(5):1207–16.

[20] Albarqouni S, Baur C, Achilles F, Belagiannis V, Demirci S, Navab N. AggNet: deep learning from crowds for mitosis detection in breast cancer histology images. IEEE Trans Med Imaging 2016;35(5):1313–21.

[21] Brosch T, Tang LY, Yoo Y, Li DK, Traboulsee A, Tam R. Deep 3D convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation. IEEE Trans Med Imaging 2016;35(5):1229–39.

[22] Pereira S, Pinto A, Alves V, Silva CA. Brain tumor segmentation using convolutional neural networks in MRI images. IEEE Trans Med Imaging 2016;35(5):1240–51.

[23] Kamnitsas K, Ledig C, Newcombe VF, Simpson JP, Kane AD, Menon DK, et al. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. Med Image Anal 2017;36:61–78.

[24] Havaei M, Davy A, Warde-Farley D, Biard A, Courville A, Bengio Y, et al. Brain tumor segmentation with deep neural networks. Med Image Anal 2017;35:18–31.

[25] Moeskops P, Viergever MA, Mendrik AM, de Vries LS, Benders MJ, Išgum I. Automatic segmentation of MR brain images with a convolutional neural network. IEEE Trans Med Imaging 2016;35(5):1252–61.

[26] Zhang W, Li R, Deng H, Wang L, Lin W, Ji S, et al. Deep convolutional neural networks for multi-modality isointense infant brain image segmentation. Neuroimage 2015;108:214–24.

[27] Chen H, Dou Q, Yu L, Qin J, Heng P-A. VoxResNet: deep voxelwise residual networks for brain segmentation from 3D MR images. NeuroImage 2017;170:446–55.

[28] Dolz J, Desrosiers C, Ayed IB. 3D fully convolutional networks for subcortical segmentation in MRI: a large-scale study. Neuroimage 2018;170:456–70.

[29] Wachinger C, Reuter M, Klein T. DeepNAT: deep convolutional neural network for segmenting neuroanatomy. Neuroimage 2018;170:434–45.

[30] Lladó X, Ganiler O, Oliver A, Martí R, Freixenet J, Valls L, et al. Automated detection of multiple sclerosis lesions in serial brain MRI. Neuroradiology 2012;54(8):787–807.

[31] Lladó X, Oliver A, Cabezas M, Freixenet J, Vilanova JC, Quiles A, et al. Segmentation of multiple sclerosis lesions in brain MRI: a review of automated approaches. Inf Sci 2012;186(1):164–85.

[32] González-Villà S, Oliver A, Valverde S, Wang L, Zwiggelaar R, Lladó X. A review on brain structures segmentation in magnetic resonance imaging. Artif Intell Med 2016;73:45–69.

[33] Mitra J, Bourgeat P, Fripp J, Ghose S, Rose S, Salvado O, et al. Lesion segmentation from multimodal MRI using random forest following ischemic stroke. Neuroimage 2014;98:324–35.

[34] Roura E, Oliver A, Cabezas M, Valverde S, Pareto D, Vilanova JC, et al. A toolbox for multiple sclerosis lesion segmentation. Neuroradiology 2015;57(10):1031–43.

[35] Shen K-k, Fripp J, Mériaudeau F, Chételat G, Salvado O, Bourgeat P, et al. Detecting global and local hippocampal shape changes in Alzheimer's disease using statistical shape models. Neuroimage 2012;59(3):2155–66.

[36] Bengio Y. Deep learning of representations: Looking forward. International conference on statistical language and speech processing. 2013. p. 1–37.

[37] Deng L, Yu D. Deep Learning. Signal Process 2014;7:3–4.

[38] Schmidhuber J. Deep learning in neural networks: an overview. Neural Netw 2015;61:85–117.

[39] Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 2013;35(8):1798–828.

[40] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521(7553):436–44.

[41] Deng L. Deep learning: from speech recognition to language and multimodal processing. APSIPA Trans Signal Inf Process 2016;5:e1.

[42] Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, Lew MS. Deep learning for visual understanding: a review. Neurocomputing 2016;187:27–48.

[43] Angermueller C, Pärnamaa T, Parts L, Stegle O. Deep learning for computational biology. Mol Syst Biol 2016;12(7):878.

[44] Mamoshina P, Vieira A, Putin E, Zhavoronkov A. Applications of deep learning in biomedicine. Mol Pharm 2016;13(5):1445–54.

[45] Shen D, Wu G, Suk H-I. Deep learning in medical image analysis. Annu Rev Biomed Eng 2017;19:221–48.

[46] Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, et al. A survey on deep learning in medical image analysis. Med Image Anal 2017;42:60–88.

[47] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE 1998;86(11):2278–324.

[48] LeCun Y, Boser BE, Denker JS, Henderson D, Howard RE, Hubbard WE, et al. Handwritten digit recognition with a back-propagation network. Advances in neural information processing systems 1990:396–404.

[49] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Net 1994;5(2):157–66.

[50] Hinton GE, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. Neural Comput 2006;18(7):1527–54.

[51] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. Aistats, vol. 15 2011:275.

[52] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition 2015:1–9.

[53] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. Proceedings of the IEEE conference on computer vision and pattern recognition 2016:2818–26.

[54] Nair V, Hinton GE. Rectified linear units improve restricted Boltzmann machines.

Proceedings of the 27th international conference on machine learning (ICML-10) 2010:807–14.

[55] He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE international conference on computer vision 2015:1026–34.

[56] Maas AL, Hannun AY, Ng AY. Rectifier nonlinearities improve neural network acoustic models. Proc international conference on machine learning, vol. 30 2013.

[57] Xu B, Wang N, Chen T, Li M. Empirical evaluation of rectified activations in convolutional network. CoRR abs/1505.00853. http://arxiv.org/abs/1505.00853.

[58] Jin X, Xu C, Feng J, Wei Y, Xiong J, Yan S. Deep learning with S-shaped rectified linear activation units. Thirtieth AAAI conference on artificial intelligence 2016.

[59] LeCun Y, Cortes C, Burges CJ. The MNIST database of handwritten digits. 1998.

[60] Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. Imagenet: a large-scale hierarchical image database. IEEE conference on computer vision and pattern recognition, 2009. CVPR 2009. 2009. p. 248–55.

[61] Goodfellow IJ, Warde-Farley D, Mirza M, Courville AC, Bengio Y. Maxout networks. ICML (3) 2013;28:1319–27.

[62] Springenberg JT, Riedmiller MA. Improving deep neural networks with probabilistic maxout units. CoRR abs/1312.6116. http://arxiv.org/abs/1312.6116.

[63] Chen H, Yu L, Dou Q, Shi L, Mok VC, Heng PA. Automatic detection of cerebral microbleeds via deep learning based 3D feature representation. 2015 IEEE 12th international symposium on biomedical imaging (ISBI), IEEE 2015:764–7.

[64] Trottier L, Giguére P, Chaib-draa B. Parametric exponential linear unit for deep convolutional neural networks. CoRR abs/1605.09332. http://arxiv.org/abs/1605.09332.

[65] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016.

[66] Boureau Y-L, Bach F, LeCun Y, Ponce J. Learning mid-level features for recognition. 2010 IEEE conference on computer vision and pattern recognition (CVPR). 2010. p. 2559–66.

[67] Boureau Y-L, Ponce J, LeCun Y. A theoretical analysis of feature pooling in visual recognition. Proceedings of the 27th international conference on machine learning (ICML-10) 2010:111–8.

[68] Zeiler MD, Fergus R. Stochastic pooling for regularization of deep convolutional neural networks. CoRR abs/1301.3557. http://arxiv.org/abs/1301.3557.

[69] Netzer Y, Wang T, Coates A, Bissacco A, Wu B, Ng AY. Reading digits in natural images with unsupervised feature learning. NIPS workshop on deep learning and unsupervised feature learning, vol. 2011 2011:5.

[70] He K, Zhang X, Ren S, Sun J. Spatial pyramid pooling in deep convolutional networks for visual recognition. European conference on computer vision. 2014. p. 346–61.

[71] Ouyang W, Wang X, Zeng X, Qiu S, Luo P, Tian Y, et al. Deepid-net: Deformable deep convolutional neural networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition 2015:2403–12.

[72] Lee C-Y, Gallagher PW, Tu Z. Generalizing pooling functions in convolutional neural networks: mixed, gated, and tree. International conference on artificial intelligence and statistics 2016.

[73] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition 2014:580–7.

[74] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition 2015:3431–40.

[75] Tang Y. Deep learning using support vector machines. CoRR abs/1306.0239. http://arxiv.org/abs/1306.0239.

[76] Liu W, CN E, Wen Y, SCUT M, Yu Z, Yang M. Large-margin softmax loss for convolutional neural networks. Proceedings of the 33rd international conference on machine learning 2016:507–16.

[77] Cruz R, Fernandes K, Cardoso JS, Costa JFP. Tackling class imbalance with ranking. 2016 International joint conference on neural networks (IJCNN), IEEE 2016:2182–7.

[78] Brosch T, Yoo Y, Tang LY, Li DK, Traboulsee A, Tam R. Deep convolutional encoder networks for multiple sclerosis lesion segmentation. International conference on medical image computing and computer-assisted intervention. 2015. p. 3–11.

[79] Milletari F, Navab N, Ahmadi S-A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. 2016 Fourth international conference on 3D Vision (3DV), IEEE 2016:565–71.

[80] Sun Y, Chen Y, Wang X, Tang X. Deep learning face representation by joint identification-verification. Advances in neural information processing systems 2014:1988–96.

[81] Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering. Proceedings of the IEEE conference on computer vision and pattern recognition 2015:815–23.

[82] Wager S, Wang S, Liang PS. Dropout training as adaptive regularization. Advances in neural information processing systems 2013:351–9.

[83] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15:1929–58.

[84] Wang SI, Manning CD. Fast dropout training. ICML (2) 2013:118–26.

[85] Wan L, Zeiler M, Zhang S, Cun YL, Fergus R. Regularization of neural networks using dropconnect. Proceedings of the 30th international conference on machine learning (ICML-13) 2013:1058–66.

[86] Howard AG. Some improvements on deep convolutional neural network based image classification. CoRR abs/1312.5402. http://arxiv.org/abs/1312.5402.

[87] Chatfield K, Simonyan K, Vedaldi A, Zisserman A. Return of the devil in the details: delving deep into convolutional nets. CoRR abs/1405.3531. http://arxiv.org/abs/1405.3531.

[88] LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, et al. Backpropagation applied to handwritten zip code recognition. Neural Comput 1989;1(4):541–51.

[89] Sutton RS. Two problems with backpropagation and other steepest-descent learning procedures for networks. Proc. 8th annual conf. Cognitive Science Society 1986:823–31.

[90] Dauphin YN, Pascanu R, Gulcehre C, Cho K, Ganguli S, Bengio Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. Advances in neural information processing systems 2014:2933–41.

[91] Nesterov Y. A method for unconstrained convex minimization problem with the rate of convergence O (1/k2). Doklady an SSSR, vol. 269 1983:543–7.

[92] Sutskever I, Martens J, Dahl GE, Hinton GE. On the importance of initialization and momentum in deep learning. ICML (3) 2013;28:1139–47.

[93] Ge R, Huang F, Jin C, Yuan Y. Escaping from saddle points–online stochastic gradient for tensor decomposition. Proceedings of the 28th conference on learning theory 2015:797–842.

[94] Anandkumar A, Ge R. Efficient approaches for escaping higher order saddle points in non-convex optimization. Conference on learning theory 2016:81–102.

[95] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res 2011;12(Jul):2121–59.

[96] Zeiler MD. ADADELTA: an adaptive learning rate method. CoRR abs/1212.5701. http://arxiv.org/abs/1212.5701.

[97] Tieleman T, Hinton G. Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA 2012;4(2).

[98] Kingma DP, Ba J. Adam: a method for stochastic optimization. CoRR abs/1412.6980. http://arxiv.org/abs/1412.6980.

[99] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. Aistats, vol. 9 2010:249–56.

[100] Lin M, Chen Q, Yan S. Network in network. CoRR abs/1312.4400. http://arxiv.org/abs/1312.4400.

[101] Salimans T, Kingma DP. Weight normalization: a simple reparameterization to accelerate training of deep neural networks. Advances in neural information processing systems 2016:901.

[102] Acosta-Cabronero J, Williams GB, Pereira JMS, Pengas G, Nestor PJ. The impact of skull-stripping and radio-frequency bias correction on grey-matter segmentation for voxel-based morphometry. Neuroimage 2008;39(4):1654–65.

[103] Popescu V, Battaglini M, Hoogstrate WS, Verfaillie SCJ, Sluimer IC, van Schijndel RA, et al. Optimizing parameter choice for FSL-Brain Extraction Tool (BET) on 3D T1 images in multiple sclerosis. Neuroimage 2012;61(4):1484–94.

[104] Sadananthan SA, Zheng W, Chee MWL, Zagorodnov V. Skull stripping using graph cuts. Neuroimage 2010;49(1):225–39.

[105] Lee J-M, Yoon U, Nam SH, Kim J-H, Kim I-Y, Kim SI. Evaluation of automated and semi-automated skull-stripping algorithms using similarity index and segmentation error. Comput Biol Med 2003;33(6):495–507.

[106] Roura E, Oliver A, Cabezas M, Vilanova JC, Rovira À, Ramió-Torrentà L, et al. MARGA: multispectral adaptive region growing algorithm for brain extraction on axial MRI. Comput Methods Programs Biomed 2014;113(2):655–73.

[107] Smith SM. Fast robust automated brain extraction. Hum Brain Mapp 2002;17(3):143–55.

[108] Jenkinson M, Pechaud M, Smith S. BET2: MR-based estimation of brain, skull and scalp surfaces. Eleventh annual meeting of the organization for human brain mapping 2005.

[109] Shattuck DW, Sandor-Leahy SR, Schaper KA, Rottenberg DA, Leahy RM. Magnetic resonance image tissue classification using a partial volume model. Neuroimage 2001;13(5):856–76.

[110] Iglesias JE, Liu C-Y, Thompson PM, Tu Z. Robust brain extraction across datasets and comparison with publicly available methods. IEEE Trans Med Imaging 2011;30(9):1617–34.

[111] Eskildsen SF, Coupé P, Fonov V, Manjón JV, Leung KK, Guizard N, et al. BEaST: brain extraction based on nonlocal segmentation technique. Neuroimage 2012;59(3):2362–73.

[112] Yoo Y, Brosch T, Traboulsee A, Li DK, Tam R. Deep learning of image features from unlabeled data for multiple sclerosis lesion segmentation. International workshop on machine learning in medical imaging. 2014. p. 117–24.

[113] Shi F, Wang L, Dai Y, Gilmore JH, Lin W, Shen D. LABEL: pediatric brain extraction using learning-based meta-algorithm. Neuroimage 2012;62(3):1975–86.

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

[115] Simmons A, Tofts PS, Barker GJ, Arridge SR. Sources of intensity nonuniformity in spin echo images at 1.5 T. Magn Reson Med 1994;32(1):121–8.

[116] Juntu J, Sijbers J, Van Dyck D, Gielen J. Bias field correction for MRI images. Computer recognition systems. 2005. p. 543–51.

[117] Arnold JB, Liow J-S, Schaper KA, Stern JJ, Sled JG, Shattuck DW, et al. Qualitative and quantitative evaluation of six algorithms for correcting intensity nonuniformity effects. Neuroimage 2001;13(5):931–43.

[118] Ge Y, Grossman RI, Babb JS, Rabin ML, Mannon LJ, Kolson DL. Age-related total gray matter and white matter changes in normal adult brain. Part I: volumetric M imaging analysis. Am J Neuroradiol 2002;23(8):1327–33.

[119] Sled JG, Zijdenbos AP, Evans AC. A nonparametric method for automatic correction of intensity nonuniformity in MRI data. IEEE Trans Med Imaging 1998;17(1):87–97.

[120] Tustison NJ, Avants BB, Cook PA, Zheng Y, Egan A, Yushkevich PA, et al. N4ITK: improved N3 bias correction. IEEE Trans Med Imaging 2010;29(6):1310–20.

[121] Lyksborg M, Puonti O, Agn M, Larsen R. An ensemble of 2D convolutional neural

networks for tumor segmentation. Scandinavian conference on image analysis. 2015. p. 201–11.

[122] Zikic D, Ioannou Y, Brown M, Criminisi A. Segmentation of brain tumor tissues with convolutional neural networks. Proceedings MICCAI-BRATS 2014:36–9.

[123] Birenbaum A, Greenspan H. Longitudinal multiple sclerosis lesion segmentation using multi-view convolutional neural networks. International workshop on large-scale annotation of biomedical data and expert label synthesis. 2016. p. 58–67.

[124] Urban G, Bendszus M, Hamprecht F, Kleesiek J. Multi-modal brain tumor segmentation using deep convolutional neural networks. MICCAI BraTS (Brain Tumor Segmentation) challenge. Proceedings, winning contribution 2014:31–5.

[125] Vaidhya K, Thirunavukkarasu S, Alex V, Krishnamurthy G. Multi-modal brain tumor segmentation using stacked denoising autoencoders. International workshop on brainlesion: glioma, multiple sclerosis, stroke and traumatic brain injuries. 2015. p. 181–94.

[126] Valverde S, Cabezas M, Roura E, González-Villà S, Pareto D, Vilanova JC, et al. Improving automated multiple sclerosis lesion segmentation with a cascaded 3D convolutional neural network approach. Neuroimage 2017;155:159–68.

[127] Nyúl LG, Udupa JK, Zhang X. New variants of a method of MRI scale standardization. IEEE Trans Med Imaging 2000;19(2):143–50.

[128] Pizer SM, Amburn EP, Austin JD, Cromartie R, Geselowitz A, Greer T, et al. Adaptive histogram equalization and its variations. Comput Vis Graph Image Process 1987;39(3):355–68.

[129] Stollenga MF, Byeon W, Liwicki M, Schmidhuber J. Parallel multi-dimensional LSTM, with application to fast biomedical volumetric image segmentation. Proceedings of the 28th international conference on neural information processing systems, NIPS'15. 2015. p. 2998–3006.

[130] Dvorak P, Menze B. Structured prediction with convolutional neural networks for multimodal brain tumor segmentation. Proceeding of the multimodal brain tumor image segmentation challenge 2015:13–24.

[131] Bengio Y. Practical recommendations for gradient-based training of deep architectures. Neural networks: tricks of the trade. 2012. p. 437–78.

[132] Lai M. Deep learning for medical image segmentation. CoRR abs/1505.02000. http://arxiv.org/abs/1505.02000.

[133] Milletari F, Ahmadi S-A, Kroll C, Plate A, Rozanski V, Maiostre J, et al. Hough-CNN: deep learning for segmentation of deep brain regions in MRI and ultrasound. Comput Vis Image Underst 2017. https://doi.org/10.1016/j.cviu.2017.04.002.

[134] Li W, Wang G, Fidon L, Ourselin S, Cardoso MJ, Vercauteren T. On the compactness, efficiency, and representation of 3D convolutional networks: brain parcellation as a pretext task. International conference on information processing in medical imaging. 2017. p. 348–60.

[135] de Brebisson A, Montana G. Deep neural networks for anatomical brain segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition workshops 2015:20–8.

[136] Zhao L, Jia K. Multiscale CNNs for brain tumor segmentation and diagnosis. Computational and mathematical methods in medicine 2016.

[137] Likar B, Viergever MA, Pernus F. Retrospective correction of MR intensity inhomogeneity by information minimization. IEEE Trans Med Imaging 2001;20(12):1398–410.

[138] Shakeri M, Tsogkas S, Ferrante E, Lippe S, Kadoury S, Paragios N, et al. Sub-cortical brain structure segmentation using F-CNN's. 2016 IEEE 13th international symposium on biomedical imaging (ISBI). 2016. p. 269–72.

[139] Mehta R, Majumdar A, Sivaswamy J. BrainSegNet: a convolutional neural network architecture for automated segmentation of human brain structures. J Med Imaging 2017;4(2). 024003.

[140] Krähenbühl P, Koltun V. Efficient inference in fully connected CRFs with Gaussian edge potentials. Advances in neural information processing systems 2011:109–17.

[141] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556. http://arxiv.org/abs/1409.1556.

[142] Hariharan B, Arbeláez P, Girshick R, Malik J. Hypercolumns for object segmentation and fine-grained localization. Proceedings of the IEEE conference on computer vision and pattern recognition 2015:447–56.

[143] Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. International conference on medical image computing and computer-assisted intervention. 2015. p. 234–41.

[144] Arganda-Carreras I, Turaga SC, Berger DR, Cireşan D, Giusti A, Gambardella LM, et al. Crowdsourcing the creation of image segmentation algorithms for connectomics. Front Neuroanat. 2015;9:142. https://doi.org/10.3389/fnana.2015.

00142.

[145] Yu L, Yang X, Chen H, Qin J, Heng P. Volumetric convnets with mixed residual connections for automated prostate segmentation from 3d MR images. Proceedings of the thirty-first AAAI conference on artificial intelligence. 2017. p. 66–72.

[146] Çiçek Ö, Abdulkadir A, Lienkamp SS, Brox T, Ronneberger O. 3D u-net: learning dense volumetric segmentation from sparse annotation. International conference on medical image computing and computer-assisted intervention. 2016. p. 424–32.

[147] Christ PF, Elshaer MEA, Ettlinger F, Tatavarty S, Bickel M, Bilic P, et al. Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random fields. International conference on medical image computing and computer-assisted intervention. 2016. p. 415–23.

[148] Lombaert H, Sporring J, Siddiqi K. Diffeomorphic spectral matching of cortical surfaces. IPMI, vol. 7917 2013:376–89.

[149] Wachinger C, Brennan M, Sharp G, Golland P. On the importance of location and features for the patch-based segmentation of parotid glands. MICCAI workshop on image-guided adaptive radiation therapy 2014.

[150] Wachinger C, Brennan M, Sharp G, Golland P. Efficient descriptor-based segmentation of parotid glands with non-local means. IEEE Trans Biomed Eng 2016;PP(99):1.

[151] Vezhnevets V, Konouchine V. GrowCut: interactive multi-label ND image segmentation by cellular automata. Proc of Graphicon, vol. 1. 2005. p. 150–6.

[152] Styner M, Lee J, Chin B, Chin M, Commowick O, Tran H, et al. 3D segmentation in the clinic: a grand challenge II: MS lesion segmentation. Midas J 2008;2008:1–6.

[153] Išgum I, Benders MJ, Avants B, Cardoso MJ, Counsell SJ, Gomez EF, et al. Evaluation of automatic neonatal brain segmentation algorithms: the NeoBrainS12 challenge. Med Image Anal 2015;20(1):135–51.

[154] Mendrik AM, Vincken KL, Kuijf HJ, Breeuwer M, Bouvy WH, De Bresser J, et al. MRBrainS challenge: online evaluation framework for brain image segmentation in 3T MRI scans. Comput Intell Neurosci 2015;2015:1.

[155] Menze BH, Jakab A, Bauer S, Kalpathy-Cramer J, Farahani K, Kirby J, et al. The multimodal brain tumor image segmentation benchmark (BRATS). IEEE Trans Med Imaging 2015;34(10):1993–2024.

[156] Carass A, Roy S, Jog A, Cuzzocreo JL, Magrath E, Gherman A, et al. Longitudinal multiple sclerosis lesion segmentation: resource and challenge. Neuroimage 2017;148:77–102.

[157] Maier O, Menze BH, von der Gablentz J, Häni L, Heinrich MP, Liebrand M, et al. ISLES 2015-A public evaluation benchmark for ischemic stroke lesion segmentation from multispectral MRI. Med Image Anal 2017;35:250–69.

[158] Commowick O, Cervenansky F, Ameli R. MSSEG challenge proceedings: multiple sclerosis lesions segmentation challenge using a data management and processing infrastructure. 2016.

[159] Sudre CH, Li W, Vercauteren T, Ourselin S, Cardoso MJ. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. Deep learning in medical image analysis and multimodal learning for clinical decision support. 2017. p. 240–8.

[160] Fidon L, Li W, Garcia-Peraza-Herrera LC, Ekanayake J, Kitchen N, Ourselin S, et al. Generalised Wasserstein dice score for imbalanced multi-class segmentation using holistic convolutional networks. In: Crimi A, Bakas S, Kuijf H, Menze B, Reyes M, editors. Brainlesion: glioma, multiple sclerosis, stroke and traumatic brain injuries. 2018. p. 64–76.

[161] Zhang Y, Brady M, Smith S. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. IEEE Trans Med Imaging 2001;20(1):45–57.

[162] Ashburner J, Friston KJ. Unified segmentation. Neuroimage 2005;26(3):839–51.

[163] Ashburner J, Barnes G, Chen C. SPM8 manual. 2012 [accessed 18.05.17]. http://www.fil.ion.ucl.ac.uk/.

[164] Valverde S, Oliver A, Roura E, González-Villà S, Pareto D, Vilanova JC, et al. Automated tissue segmentation of MR brain images in the presence of white matter lesions. Med Image Anal 2017;35:446–57.

[165] Lavin A, Gray S. Fast algorithms for convolutional neural networks. Proceedings of the IEEE conference on computer vision and pattern recognition 2016:4013–21.

[166] Paine T, Jin H, Yang J, Lin Z, Huang TS. GPU asynchronous stochastic gradient descent to speed up neural network training. CoRR abs/1312.6186. http://arxiv.org/abs/1312.6186.