15

*Information Fusion in Deep Convolutional Neural Networks for Biomedical Image Segmentation**

Mohammad Havaei, Nicolas Guizard, Nicolas Chapados, and Yoshua Bengio

CONTENTS

15.1 Introduction

Among clinical datasets, medical images are perhaps the largest and most complex to analyze. Typical individual computed tomography (CT) scans and magnetic resonance images (MRIs) range from a few tens of megabytes to multi-gigabytes per study. Moreover, with the greater availability of equipment, the number of scans performed per year has been steadily increasing, both in the United States and worldwide [1]. Figure 15.1 illustrates the increase in per-capita CT and MRI scans in the United States over the last 20 years. In addition, new screening guidelines coming into force for cancers such as lung, in countries such as the United States [2] and Canada (Canadian Task Force on Preventive Health Care, 2016), imply that a great increase in scans is to be further expected, leading to a bottleneck in the availability of radiologists to analyze all these data. Hence, tools to automate some of this analysis are beneficial. This chapter proposes a technique based on deep convolutional neural networks (CNNs) to carry out important analytical tasks—such as image segmentation—given a varying set of available imaging modalities.

In medical image analysis, image segmentation is a common subtask to achieve a number of clinical goals, such as lesion detection and characterization, and is primordial to visualize and quantify the severity of the pathology in clinical practice. Multi-modality imaging provides complementary information to discriminate specific tissues, anatomies, and pathologies. However, manual segmentation is long, painstaking, and subject to human variability. In the last decades, numerous automatic approaches have been developed to speed up medical image segmentation. These methods can be grouped into two categories: The first, *multi-atlas approaches* estimate online intensity similarities between the subject being segmented and multi-atlases (images with expert labels). These techniques have shown excellent results in structural segmentation when using non-linear registration [4] when combined with non-local approaches, they have proven effective in segmenting diffuse and sparse pathologies (i.e., multiple sclerosis [MS] lesions [6] as well as more complex multi-

* Portions of this work were previously published in Havaei et al. [5].

9781498773454_C015.3d 301 06/04/2018 08:29:33am

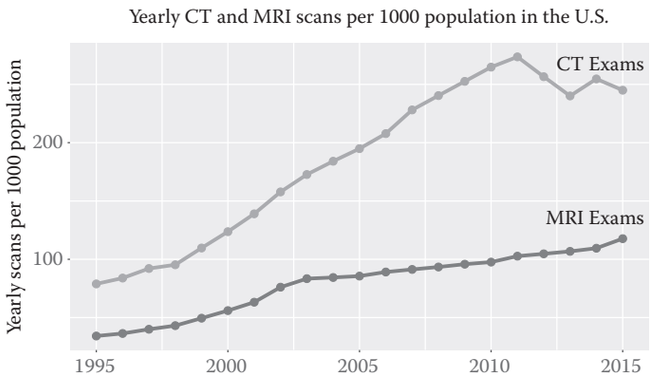

FIGURE 15.1
Yearly CT and MRI scans per 1000 population in the United States. These numbers have been steadily increasing for the past 20 years with the greater availability of equipment, making automated analysis tools ever more desirable.

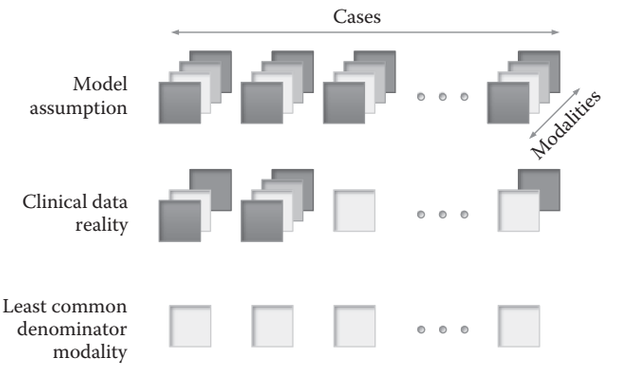

FIGURE 15.2
Most multi-modal computer vision and machine learning models assume that all modalities are available for all patients (cases), illustrated on top; however, clinical datasets available in practice will generally omit one or more modalities from most cases (middle), often reducing model scope to the least common denominator modality (bottom).

label pathology, e.g., in glioblastoma [7]. Multi-atlas methods rely on image intensity and spatial similarity, which can be difficult to be fully described by the atlases and heavily dependent on the image pre-processing. In contrast, *model-based approaches* are typically trained offline to identify a discriminative model of image intensity features. These features can be predefined by the user (e.g., with random forests; [8] or extracted and learned hierarchically directly from the images [9].

Both strategies are typically optimized for a specific set of multi-modal images and usually require these modalities to be available. In clinical settings, image acquisition and patient artifacts, among other hurdles, make it difficult to fully exploit all the modalities; as such, it is common to have one or more modalities to be missing for a given instance, a problem illustrated in Figure 15.2. This problem is not new, and the subject of missing data analysis has spawned an immense literature in statistics (see, e.g., Van Buuren, [10]). In medical imaging, a number of approaches have been proposed, some of which require to re-train a specific model with the missing modalities or to synthesize them [11]. Synthesis can improve multi-modal classification by adding information of the missing modalities in the context of a simple classifier such as random forests [12]. Approaches to imitate with fewer features a classifier trained with a complete set of features have also been proposed [13]. Nevertheless, it should stand to reason that a more complex model should be capable of extracting relevant features from just the available modalities without relying on artificial intermediate steps such as imputation or synthesis.

This chapter introduces a deep learning framework (called *Heteromodal Image Segmentation*, or HeMIS) that can segment medical images from incomplete multi-modal datasets. Deep learning [14] has shown an increasing popularity in medical image processing for segmenting but also to synthesize missing modalities [15]. Here, the proposed approach learns, separately for each modality, an embedding of the input image into a latent space. In this space, arithmetic operations (such as computing first and second moments of a collection of vectors) are well defined and can be taken over the different modalities available at inference time. These computed moments can then be further processed to estimate the final segmentation; an overview of the approach is shown in Figure 15.3. This approach presents the advantage of being robust to any combinatorial subset of available modalities provided as input, without the need to learn a combinatorial number of imputation models.

After presenting a background refresher on deep CNNs (Section 15.2), we describe the method (Section 15.3), follow with a description of the datasets (Section 15.4)

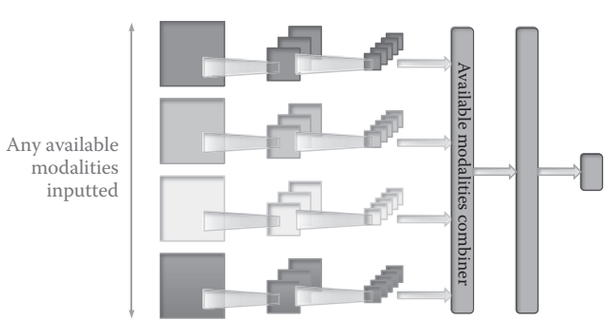

FIGURE 15.3
High-level view of the HeMIS deep CNN structure, wherein a separate projection to a common embedding is learned for each possible modality, which are then combined and further processed before a final output is produced. The following sections detail this model.

and experiments (Section 15.5), and finally offer concluding remarks (Section 15.6).

15.2 Background: Deep CNNs

CNNs are a type of neural network adopted for spatially ordered input. The main building block used to construct a CNN architecture is the *convolutional layer*. As in a regular neural network, several convolutional layers can be stacked on top of each other forming a hierarchy of features. Each layer can be understood as extracting features from its preceding layer in the hierarchy. A single convolutional layer takes as input a stack of input planes and produces as output some number of output planes or *feature maps*. Each feature map can be thought of as a topologically arranged map of responses of a particular spatially local non-linear feature extractor (the parameters of which are learned), applied identically to each spatial neighborhood of the input planes in a sliding window fashion. In the case of a first convolutional layer, the individual input planes correspond to different input channels. In the case of MRI, it can be single slices of different MR imaging modalities,* and in the case of color images, it can be different color channels. In subsequent layers, the input planes typically consist of the feature maps of the previous layer. Computing a feature map in a convolutional layer (see Figure 15.4) consists of the following three steps:

1. *Convolution of kernels (filters):* Each feature map $\mathbf{O}_s$ of the layer is associated with one kernel (or several, in the case of Maxout†). The feature map $\mathbf{O}_s$ is computed as follows:

$$\mathbf{O}_s = b_s + \sum_r \mathbf{W}_{sr} * \mathbf{X}_r \qquad (15.1)$$

where $\mathbf{X}_r$ is the rth input channel, $\mathbf{W}_{sr}$ is the subkernel for that channel, * is the convolution operation, and b_s is a bias term.‡ In other words, the affine operation being performed for each feature map is the *sum* of the application of R different two-dimensional $N \times N$ convolution filters (one per input channel/modality), plus a bias term that is added pixel-wise to each resulting spatial position. The convolutional operation of image $\mathbf{X}$ and kernel $\mathbf{W}$ is computed as

$$\mathbf{C}_{ij} = (\mathbf{W} * \mathbf{X})_{ij} = \sum_m \sum_n \mathbf{X}_{i+m,j+n} \mathbf{W}_{-m,-n}. \qquad (15.2)$$

In the above equation, the region in matrix $\mathbf{X}$ that is used in computation of $\mathbf{C}_{ij}$ is referred to as the *local receptive field* for $\mathbf{C}_{ij}$, and so $\mathbf{C}_{ij}$ is only connected to its receptive field, rather than the whole image as it is the case with standard multi-layer perceptrons (MLPs). This greatly reduces the number of parameters of the model. This receptive field is slid across the entire image, a process illustrated in Figure 15.5 [15]. For each receptive field, there is a different hidden neuron (i.e., $\mathbf{C}_{s,ij}$). However, the weights to compute every hidden neuron are shared. This further reduces the parameters of the model by a factor of the number of neurons in that feature map. Intuitively, the reason for sharing parameters is that each kernel can be thought of as a feature detector that tries to identify that particular feature at different spatial positions in the image. Also, by sharing parameters, we can greatly reduce the parameters of the model and reduce risk of overfitting. The combination of several of those is illustrated in Figure 15.6 [15].

Whereas traditional image feature extraction methods rely on a fixed recipe (sometimes taking the form of convolutions with a linear filter bank), the key to the success of CNNs is their ability to learn the weights and biases of individual feature maps, giving rise to data-driven, customized, task-specific dense feature extractors. These parameters are learned via stochastic gradient descent on a surrogate loss function, with gradients computed efficiently via the backpropagation algorithm.

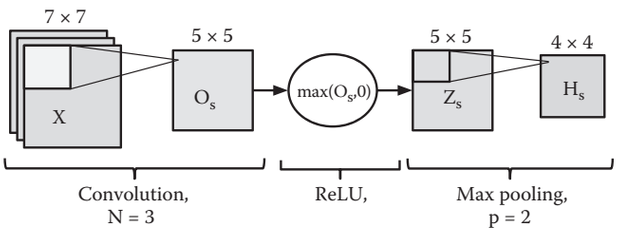

FIGURE 15.4
Single convolution layer block showing computations for a single feature map. The input patch (here 7 × 7) is convolved with a series of kernels (here 3 × 3) followed by ReLU and max-pooling.

* In this work, we assume 2D convolutional networks, although the formulation generalize straightforwardly to 3D and higher dimensionalities.

† Maxout will be discussed later in this chapter.

‡ Since the convolutional layer is associated to R input channels, $\mathbf{X}$ contains $M \times M \times R$ gray-scale values and thus each kernel $\mathbf{W}_s$ contains $N \times N \times R$ weights. Accordingly, the number of parameters in a convolutional block, consisting of S feature maps, is equal to $R \times M \times M \times S$.

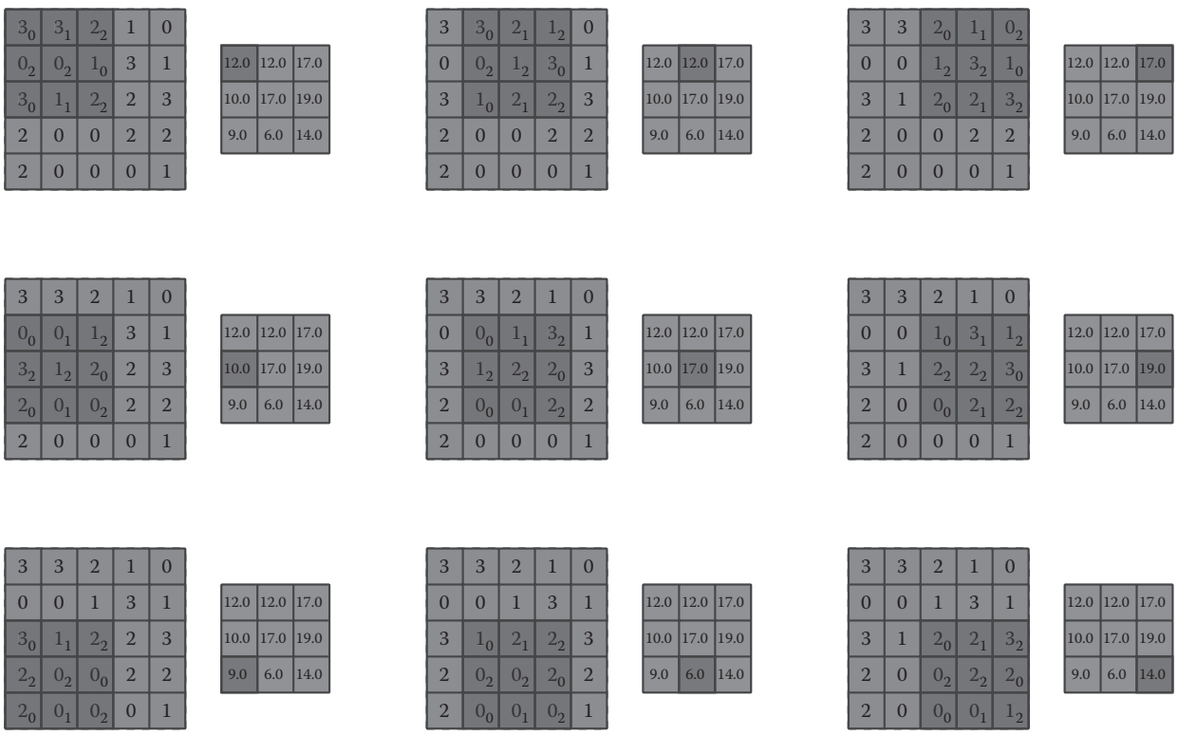

FIGURE 15.5
Computing the output values of a discrete convolution. (From Dumoulin, V. and F. Visin (2016, March). A guide to convolution arithmetic for deep learning. *ArXiv e-prints*. With permission.)

Special attention must be paid to the treatment of border pixels by the convolution operation. One option is to employ the so-called *valid* mode convolution, meaning that the filter response is not computed for pixel positions that are less than $\lfloor N / 2 \rfloor$ pixels away from the image border. An $M \times M$ input convolved with an $N \times N$ filter patch will result in a $Q \times Q$ output, where $Q = M - N + 1$. In Figure 15.4, $M = 7$, $N = 3$, and thus $Q = 5$. Note that the size (spatial width and height) of the kernels are hyper-parameters that must be specified by the user. One can apply the convolutions in the *same* mode to preserve the input size. In this mode, zero padding is applied around the input prior to the convolution operation.

2. *Non-linear activation function:* To obtain features that are non-linear transformations of the input, an elementwise non-linearity is applied to the result of the kernel convolution. There are multiple choices for this non-linearity, such as the sigmoid, hyperbolic tangent, and rectified linear functions [16,17] or maxout [18]. The rectified linear function, or ReLU, is defined elementwise simply as

$$\mathrm{ReLU}(\cdot) = \max(0, \cdot),$$

whereas maxout features are associated with multiple kernels $\mathbf{W}_s$, wherein maxout map $\mathbf{Z}_s$ is associated with K feature maps: $\{\mathbf{O}_{Ks}, O_{Ks+1}, \ldots, \mathbf{O}_{Ks+K-1}\}$. Maxout features correspond to taking the max over the feature maps $\mathbf{O}$, individually for each spatial position,

$$Z_{s,i,j} = \max\left\{O_{Ks,i,j}, O_{Ks+1,i,j}, \ldots, O_{Ks+K-1,i,j}\right\}, \quad (15.3)$$

where i, j are spatial positions. Maxout features are thus equivalent to using a convex activation function, but whose shape is adaptive and depends on the values taken by the kernels. The ReLU function can be considered a special form of Maxout where the max operation is taken over every feature map and a zero matrix of the same size for each spatial position (i.e. $\max(\mathbf{O}_s, \mathbf{0})$),

$$Z_{s,i,j} = \max\left\{O_{s,i,j}, 0_{i,j}\right\}. \quad (15.4)$$

Note that in Figure 15.4, the ReLU activation function is used.

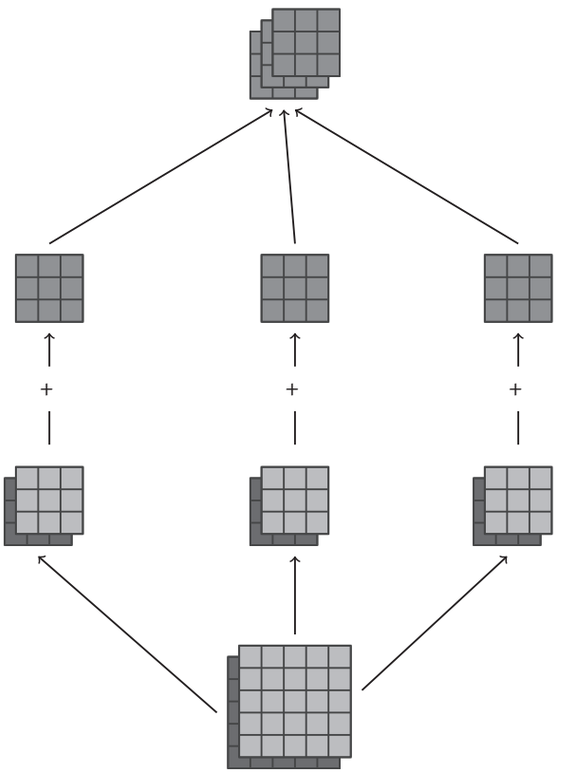

FIGURE 15.6
Convolution mapping from two input feature maps to three output feature maps using a $3 \times 2 \times 3 \times 3$ collection of kernels $\mathbf{w}$. In the left pathway, input feature map 1 is convolved with kernel $\mathbf{w}_{1,1}$ and input feature map 2 is convolved with kernel $\mathbf{w}_{1,2}$, and the results are summed together elementwise to form the first output feature map. The same is repeated for the middle and right pathways to form the second and third feature maps, and all three output feature maps are grouped together to form the output. (From Dumoulin, V. and F. Visin (2016, March). A guide to convolution arithmetic for deep learning. *ArXiv e-prints*. With permission.)

3. *Max-pooling:* This operation consists of taking the maximum feature (neuron) value over sub-windows within each feature map. This can be formalized as follows:

$$H_{s,i,j} = \max_{p} Z_{s,Si+p,Sj+p}, \tag{15.5}$$

where p determines the max-pooling window size, and S is the stride value that corresponds to the horizontal and vertical increments at which pooling sub-windows are positioned. Depending on the stride value, the sub-windows can be overlapping or not (Figure 15.4 shows an overlapping configuration). The max-pooling operation shrinks the size of the feature map. This is controlled by the pooling size p and the stride hyper-parameter. Let $Q \times Q$ be the shape of the feature map before max-pooling. The output of the max-pooling operation would be of size $D \times D$, where $D = (Q - p) / S + 1$.* In Figure 15.4, since $Q = 5$, $p = 2$, $S = 1$, the max-pooling operation results into a $D = 4$ output feature map. The motivation for this operation is to introduce invariance to local translations. This subsampling procedure has been found beneficial in other applications [19].

15.3 Method

15.3.1 Hetero-Modal Image Segmentation

Typical CNN architectures take a multiplane image as input and process it through a sequence of convolutional layers (followed by nonlinearities such as ReLU), alternating with optional pooling layers, to yield a per-pixel or per-image output [14]. In such networks, every input plane is assumed to be present within a given instance: since the very first convolutional layer mixes input values coming from all planes, any missing plane introduces a bias in the computation that the network is not equipped to deal with.

We propose an approach wherein each modality is initially processed by its own convolutional pipeline, independently of all others. After a few independent stages, feature maps from all available modalities are *merged* by computing mapwise statistics such as the mean and the variance, quantities whose expectation does not depend on the number of terms (i.e., modalities) that are provided. After merging, the mean and variance feature maps are concatenated and fed into a final set of convolutional stages to obtain network output. This is illustrated in Figure 15.7.

In this procedure, each modality contributes a separate term to the mean and variance; in contrast to a vanilla CNN architecture, a missing modality does not throw this computation off: the mean and variance terms simply become estimated with larger uncertainty. In seeking to be robust to any subset of missing modalities, we call this approach *hetero-modal* rather than multimodal, recognizing that in addition to taking advantage of several modalities, it can take advantage of a diverse, instance-varying, set of modalities. In particular, it does not require that a "least common denominator" modality be present for every instance, as sometimes needed by common imputation methods.

Let $k \in K \subseteq \{1, \ldots, N\}$ denote a modality within the set of available modalities for a given instance, and M_k

* Note that values p and S should be chosen in a way that the pooling window fits the feature map (i.e. D should be an integer). Alternatively, we can zero pad the feature map Q accordingly.

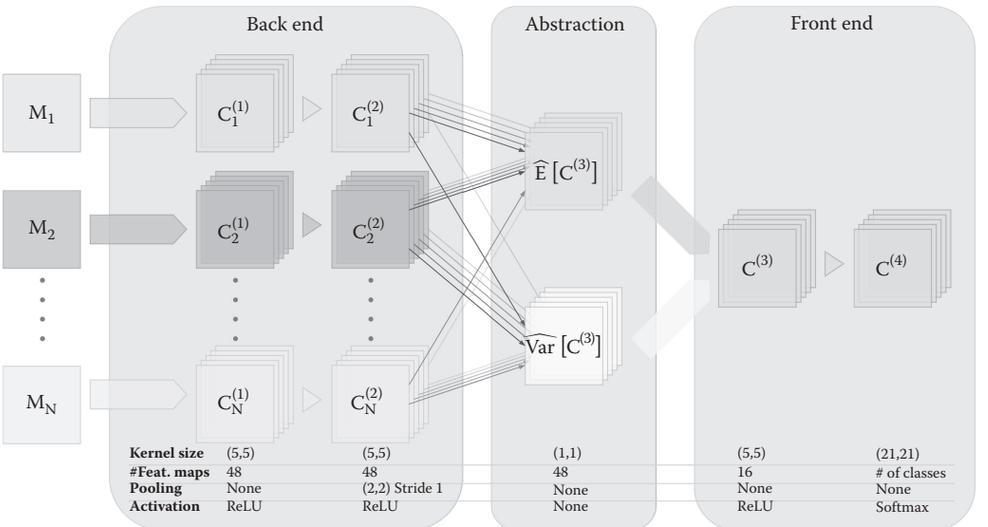

FIGURE 15.7
Illustration of the Hetero-Modal Image Segmentation architecture. Modalities available at inference time, M_k, are provided to independent modality-specific convolutional layers in the *back-end*. Feature maps statistics (first and second moments) are computed in the *abstraction layer*, which after concatenation are processed by further convolutional layers in the *front-end*, yielding pixelwise classification outputs.

represent the image of the kth modality. For simplicity, in this work we assume 2D data (e.g., a single slice of a tomographic image), but it can be extended in an obvious way to full 3D sections. As shown in Figure 15.7, HeMIS proceeds in three stages:

1. *Back-end*: In our implementation, this consists of two convolutional layers with ReLU, the second followed with a (2, 2) max-pooling layer, denoted $C_k^{(1)}$ and $C_k^{(2)}$, respectively. To ensure that the output layer consists of the same number of pixels as the input image, the convolutions are zero-padded and the stride for all operations (including max-pooling) is 1. In particular, pooling with a stride of 1 *does not downsample*, but simply "thickens" the feature maps; this is found to add some robustness to the results. The number of feature maps in each layer is given in Figure 15.7. Let $C_{k,\ell}^{(j)}$ be the ℓth feature map of $C_k^{(j)}$.
2. *Abstraction layer*: Modality fusion is computed here, as first and second moments across available modalities in $C^{(2)}$, separately for each feature map ℓ,

$$\hat{\mathrm{E}}_\ell\left[C^{(2)}\right] = \frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}} C_{k,\ell}^{(2)} \quad \text{and} \quad \widehat{\mathrm{Var}}_\ell\left[C^{(2)}\right] = \frac{1}{|\mathcal{K}|-1}\sum_{k\in\mathcal{K}}\left(C_{k,\ell}^{(2)} - \hat{\mathrm{E}}_\ell\left[C^{(2)}\right]\right)^2,$$

with $\hat{\mathrm{Var}}_\ell[C^{(2)}]$ defined to be zero if $|\mathcal{K}| = 1$ (a single available modality).
3. *Front-end*: Finally the front-end combines the merged modalities to produce the final model output. In our implementation, we concatenate all $\hat{\mathrm{E}}[C^{(2)}]$ and $\hat{\mathrm{Var}}[C^{(2)}]$ feature maps, pass them through a convolutional layer $C^{(3)}$ with ReLU activation, to finish with a final layer $C^{(4)}$ that has as many feature maps as there are target segmentation classes. The pixelwise posterior class probabilities are given by applying a softmax function across the $C^{(4)}$ feature maps, and a full image segmentation is obtained by taking the pixelwise most likely posterior class. No further postprocessing on the resulting segment classes (such as smoothing) is done.

15.3.2 Pseudo-Curriculum Training Procedure

To carry out segmentation efficiently, the model is trained fully convolutionnally to minimize a pixelwise class cross-entropy loss, in the spirit of Long et al. [20]. It has long been known that noise injection during training is a powerful technique to make neural networks more robust, as shown among others with denoising autoencoders [21], and dropout and related procedures [22]. Here, we make the HeMIS architecture robust to missing modalities by randomly dropping any number for a given training example. Inspired by previous works on curriculum learning [23]—where the model starts

learning from easy scenarios before turning to more difficult ones—we used a pseudo-curriculum learning scheme where after a few warm-up epochs where all modalities are shown to the model, we start randomly dropping modalities, ensuring a higher probability of dropping zero or one modality only.

15.3.3 Interpretation as an Embedding

An embedding is a mapping from an arbitrary source space to a target real-valued vector space of fixed dimensionality. In recent years, embeddings have been shown to yield unexpectedly powerful representations for a wide array of data types, including single words [24], variable-length word sequences [25] and images [26], and more.

In the context of HeMIS, the back-end can be interpreted as learning to separately map each modality into an *embedding common to all modalities*, within which vector algebra operations carry well-defined semantics. As such, computing empirical moments to carry out modality fusion is sensible. Since the model is trained entirely end-to-end with backpropagation, the key aspect of the architecture is that this embedding only needs to be defined implicitly as that which minimizes the overall training loss. Cross-modality interactions can be captured within specific embedding dimensions, as long as there are a sufficient number of them (i.e., enough feature maps within $C^{(2)}$), as they can be combined by $C^{(3)}$.

With this interpretation, the back-end consists of a modular assembly of operators, viewed as reusable building blocks that may or may not be needed for a given instance, each computing the embedding from its own input modality. These projections are summarized in the abstraction layer (with a mean and variance, although additional summary statistics are simple to entertain), and this summary further processed in the front-end to yield the final model output.

15.4 Data and Implementation Details

We studied the HeMIS framework on two neurological pathologies: MS with the MS Grand Challenge (MSGC) and a large relapsing–remitting MS (RRMS) cohort, as well as glioma with the brain tumor segmentation (BRATS) dataset [27].

15.4.1 MS Datasets

MSGC: The MSGC dataset [28] provides 20 training MR cases with manual ground truth lesion segmentation and 23 testing cases from the Boston Children's Hospital (CHB) and the University of North Carolina (UNC). We downloaded* the co-registered T1W, T2W, FLAIR images for all 43 cases as well as the ground truth lesion mask images for the 20 training cases. While lesion masks for the 23 testing cases are not available for download, an automated system is available to evaluate the output of a given segmentation algorithm.

RRMS: This dataset is obtained from a multi-site clinical study with 300 RRMS patients (mean age 37.5 years, SD 10.0 years). Each patient underwent an MRI that included sagittal T1W, T2W, and T1 post-contrast (T1C) images. The MRI data were acquired on 1.5 T scanners from different manufacturers (GE, Philips, and Siemens).

15.4.2 Brain Tumors (BRATS)

The *BRATS-2015* dataset contains 220 subjects with high-grade and 54 subjects with low-grade tumors. Each subject contains four MR modalities (FLAIR, T1W, T1C, and T2) and comes with a voxel level segmentation ground truth of five labels: *healthy*, *necrosis*, *edema*, *non-enhancing tumor*, and *enhancing tumor*. As done by Menze et al. [27], we transform each segmentation map into three binary maps, which correspond to three tumor categories, namely, *Complete* (which contains all tumor classes), *Core* (which contains all tumor subclasses except "edema"), and *Enhancing* (which includes the "enhanced tumor" subclass). For each binary map, the Dice similarity coefficient (DSC) is calculated [27].

BRATS-2013 contains two test datasets; Challenge and Leaderboard. The Challenge dataset contains 10 subjects with high-grade tumors while the Leaderboard dataset contains 15 subjects with high-grade tumors and 10 subjects with low-grade tumors. There is no ground truth provided for these datasets and thus quantitative evaluation can be achieved via an online evaluation system [27]. In our experiments, we used Challenge and Leaderboard datasets to compare the HeMIS segmentation performance to the state-of-the-art, when trained on all modalities.

15.4.3 Pre-Processing and Implementation Details

Before being provided to the network, bias field correction [29], and intensity normalization with a zero mean, truncation of 0.001 quantile and unit variance is applied to the image intensity. The multi-modal images are co-registered to the T1W and interpolated to 1 mm isotropic resolution.

We used Keras library [30] for our implementation. To deal with class imbalance, we adopt the patch-wise training procedure described by Havaei et al. [31]. We

* http://www.nitrc.org/projects/msseg/

first train the model with a balanced dataset that allows learning features that are agnostic to the class distribution. In a second phase, we train only the final classification layer with a distribution close to the ground truth. This ensures that we learn good features yet keep the correct class priors. The method was trained using an Nvidia TitanX GPU, with a stochastic gradient learning rate of 0.001, decay constant of 0.0001, and Nesterov momentum coefficient of 0.9 [32]. For both BRATS-2015 and MS, we split the dataset into three separate subsets—train, valid, and test—with ratios of 70%, 10%, and 20% respectively. To avoid over-fitting, we used early stopping on the validation set.

15.5 Experiments and Results

The main segmentation evaluation metric that we use is the DSC [33], which represents a normalized measure of overlap between the proposed segmentation and the ground truth, defined as

$$DSC(P,T) = \frac{|P_1 \wedge T_1|}{\frac{1}{2}(|P_1| + |T_1|)},$$

where $P \in \{0, 1\}$ is the mask volume of the predictions made by the model, $T \in \{0, 1\}$ is the mask volume of the ground truth segmentation, P_1 and T_1 respectively represent the set of voxels with $P = 1$ and $T = 1$, $|\cdot|$ is the set cardinality operator, and $\wedge$ is the logical "and" operator. These regions are illustrated in Figure 15.8.

We first validate HeMIS performance against state-of-the-art segmentation methods on the two challenge datasets: MSGC and BRATS. Since the test data and the ranking table for BRATS 2015 are not available, we submitted results to BRATS 2013 challenge and leaderboard. These results are presented in Table 15.1.* As we observe, HeMIS outperforms Tustison et al. [34], the winner of the BRATS 2013 challenge, on most tumor region categories.

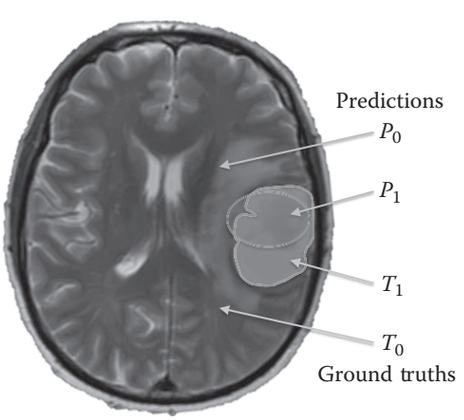

FIGURE 15.8
Illustration of the "Ground truths" and "Predictions" regions that are used in the computation of the DSC.

The MSGC dataset illustrates a direct application of HeMIS flexibility as only three modalities (T1W, T2W, and FLAIR) are provided for a small training set. Therefore, given the small number of subjects, we first trained HeMIS on RRMS dataset with four modalities and fine-tuned on MSGC. Our results were submitted to the MSGC website,† with a results summary appearing in Table 15.2. The MSGC segmentation results include three other supervised approaches: a Gaussian mixture model winner of the MSGC 2008 [35], a random decision forest classifier [8], and a more similar approach to ours using a deep convolutional encoder [9]. When compared to them, HeMIS obtains highly competitive results with a combined score of 83.2%, where 90.0% would represent human performance given inter-rater variability. This score is higher than the Gaussian mixture (80.0%) and the random forest (82.1%) a somewhat lower than the auto-encoder approach (84.0%).

For each rater (CHB and UNC), we provide the volume difference (VD), surface distance (SD), true positive rate (TPR), false positive rate (FPR), and the method's score as in Styner et al. [28].

The main advantage of HeMIS lies in its ability to deal with missing modalities, specifically when different subjects are missing different modalities. To illustrate the model's flexibility in such circumstances, we compare HeMIS performance to two common approaches to deal with random missing modalities. The first, *mean filling*, is to replace a missing modality by the modality's mean value. In our case, since all means are zero by construction, replacing a missing modality by zeros can be viewed as imputing with the mean. The second approach is to train an MLP to predict the expected value of specific missing modality given the available ones. Since neural networks are generally trained for a unique task, we need to train 28 different MLPs (one for each ◦ in Table 15.3 for a given dataset) to account for different possibilities of missing modalities. We used the same MLP architecture for all these models, which consists of 2 hidden layers with 100 hidden units each, trained to minimize the mean squared error. Figure 15.9 shows an example of predicted modalities for an MS patient.

The table shows the DSC for all possible configurations of MRI modalities being either absent (◦) or present (•), in order of FLAIR (F), T1W (T_1), T1C (T_1c), and

* Note that the results mentioned in Table 15.1 are from methods competing in the BRATS 2013 challenge for which a static table is provided at https://www.virtualskeleton.ch/BraTS/StaticResults 2013. Since then, other methods have been added to the scoreboard but for which no reference is available.

† http://www.ia.unc.edu/MSseg

TABLE 15.1

Comparison of HeMIS When Trained on All Modalities against BRATS-2013 Leaderboard and Challenge Winners, in Terms of Dice Similarity

	Leaderboard			Challenge		
Method	**Complete**	**Core**	**Enhancing**	**Complete**	**Core**	**Enhancing**
Tustison et al. [34]	79	65	53	87	**78**	**74**
Zhao et al. [36]	79	59	47	84	70	65
Menze et al. [27]	72	60	53	82	73	69
HeMIS	**83**	**67**	**57**	**88**	75	**74**

Source: Scores from Menze, B. et al., (2015, Oct), *IEEE TMI 34*(10), 1993–2024.

TABLE 15.2

Results of the Full Dataset Training on the MSGC

Method	**Rater**	**VD (%)**	**SD (mm)**	**TPR (%)**	**FPR (%)**	**Score**
Souplet et al. [35]	CHB	86.4	8.4	58.2	70.6	80.0
	UNC	57.9	7.5	49.1	76.3	
Geremia et al. [8]	CHB	**52.4**	**5.4**	59.0	71.5	82.1
	UNC	**45.0**	**5.7**	51.2	76.7	
Brosch et al. [9]	CHB	63.5	7.4	47.1	**52.7**	**84.0**
	UNC	52.0	6.4	**56.0**	**49.8**	
HeMIS	CHB	127.4	7.5	**66.1**	55.3	83.2
	UNC	68.2	6.6	52.3	61.3	

TABLE 15.3

DSC Results on the RRMS and BRATS Test Sets (%) When Modalities Are Dropped

				RRMS			BRATS								
Modalities				**Lesion**			**Complete**			**Core**			**Enhancing**		
F	T_1	T_1c	T_2	HeMIS	Mean	MLP	HeMIS	Mean	MLP	HeMIS	Mean	MLP	HeMIS	Mean	MLP
◦	◦	◦	•	1.74	2.66	**12.77**	58.48	2.70	**61.50**	**40.18**	4.00	37.32	**20.31**	6.25	18.62
◦	◦	•	◦	2.67	0.00	**3.51**	**33.46**	23.11	2.04	**44.55**	23.90	17.70	**49.93**	30.02	32.92
◦	•	◦	◦	3.89	0.00	**6.64**	**33.22**	0.00	2.07	**17.42**	0.00	10.52	4.67	6.25	**10.78**
•	◦	◦	◦	34.48	9.77	**38.46**	71.26	**72.30**	63.81	**37.45**	0.00	34.26	5.57	6.25	**15.90**
◦	◦	•	•	**27.52**	4.31	25.83	**67.59**	35.01	64.97	**63.39**	30.92	49.38	**65.38**	39.00	60.30
◦	•	•	◦	8.21	0.00	**8.26**	**45.93**	23.63	1.99	**55.06**	41.89	26.55	**62.40**	43.80	40.93
•	•	◦	◦	38.81	11.62	**39.15**	**80.28**	75.58	78.13	**49.52**	0.00	48.97	22.26	6.25	**25.18**
◦	•	◦	•	**31.25**	8.31	29.39	**69.56**	1.77	66.88	**47.26**	2.63	43.66	23.56	6.25	**26.37**
•	◦	◦	•	**39.64**	33.31	38.55	**82.1**	81.01	81.35	**53.42**	25.94	52.41	23.19	6.25	**25.01**
◦	◦	•	◦	**41.38**	6.42	39.33	79.8	45.97	**81.13**	**66.12**	29.85	65.51	**67.12**	35.14	66.19
•	•	•	◦	**41.97**	9.00	40.63	80.88	81.57	**82.19**	69.26	62.13	**69.34**	**71.30**	67.13	70.93
•	•	◦	•	**46.6**	41.12	41.83	**83.87**	77.84	80.40	**57.76**	20.66	53.46	**28.46**	6.25	28.34
◦	◦	•	•	**41.90**	38.95	41.47	82.78	64.19	**83.37**	**70.62**	42.36	70.45	70.52	49.62	**70.56**
◦	•	•	•	**34.98**	5.78	29.46	**70.98**	30.86	67.85	**66.60**	45.79	55.40	**67.84**	50.21	64.81
•	•	•	•	**48.66**	43.48	43.48	**83.15**	82.43	82.43	**72.5**	71.46	71.46	**75.37**	72.08	72.08
# Wins / 15				**9**	0	6	**10**	1	4	**14**	0	1	**9**	0	6

T2W (T_2). Results are reported for HeMIS, mean (mean filling), and the imputation MLP (MLP).

Table 15.3 shows the DSC for this experiment on the test set.

On the BRATS dataset, for the Core category, HeMIS achieves the best segmentation in almost all cases (14 out of 15), and for the Complete and Enhancing categories, it leads in most cases (10 and 9 cases out of 15, respectively). Also, the mean-filling approach hardly outperforms HeMIS or MLP imputation. These results are consistent with the MS lesion segmentation dataset, where HeMIS outperforms other imputation approaches in 9 out of 15 cases. In scenarios where only one or two modalities are missing, while both HeMIS and MLP imputation obtain good results, HeMIS outperforms the latter in most cases on both datasets.

In cases where two modalities are missing, HeMIS performance is either close to or much higher than that of MLP across all categories.

On BRATS, when missing three out of four modalities, HeMIS outperforms the MLP in a majority of cases. Moreover, whereas the HeMIS performance only gradually drops as additional modalities become missing, the performance drop for MLP imputation and mean filling is much more severe. On the RRMS cohort, the MLP imputation appears to obtain slightly better segmentations when only one modality is available.

Although it is expected that tumor sub-label segmentations should be less accurate with fewer modalities, we should still hope for the model to report a sensible characterization of the tumor "footprint." While MLP and mean filling fail in this respect, HeMIS quite well achieves this goal by outperforming in almost all cases of the Complete and Core tumor categories. This can also be seen in Figure 15.10 where we show how adding modalities to HeMIS improves its ability to achieve a more accurate segmentation. From Table 15.3, we can also infer that the FLAIR modality is the most relevant for identifying the Complete tumor, while T1C is the most relevant for identifying Core and Enhancing tumor categories. On the RRMS dataset, HeMIS results are also seen to degrade slower than the other imputation approaches, preserving good segmentation when modalities go missing. Indeed, as seen in Figure 15.10, even though with FLAIR alone HeMIS already produces good segmentations, it is capable of further refining its results when adding modalities by removing false positives and improving outlines of the correctly identified lesions or tumor.

15.6 Conclusion

We have proposed a new fully automatic segmentation framework for heterogenous multi-modal MRI using a specialized convolutional deep neural network. The embedding of the multi-modal CNN back-end allows to train and segment datasets with missing modalities. We carried out an extensive validation on MS and glioma and achieved state-of-the art segmentation results on two challenging neurological pathology image processing tasks. Importantly, we contrasted the graceful performance degradation of the proposed approach as modalities go missing, compared with other popular imputation approaches, which it achieves without requiring training specific models for every potential missing modality combination.

This type of model offers a direction to fuse data from multiple imaging modalities, making it a step to deal with the information glut arising from the rapid increase in scans performed worldwide.

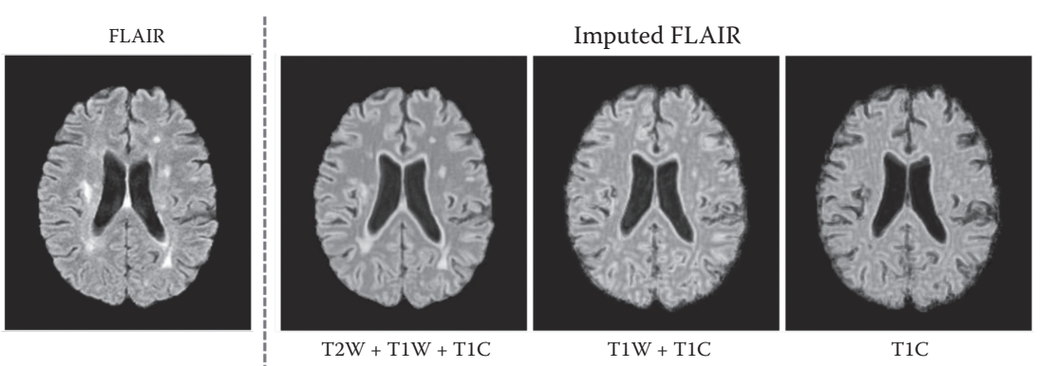

FIGURE 15.9
MLP-imputed FLAIR for an MS patient. The figure shows from left to right the original modality and the predicted FLAIR given other modalities.

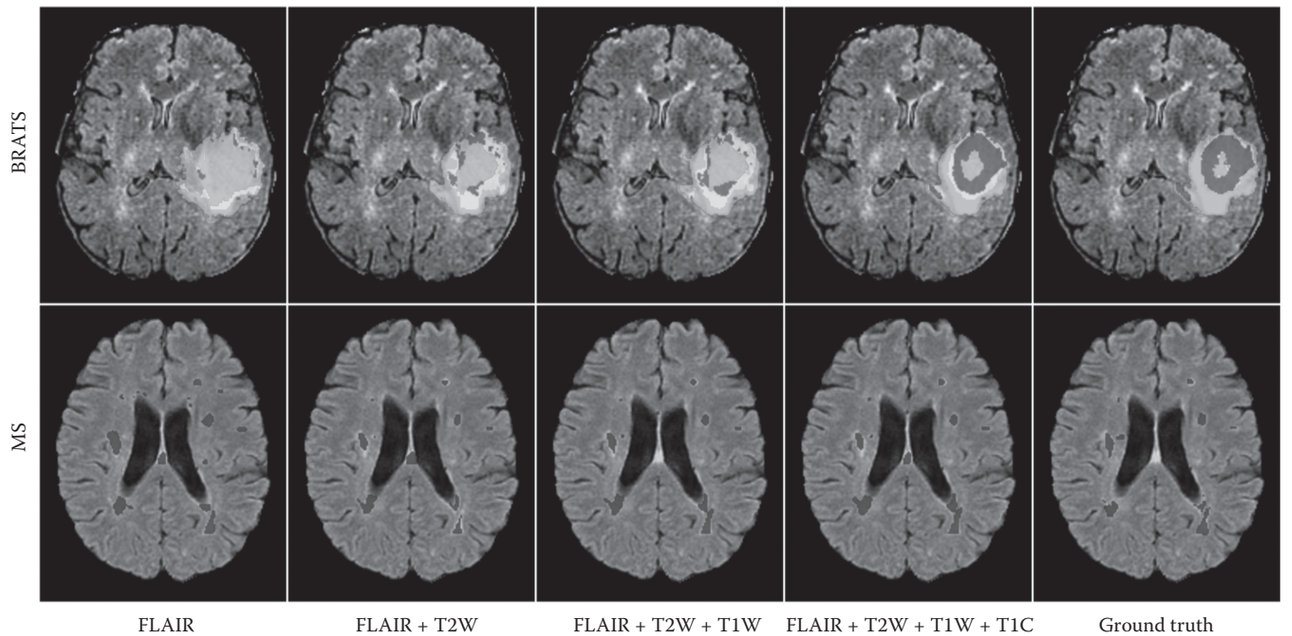

FIGURE 15.10
Example of HeMIS segmentation results on BRATS and MS subjects for different combinations of input modalities. For both cohorts, an axial FLAIR slice of a subject is overlaid with the results where for BRATS (first row) the segmentation colors describe necrosis (blue), non-enhancing (yellow), active core (orange), and edema (green). For the MS case, the lesions are highlighted in red. The columns present the results for different combinations of input modalities, with ground truth in the last column.

References

1. OECD (2016). Health at a glance, 2015.
2. U.S. Preventive Services Task Force (2014). Screening for lung cancer: U.S. preventive services task force recommendation statement. *Annals of Internal Medicine* 160(5), 330–338.
3. Canadian Task Force on Preventive Health Care (2016). Recommendations on screening for lung cancer. *Canadian Medical Association Journal*.
4. Iglesias, J. E. and M. R. Sabuncu (2015). Multi-atlas segmentation of biomedical images: A survey. *Medical Image Analysis* 24(1), 205–219.
5. Havaei, M., N. Guizard, N. Chapados, and Y. Bengio (2016). *HeMIS: Hetero-Modal Image Segmentation*, pp. 469–477. Springer International Publishing.
6. Guizard, N., P. Coupé, V. S. Fonov, J. V. Manjón, D. L. Arnold, and D. L. Collins (2015). Rotation-invariant multi-contrast non-local means for ms lesion segmentation. *NeuroImage: Clinical* 8, 376–389.
7. Cordier, N., H. Delingette, and N. Ayache (2016). A patch-based approach for the segmentation of pathologies: Application to glioma labelling. *IEEE TMI* PP(99), 1.
8. Geremia, E., B. H. Menze, and N. Ayache (2013). Spatially adaptive random forests. Biomedical Imaging (ISBI), *2013 IEEE 10th International Symposium on*, pp. 1344–1347.
9. Brosch, T., Y. Yoo, L. Y. W. Tang, D. K. B. Li, A. Traboulsee, and R. Tam (2015). *MICCAI Proceedings*, Chapter "Deep convolutional encoder networks for multiple sclerosis lesion segmentation," pp. 3–11. Springer.
10. Van Buuren, S. (2012). *Flexible Imputation of Missing Data*. CRC Press, Boca Raton, FL.
11. Hofmann, M., F. Steinke, V. Scheel, G. Charpiat, J. Farquhar, P. Aschoff, M. Brady, B. Schölkopf, and B. J. Pichler (2008). MRI-based attenuation correction for PET/MRI: A novel approach combining pattern recognition and atlas registration. *Journal of Nuclear Medicine* 49(11), 1875–1883.
12. Tulder, G. and M. Bruijne (2015). *MICCAI Proceedings*, Chapter, "Why does synthesized data improve multi-sequence classification?" pp. 531–538. Springer.
13. Hor, S. and M. Moradi (2015). Scandent tree: A random forest learning method for incomplete multimodal datasets. In *MICCAI 2015*, pp. 694–701. Springer.
14. Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. Book in preparation for MIT Press. URL http://www.deeplearningbook.org.
15. Dumoulin, V. and F. Visin (2016, March). A guide to convolution arithmetic for deep learning. *ArXiv e-prints*.
16. Jarrett, K., K. Kavukcuoglu, M. Ranzato, and Y. LeCun (2009, Sept). What is the best multi-stage architecture for object recognition? In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2146–2153.

17. Glorot, X., A. Bordes, and Y. Bengio (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 513–520.
18. Goodfellow, I. J., D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio (2013). Maxout networks. In *ICML*.
19. Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). Imagenet classification with deep convolutional neural networks. In *NIPS*, Volume 1, pp. 1097–1105. NIPS.
20. Long, J., E. Shelhamer, and T. Darrell (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE CVPR Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440.
21. Vincent, P., H. Larochelle, Y. Bengio, and P.-A. Manzagol (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103. ACM.
22. Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1), 1929–1958.
23. Bengio, Y., J. Louradour, R. Collobert, and J. Weston (2009). Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48. ACM.
24. Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013). Distributed representations of words and phrases and their compositionality. In *NIPS*, pp. 3111–3119.
25. Sutskever, I., O. Vinyals, and Q. V. Le (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pp. 3104–3112.
26. Xu, K., J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, A. Bengio, Yoshuaand Courville, R. Salakhudinov, R. Zemel, and Y. Bengio (2015). Show, attend and tell: Neural image caption generation with visual attention. In D. Blei and F. Bach (Eds.), *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 2048–2057. JMLR Workshop and Conference Proceedings.
27. Menze, B., A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, and J. e. a. Kirby (2015, Oct). The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE TMI* 34(10), 1993–2024.
28. Styner, M., J. Lee, B. Chin, M. Chin, O. Commowick, H. Tran, S. Markovic-Plese, V. Jewells, and S. Warfield (2008). 3D segmentation in the clinic: A grand challenge ii: Ms lesion segmentation. *MIDAS* 2008, 1–6.
29. Sled, J. G., A. P. Zijdenbos, and A. C. Evans (1998). A nonparametric method for automatic correction of intensity nonuniformity in MRI data. *IEEE TMI* 17(1), 87–97.
30. Chollet, F. (2015). Keras. URL https://github.com/keras-team/keras.
31. Havaei, M., A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle (2015). Brain tumor segmentation with deep neural networks. *arXiv preprint arXiv:1505.03540*.
32. Sutskever, I., J. Martens, G. Dahl, and G. Hinton (2013). On the importance of initialization and momentum in deep learning. In *ICML-13*, pp. 1139–1147.
33. Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology* 26(3), 297–302.
34. Tustison, N. J., K. Shrinidhi, M. Wintermark, C. R. Durst, B. M. Kandel, J. C. Gee, M. C. Grossman, and B. B. Avants (2015). Optimal symmetric multimodal templates and concatenated random forests for supervised brain tumor segmentation (simplified) with ANTsR. *Neuroinformatics* 13(2), 209–225.
35. Souplet, J., C. Lebrun, N. Ayache, and G. Malandain (2008, 07). An automatic segmentation of T2-FLAIR multiple sclerosis lesions.
36. Zhao, L., W. Wu, and J. J. Corso (2013). *MICCAI Proceedings*, Chapter, "Semi-automatic brain tumor segmentation by constrained MRFs using structural trajectories," pp. 567–575. Springer.