CHAPTER 1

Characterisation of errors in deep learning-based brain MRI segmentation

Akshay Pai[a,*,†], Yuan-Ching Teng[†], Joseph Blair[**], Michiel Kallenberg[*], Erik B. Dam[*], Stefan Sommer[†], Christian Igel[†] and Mads Nielsen[a,*,†]

* Biomediq A/S, Copenhagen, Denmark.
** University of Copenhagen, Department of Biology, Copenhagen, Denmark.
† University of Copenhagen, Department of Computer Science, Copenhagen, Denmark.
[a] Corresponding: akshay@biomediq.com, madsn@biomediq.com

Contents

Abstract

With ever-increasing data in the field of medical imaging, the availability of robust methods for quantitative analysis in large-scale studies is the need of the hour. In recent times, there has been a significant increase in the use of deep learning, in particular of convolution neural networks (CNNs), in the field of computer vision and image analysis. In contrast to traditional shallow classifiers, deep learning methods need less domain-specific feature engineering. The architecture can automatically learn hierarchies of relevant features from

raw data. Despite the many success stories from computer vision, so far there are only rather few studies on deep learning in the field of medical imaging. In this chapter, we will look more closely at a specific application of CNNs, namely segmentation of normal brains from magnetic resonance images (MRI). We will characterize the types of errors from CNN-based segmentation and compare them with the errors from a model-based registration approach. The emphasis of this chapter is on comparing errors made by model-driven and data-driven approaches. In conclusion, we notice that the two methods make complementary errors. The CNN errors can be reduced by including more training data and by finding ways to incorporate the geometric information that registration-based algorithms rely on.

Chapter points

- Deep Learning methods require large training datasets.
- Geometric information of the anatomy improves the performance of deep learning for medical image segmentation.
- In our experiments, deep learning methods and registration-based methods produced complimentary types of errors. Thus, combining model-based and data-driven segmentation approaches is a promising future research direction.

1. Introduction

Quantitative analysis of medical images often requires segmentation of anatomical structures observed in them. For instance, the volume of the hippocampus in the brain is associated with Dementia of the Alzheimer's type [14]. In addition to volume quantification, segmentation aids in the analysis of regional statistics such as shape, structure, and texture. Segmentation also extends to detecting abnormal biological processes or even pathological anomalies, for instance microbleeds, tumors so on and so forth. In this chapter, we will specifically deal with segmentation of normal brains from magnetic resonance images (MRI) using deep learning. The purpose is to evaluate and understand the characteristics of errors made by deep learning approaches as opposed to a model-based approach such as segmentation based on multi-atlas nonlinear registration. In contrast to the deep learning approach, registration-based methods rely heavily on topological assumptions about the objects in the image, i.e., that the anatomical structures are similar enough to be mapped onto each other.

Segmentation essentially involves dividing images into meaningful regions, which can be viewed as a voxel classification task. The most simplistic approach is to manually segment brains. However, this is a time-consuming process and has significant interoperable variations. Automating delineation provides a systematic way of obtaining regions of interest in a brain on the fly as soon as a medical image is acquired. The field of brain segmentation is dominated by multi-atlas based methods. They are driven by a combination of spatial normalization through registration followed by a classifi-

cation step, which can be simple majority voting or a more sophisticated method such as Bayesian weighting.

With the advance in computational capabilities and refinement of algorithms, deep learning based methods have seen an increased usage [27, 17]. They have proven to be extremely efficient, stable, and state-of-art in many computer vision applications. The architecture of a deep neural network is loosely inspired by the human cortex [16]. It consists of several layers, each performing a non-linear transformation. In contrast to most traditional machine learning approaches where features typically need to be handcrafted to suit the problem at hand, deep learning methods aim at automatically learning features from raw or only slightly pre-processed input. The idea is that each layer adds a higher level of abstraction—a more abstract feature representation—on top of the previous one.

This chapter will focus on the application of deep neural networks, specifically convolutional neural networks (CNNs), to brain MRI segmentation. After a short introduction, we will evaluate the accuracy of the CNNs compared to a standard multi-atlas registration based method. We reproduce results from [7] on the MICCAI 2012 challenge dataset. We will focus on the characterization of the errors made by CNNs in comparison with the registration-based method. Finally, we will discuss some directions for future work such as ensemble learning of multiple classifiers including CNNs.

2. Deep Learning for Segmentation

Most segmentation methods either rely soley on intensity information (e.g., appearance models) or combine intensity and higher order structural information of objects. A popular example of the latter is multi-atlas registration, which is based on a specific set of templates that, in a medical application, typically contains labelings of anatomical structures. An image registration algorithm then maps these templates to a specific subject space through a non-linear transformation. Voting schemes are then applied to choose the right label from a given set of labels obtained from the transformed templates [6].

In a typical model-based approach, images are first registered to a common anatomical space and then specific features are extracted from the aligned images. The features are often higher order statistics of a region such as histograms of gradients or curvatures. Learning algorithms such as support vector machines (SVMs) or neural networks are then used to classify regions in a medical image. *Fischl et al* [10] have integrated both learning and model based methods for segmentation. Images are normalized to a common template, and then a Markov random field learns anatomically corresponding positions and intensity patterns for different anatomical structures. In the above-mentioned algorithms, domain knowledge about spatial locations of various

anatomical structures is incorporated as a Bayesian prior distribution of the segmentation algorithm.

In contrast, data-driven deep learning approaches aim at learning a multi-level feature representation from image intensities, typically not using anatomical background knowledge about spatial relations. The feature representation at each level results from a non-linear transformation of the representations at the previous level. This is in contrast to shallow architectures, which have at most one distinguished layer of non-linear processing (i.e., one level of data abstraction). Popular shallow architectures include linear methods such as logistic regression and linear discriminant analysis as well as kernel methods such as SVMs and Gaussian processes. Kernel classifiers use kernel functions to map input from a space where linear separation is difficult to a space where the separation is easier. The class of kernel functions, and therefore the feature representation, has to be chosen *a priori*. Because deep learning methods learn the feature representation, it has been argued that they have the advantage of being less dependent on prior knowledge of the domain. However, feature learning requires that enough training data is available to learn the representations. Deep neural networks are, in general, highly complex non-linear models having many free parameters. Sufficient training data is needed to learn these parameters and to constrain the model such that it generalizes well to unseen data. Therefore, deep learning methods excel in applications where there is an abundance of training data—which is typically not the case in medical imaging where data is often scarce.

Accordingly, so far there are only rather few applications of deep learning methods in medical imaging, specifically brain segmentation. A prominent example of CNNs for classification applied to 2D medical images is mitosis detection - winning the MICCAI 2013 Grand Challenge on Mitosis Detection [5, 33, 32]. Some more recent notable applications of deep learning in the field of medical image analysis are collected in [12]. Highlights include classification of brain lesions [4, 23], microbleeds in brain [8], histopathalogical images [29], colonic polyp [31], lung nodules [28], and mammograms [15].

Limitations of deep learning applications in medical image analysis are often attributed to computational complexity and large variance in the data. In order to address these methodological contributions have been made to improve both the robustness, and computational complexity of CNN. For instance, the tri-planar approach was proposed to overcome the computational complexity of the 3D CNNs[24] and later *Roth et al.,* [25] propose random rotations of the triplanar views as way to obtain representations of the 3D data. *Brosch et al.,* [4] use a combination of convolutional and deconvolutional layers to learn different feature scales.

3. Convolutional Neural Network Architecture

In the following, we describe the convolutional neural network (CNN) architecture we considered in this study, which is depicted in Fig. 1.1. An in-depth introduction to neural networks and CNNs is beyond the scope of this chapter, we refer to [11].

3.1. Basic CNN Architecture

The basic principles of CNNs have been introduced by *LeCun et al.* in the late 1990s [18]. Deep neural networks consist of several layers. Different types of layers can be distinguished in a CNN. First, we have convolutional layers. This type of layer computes a convolution of its input with a linear filter (which has been suggested for modelling neural responses in the visual cortex a long time ago, e.g., [34]). The filter mask is also referred to as convolution kernel or—in analogy to neurons—as receptive field. The coefficients of each convolution kernel are learned in the same way as weights in a neural network. Typically, after the convolution a non-linear (activation) function is applied to each element of the convolution result. Thus, a convolutional layer can be viewed as a layer in a standard neural network in which the neurons share weights. The output of the layer is referred to as a *feature map*. The input to the convolutional layer can be the input image or the output of a preceding layer, that is, another feature map. The particular type of weight sharing in a convolutional layer does not only reduce the degrees of freedom, it ensures *equi-variance to translation* [11], which means that, ignoring boundary effects, if we translate the input image, the resulting feature map is translated in the same way.

Often convolutional layers are followed by a pooling layer. Pooling layers compute the maximum or average over a region of a feature map. This results in a subsampling of the input. Pooling reduces the dimensionality, increases the scale, and also supports translation invariance in the sense that a slight translation of the input does not (or only marginally) change the output.

Convolutional layers and pooling layers take the spatial structure of their input into account, which is the main reason for their good performance in computer vision and image analysis. In contrast, when standard non-convolutional neural networks are applied to raw (or only slightly preprocessed) images, the images are vectorized. The resulting vectors serve as the input to the neural network. In this process, information about the spatial relation between pixels or voxels is discarded and there is no inherent equi-variance to translation nor translation invariance, which are important properties to achieve generalization in object recognition tasks.

The final layers in a CNN correspond to a standard multi-layer perceptron neural network [2], which receives the vectorized (or *flattened*) feature maps of the preceding layer as input. There can be zero, one, or more hidden layers, which compute an affine linear combination of their inputs and apply a non-linear activation function to it. The

final layer is the output layer. For multi-class classification, it typically computes the softmax function, turning the input into a probability distribution over the classes. For learning, we can then consider the cross-entropy error function, which corresponds to minimizing the negative logarithmic likelihood of the training data [2].

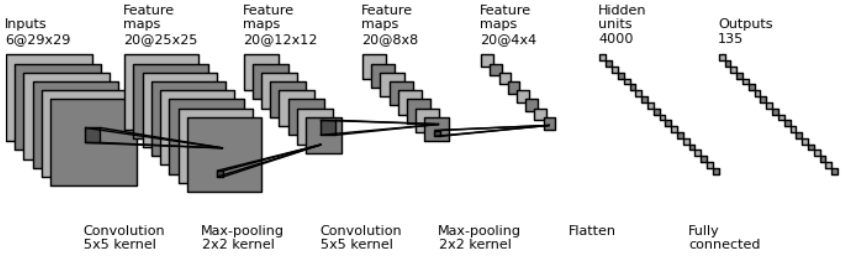

Figure 1.1 Convolutional neural network with 3 input channels. n@kxk indicates n feature maps with size $k \times k$

3.2. Tri-planar CNN for 3D Image Analysis

Processing 3D input images, such as brain scans, with CNNs is computationally demanding even if parallel hardware is utilized. To classify (e.g., segment) a voxel with a 3D CNN, an image patch around the voxel is extracted. This patch, which is a small 3D image, serves as the input to the CNN, which computes 3D feature maps and applies 3D convolution kernels. To reduce the computational complexity and the degrees of freedom of the model, *tri-planar* CNNs can be used instead of a 3D CNN. A tri-planar CNN for 3D image analysis considers three two-dimensional image patches, which are processed independently in the convolutional and pooling layers; all feature maps and convolutions are two-dimensional. For each voxel, one $n \times n$ patch is extracted in each image plane (sagittal, coronal and transverse), centered around the voxel to be classified.

This tri-planar approach [24, 26] reduces the computational complexity compared to using three-dimensional patches, while still capturing spatial information in all three planes. Due to the reduction in computation time, it is often possible to consider larger patches, which incorporate more distant information.

The tri-planar architecture considered in this study is sketched in Figure 1.1. It consists of two convolutional layers (including applying non-linear activation functions) each followed by a pooling layer. The latter compute the maximum over the pooling regions. Then the feature maps are *flattened* and fed into a hidden layer with 4000 neurons. As activation functions we use the unbounded Rectifier linear unit (ReLU) $a \mapsto \max(0, a)$, which has become the standard activation function in deep CNNs. The

absolute values of the derivatives of standard sigmoid activation functions are always smaller than one, which leads to the *vanishing gradient* effect in deep networks (e.g., see [1]), which is addressed by ReLUs. The number of neurons in the softmax output layer corresponds to the 135 different classes we consider in our segmentation.

The efficient tri-planer architecture allows us to consider input patches at two different scales, one for local and one for more global information. Accordingly, we have in total six two-dimensional inputs to our CNN architecture (2D patches from three planes at two scales).

4. Experiments

In this section, we will use three different feature representations for CNNs for segmenting structures in a normal brain using T1 weighted magnetic resonance images (MRI). To evaluate characteristics of the errors made by CNN, we compare the segmentations obtained to those obtained using a common multi-atlas registration-based method.

We will first evaluate the performance of CNN-based segmentation method (with and without centroid distances) in comparison with the registration-based segmentation. After that, we will characterize the errors from both the methods to understand the limitations of CNNs in segmenting brain MRIs with limited training data. For a fair evaluation of the methods, we exclude the outliers i.e., images where the registration-based method fail.

4.1. Dataset

All the experiments were performed on a dataset that was released for the MICCAI 2012 multi-atlas challenge[1]. The dataset contains 15 atlases in the training set, and 20 atlases in the testing set. All 35 images are T1-weighted structural MRIs selected from the OASIS project [19]. The cortical regions were labeled using the BrainCOLOR protocol; the non-cortical regions were labeled using the NeuroMorphometric protocol.These images were acquired in the sagittal plane and the voxel size is $1.0 \times 1.0 \times 1.0\ mm^3$. Only the labels appearing in all images are used. The labels consist of 40 cortical and 94 non-cortical regions.

4.2. CNN Parameters

All voxels in the brain were considered during the training phase. Due to similar slice thickness relative to the in-plane voxel size, we use information from both 2D and tri-planar planes for segmenting the volumetric images. The first three rows of the input layers are a set of tri-planar patches. The next three rows are also tri-planar however,

[1] https://masi.vuse.vanderbilt.edu/workshop2012/index.php/Main_Page

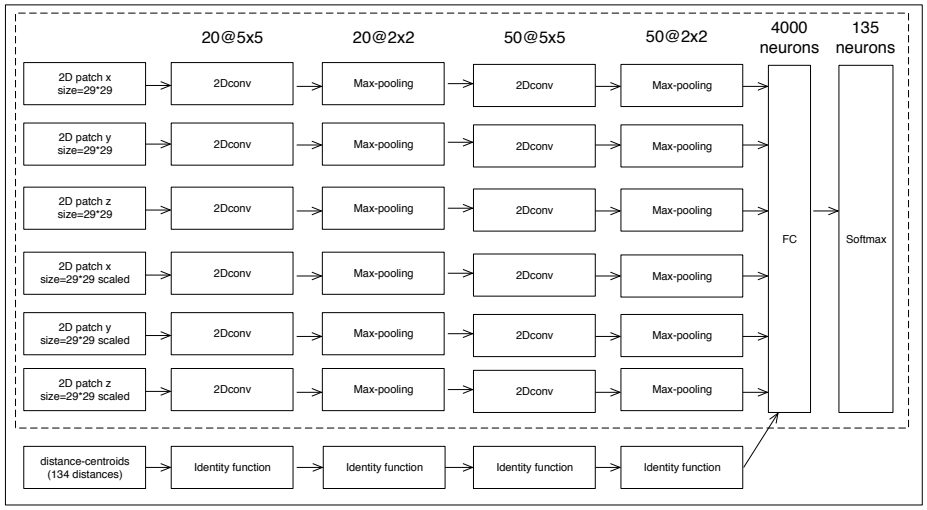

Figure 1.2 The convolutional neural network used in the experiment. The dotted box represents the architecture of CNN without centroids and the solid box (outermost) represents the architecture of the CNN with centroids as an input feature. There are 135 neurons in the softmax layer (instead of 134) since the background class is also included.

with a different patch area. First a patch with a larger (than the first three rows) size is chosen, and then the patch is downscaled to match the dimensions of the patches from the first three rows.

A CNN with multiple convolution layers was used. A schematic representation of the CNN architecture can be found in Figure 1.2. In the first convolutional layer, we trained 20 kernels. For patches of size 29×29, we used a kernel size of 5×5. Max-pooling with kernels of 2×2 was performed on the convolved images. The output images of the max-pooling layer served as input to the next layer, where 50 kernels of size 5×5 were used. In addition, max-pooling with kernels of size 2×2 was performed on the convolved images. The output of the second layer of max-pooling served as input to a fully-connected layer with 4000 nodes. The output layer of the CNN computed the soft-max activation function. This function maps its inputs to positive values summing to 1, so that the outputs can be interpreted as the posterior probabilities for the classes. The class with the highest probability is finally chosen for prediction.

4.3. Training

In order not to overfit the training set when choosing the network architecture, we randomly chose 10 images as a training set and the remaining 5 images as a validation set. We split the training dataset into $n = 4$ parts (mini-batches) so that each part is small enough to fit GPU memory while avoiding too much of I/O. We trained for 60 epochs (one epoch corresponds to training on n mini-batches). Before training, we randomly sampled 400,000 voxels evenly from the training set, 200,000 voxels evenly from the validation set for each epoch. We then extract patch features from those voxels as inputs to the network.

The validation set was used to detect overfitting. No overfitting was observed in the initial experiment and the validation error did not decrease after 10 epochs. Thus, the data from the training and validation set were again combined and the network was retrained on all 15 images for 10 epochs.

After the network was trained, images in the test set were classified. We evaluate the performance using Dice coefficients between true and the CNN-classified labels across all the 134 anatomical regions:

$$\mathrm{Dice}(A, B) = 2 \times \frac{|A \cap B|}{|A| + |B|} \quad , \tag{1.1}$$

where $A$and B are the true labels and predicted labels, respectively. A Dice score of 1 indicates perfect match.

The loss function used in this chapter can be found in [3]. The optimization was performed using stochastic gradient descent with a momentum term [3].

The code we used in this chapter is based on Theano[2], a Python-based library that enables symbolic expressions to be evaluated on GPUs [30].

4.4. Estimation of Centroid Distances

Unlike computer vision problems where segmentation tasks are more or less unstructured, brain regions are consistent in their spatial position. To incorporate this information, *Brebisson et al.* [3] used relative centroid distances as an input feature. Relative centroid distance is the Euclidean distance between each voxel and the centroid of each class.

In the training phase, the true labels were used to generate the centroids. In the testing phase, the trained network without the centroids (see the dotted box in Figure 1.2), was used to generate an initial segmentation from which centroid distances were extracted. This layer may be replaced by any segmentation method or a layer that provides relevant geometric information (e.g., Freesurfer [10]). We tried replacing the

[2]http://deeplearning.net/software/theano/

Predicted/True	Segment	Non-Segment
Segment	TP	FP
Non-Segment	FN	TN

Table 1.1 Confusion matrix used to compute error metrics such as Dice and Jaccard. TP: true positive, FP: false positive, TN: true negative, and FN: false negative.

initial segmentations used to generate centroid distances by a registration-based segmentation. However, this did not yield any improvement in the Dice scores. This indicates that including geometric information must be more intricate that just incorporating spatial positions. In the experiments, we therefore used CNNs to generate initial segmentations and the corresponding centroid distances during testing. The final CNN network is illustrated in Figure 1.2.

4.5. Registration-based Segmentation

In the MICCAI challenge, the top performing methods relied significantly on non-linear image registration. Hence, we chose to compare to a registration-based segmentation method. Since the purpose of the chapter is to only evaluate the types of error made by a registration-based method, we sticked to a simple majority voting scheme for classification.

We first linearly registered the images using an inverse consistent affine transformation with 12 degrees freedom and mutual information as a similarity measure. After that, we performed an inverse consistent diffeomorphic non-linear registration using the kernel bundle framework and stationary velocity fields [22]. The parameters of the registration were the same as in [22]. We used a simple majority voting to fuse the labels. More sophisticated voting schemes were avoided since they can be viewed as an ensemble layer which optimizes classifications obtained from registration.

4.6. Characterization of Errors

In order to assess performances of both the CNN and the registration-based method in depth, we characterize the errors made by the two methods. Typically, segmentation errors are computed via either Dice or Jaccard indices, which are computed from the confusion matrix, see Table 1.1:

$$\text{DiceScore} = \frac{2\text{TP}}{|\ \text{TP} + \text{FN}\ | + |\ \text{TP} + \text{FP}\ |} = \frac{2\text{TP}}{2\text{TP} + \text{FN} + \text{FP}} \sim \frac{2\text{TP} + 1}{2\text{TP} + \text{FN} + \text{FP} + 1} \tag{1.2}$$

$$\text{Jaccard} = \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}} \sim \frac{\text{TP} + 1}{\text{TP} + \text{FN} + \text{FP} + 1} \tag{1.3}$$

Predicted/True	Segment	ϵ region	Non-Segment
Segment	TP	BE-1	FP
ϵ region	BE-2	TB	BE-3
Non-Segment	FN	BE-4	TN

Table 1.2 Updated confusion matrix including boundaries indecisions. BE: Boundary errors, and TB: True boundary. With an additional definition of boundary errors, one can come up with new measures that characterize different types of errors made by the segmentation method.

For instance, the Dice is a size of the overlap of the two segmentations divided by the mean size of the two objects. This can be expressed in terms of the entries of the Table 1.1. Similarly, measures such as sensitivity, specificity may also be expressed in terms of the table entries:

$$\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \sim \frac{\text{TP} + 1}{\text{TP} + \text{FN} + 1} \tag{1.4}$$

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \sim \frac{\text{TN} + 1}{\text{TN} + \text{FP} + 1} \tag{1.5}$$

Errors in segmentations may have several characteristics. Sometimes, boundaries of regions are a source of noise. Other errors arise when lumps of regions are given the wrong-label. Let us consider ϵ-boundary regions defined by ϵ-erosion and ϵ-dilation as shown in Figure 1.3. When we exclude the ϵ-boundary when assessing the segmentation quality, see Table 1.2 and Figure 1.3, we get new quality measures:

$$\text{Dice}_\epsilon = \frac{2\text{TP}_\epsilon + 1}{2\text{TP}_\epsilon + \text{FN}_\epsilon + \text{FP}_\epsilon + 1} \tag{1.6}$$

$$\text{Jaccard}_\epsilon = \frac{\text{TP}_\epsilon + 1}{\text{TP}_\epsilon + \text{FN}_\epsilon + \text{FP}_\epsilon + 1} \tag{1.7}$$

$$\text{sensitivity}_\epsilon = \frac{\text{TP}_\epsilon + 1}{\text{TP}_\epsilon + \text{FN}_\epsilon + 1} \tag{1.8}$$

Typically, the boundary errors can be identified by moving the boundaries of the segmentations through morphological operations. Examples illustrating these errors will be given in the results section. In a boundary error, since the core (excluding the indecisive boundaries) of the segmentations are correctly classified, the Dice scores are expected to be high. However, in labelling error, in addition to the boundary errors, there may be lumps of regions with wrong labels which will negatively affect the Dice score. In order to formalize these effects, we define a measure called the core-Dice

score (equation (1.9)):

$$\text{cDice}(A, B) = \frac{1 + 2 \times |\,\text{core}(A) \cap \text{core}(B)|}{1 + |\,\text{core}(A) \cap \text{non-boundary}(B)| + |\,\text{core}(B) \cap \text{non-boundary}(A)|} \tag{1.9}$$

where core is the inner segmentation after excluding ϵ boundaries, non-boundary$(B) = B \setminus \text{boundary}(B)$, and non-boundary$(A) = A \setminus \text{boundary}(A)$. A constant 1 added in both denominator and divider is to avoid the divide-by-zero error when the width of boundary is so large that all the regions are ignored. The value converges to 1 when the core regions are perfectly matched or totally isolated. We can, therefore, estimate where the errors come from by calculating the core dice score with increasing width of boundary. Typically, core-Dice errors will tend to increase as a function of the boundary width (see Figure 1.3) where as segmentations with labeling errors will behave vice versa.

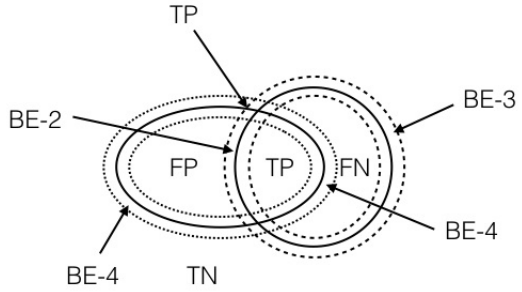

Figure 1.3 Augmentation of the region by width of ϵ. Defining an object's boundary from ϵ-erosion and ϵ-dilation: $\frac{A \cup S_\epsilon}{A \setminus S_\epsilon}$, where S_ϵ is a structural element defined as a sphere of radius ϵ

5. Results

This section will present results of the overall performance of the CNN algorithm in comparison to the registration-based algorithm. The results were divided into two categories: a) overall performance of the method in comparison to the registration-based method, b) evaluation of the types of errors made by the CNN and registration-based methods.

5.1. Overall Performance

Table 1.3 enlists the mean Dice scores obtained by the CNN and registration-based methods across the 134 regions and all the images. The CNN with spatial information, in the form of centroid, gave better Dice scores compared to CNN without centroids.

Method	Mean Dice score
Registration	0.724
CNN without centroids	0.720
CNN with centroids	0.736

Table 1.3 The mean Dice score of the test set using different methods. Registration is the multi-atlas registration method; CNN without centroids is the CNN with 2 sets of tri-planar patches; CNN with centroids is the CNN with 2 sets of tri-planar patches and the distances of centroids.

The registration-based method still achieved a Dice score close to that of CNN, indicating the importance of information about the underlying topology of the anatomies. Figure 1.5 details the Dice differences. As expected, the standard deviation in error decreased with the inclusion of centroid distances as a feature.

5.2. Errors

In order to dig into the performance analysis of CNN-based segmentation system, we look at the kind of segmentation errors the method made. We broadly classify the errors into two classes as specified in Section 4.6.

Figure 1.4 illustrates the segmentation of a test case (number 1119) using both CNN and registration. We can see that overall the CNN led to more lumps of misclassifications compared to registration. The misclassifications of registration-based segmentations were generally on the boundaries of the object.

Figure 1.7 illustrates an example of both labelling and boundary errors. The first column is a segmentation of the left lateral ventricle and the second column represents the segmentation of the left cuneus. The first row illustrates the ground truth, the second row illustrates segmentation by the registration-based method, the third row illustrates segmentation by the CNN-based method, the fourth row represents the difference between ground truth and registration, and finally the last row represents difference between ground truth and CNN segmentation. As illustrated in the first column, CNN-based segmentation has misclassification of similar looking voxels for instance misclassfication of left lateral ventricles around insular cortex, see Figure 1.7. In contrast, the errors from registration-based method is on the boundary of the ventricle.

The second column illustrates more severe boundary errors made by both CNN and registration in segmenting the left cuneus. While the segmentation is very specific, the sensitivity is low. In such regions, combining both methods may not improve the Dice score.

A graph to further illustrate the difference between the different segmentation errors is shown in Figure 1.8. Segmentation of left lateral ventricle using CNN has labeling errors since the core dice drops on change of boundary width whereas the

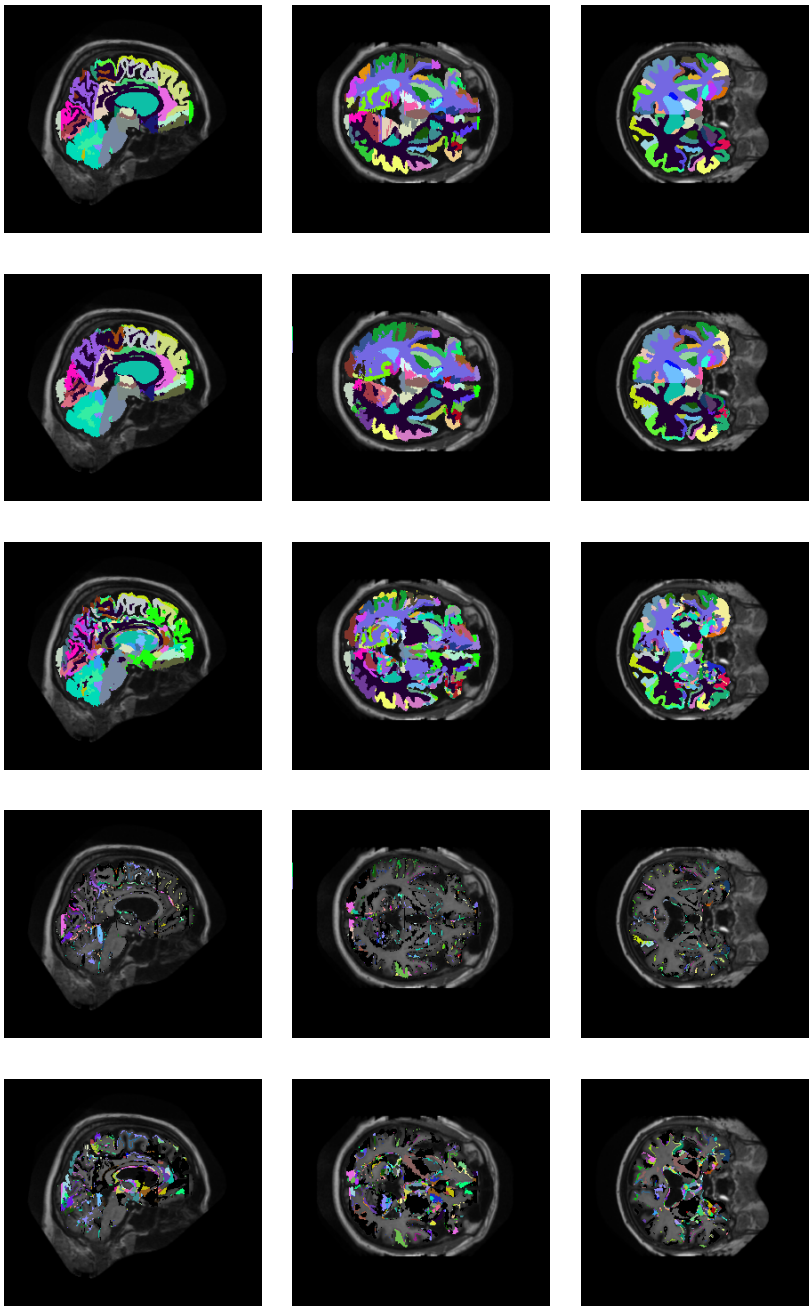

Figure 1.4 Segmentation results from both CNN and registration. The first row illustrates the true labels. Different colors indicate different types of labels. The second and third rows are predictions from registration and CNN, respectively. The last two rows are differences of each method (registration and CNN) from true labels. Difference illustrates that CNN has more misclassifications compared to CNN. The CNN results are based on CNN with centroids.

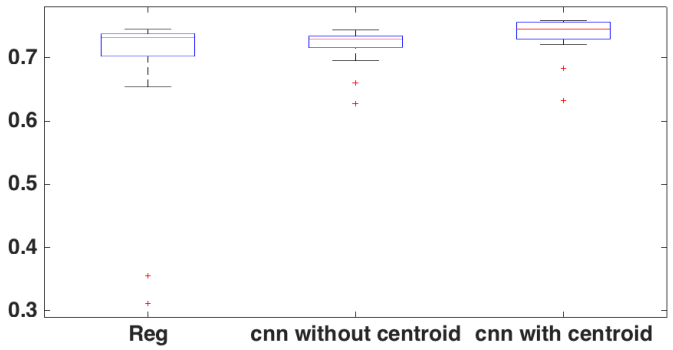

Figure 1.5 The Dice score of different methods. This box plot illustrates the variance in the Dice measure for each method.

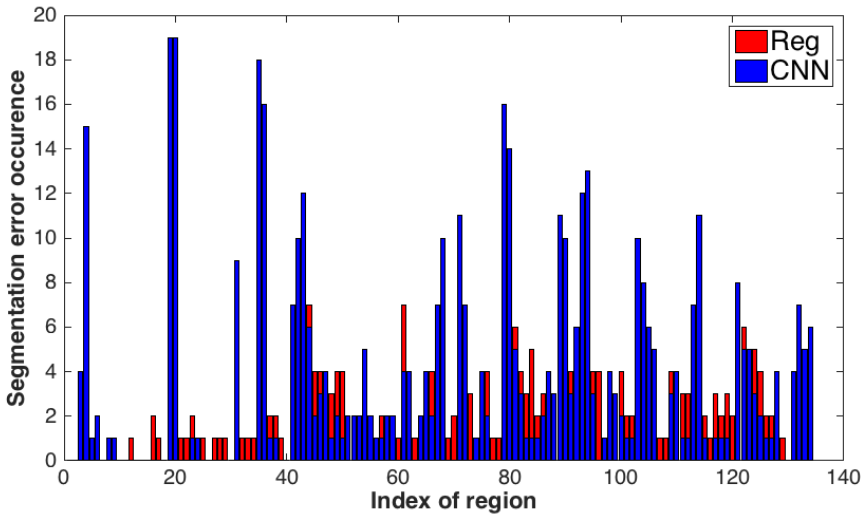

Figure 1.6 The histogram of labelling error occurrence in 134 classes across all the test images. X axis: Index of class. Y axis: Labeling error occurrence. The results are from CNN with centroid. The CNN results are based on CNN with centroids.

registration-based segmentation has 1 core dice upon the change of boundaries. In segmenting left cuneus, both registration and CNN make labeling errors. Labelling errors are errors with lumps of misclassified voxels that are not specific to the anatomy, i.e., false positives. Typically with such errors, the Dice coefficient changes negatively with a change in the boundary width. In contrast, change in boundary width makes a positive change in the Dice score when the errors are of the boundary type. This

is the logic we use to classify the errors. A positive slope in the graph is classified as a boundary error and negative slope is classified as a labelling error. Figure 1.6 shows the histograms of the occurrence of labelling errors for each class across the test data. As expected, the CNN made more labelling errors on average compared to the registration-based method. This is expected since the training dataset is small, and the registration-based methods have stronger topological constraints. On the contrary, the registration-based method made more boundary errors than labelling errors.

6. Discussion

This chapter presented a characterization of errors made by a CNN for automatic segmentation of brain MRI images. Fairly accurate segmentations were obtained. However, the performance was inferior to both a standard registration plus majority voting-based method and other model-based methods presented at the MICCAI 2012 challenge[3]. Note that the best performing methods in the challenge rely on non-linear image registration. With this as motivation, we compared the results of segmentation using a CNN to a registration-based segmentation methodology.

The performance of both the registration and the CNN-based methods could possibly be improved, which might lead to different results in the comparison. For instance, *Moeskops et al.* [21] use convolution kernels at different scales. This results in similar mean Dice scores (0.73). Studies like [3] include 3D patches in the feature stack. Including 3D patches is not trivial because of intensive memory requirements depending on the size of the patches involved. In order to avoid memory intensive representations, the tri-planar approach [24] was proposed.

As illustrated in various examples in the results section, our CNN made more labelling errors than segmentation or boundary errors. This can be caused by limited training size, lack of spatial information, or a combination of both. Unlike images arising in many computer vision problems, medical images are structured, that is, there is a spatial consistency in the location of objects. This gives a particular advantage to methods that use spatial information. The gains from the use of spatial information is apparent when the centroid distances are included as a feature. The centroid distances can be obtained from any popular segmentation algorithm. In such a case, centroid distances in the training must also be computed (as opposed to using the true labels) using the segmentation method so that the network can learn the errors made by the method.

Lack of training data is a limiting factor for the application of deep learning in medical image analysis. For instance, the specific application presented in this chapter aims at obtaining segmentations with 134 classes from just 15 training images.

[3]https://masi.vuse.vanderbilt.edu/workshop2012.

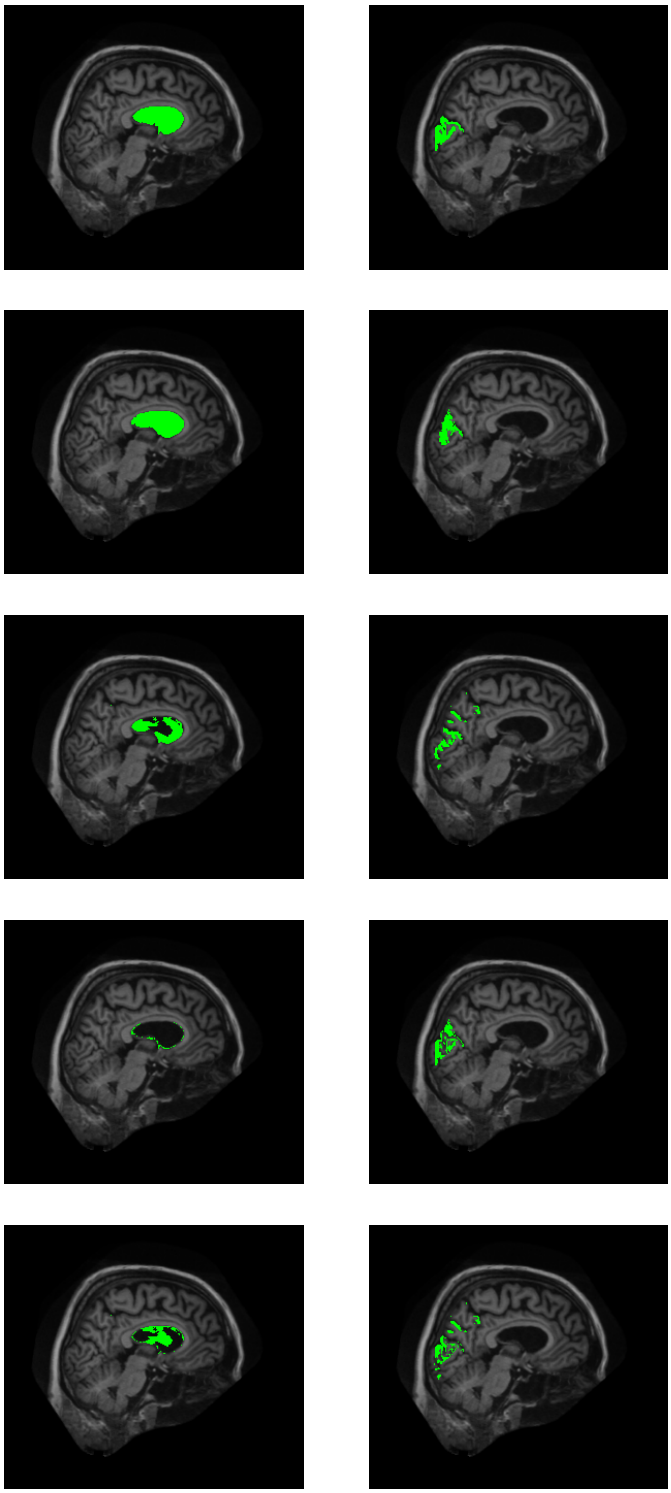

Figure 1.7 Illustration of segmentation errors for a single class. The left column shown segmentation of the left ventricle. CNN (last row, left image) shows clear misclassification (labeling error) of regions in comparison the errors by registration (fourth row, left image) are strictly boundary based. The right column illustrates segmentation of the left cuneus, where both the methods make more severe boundary errors. The second row represents segmentation using registration. Third row represents segmentation using CNN. Fourth row represents difference of registration-based segmentation and the true segmentation, and finally the fifth row represents difference of CNN-based segmentation and the true segmentation.

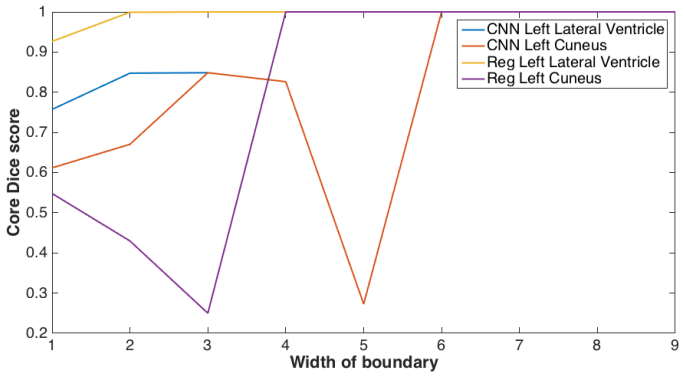

Figure 1.8 The core-Dice score with different boundary width with respect to two classes. x-axis: width of boundary, y-axis: core-Dice score.

Lowering the number of classes with the same dataset yields significantly better Dice scores [21]. When large training sets are available, CNN-based methods generally give better results [13]. In computer vision problems such as the PASCAL challenge [9], CNN-methods perform particularly well due to the presence of large and relatively cheap ground truth data. In the area of medical imaging analysis, however, obtaining ground truth is expensive. In cases where there is an upper limit to the availability of both data and corresponding annotations (what we use as ground truth), using deep learning as a regression tool to a clinical outcome may be more useful. An example application of using CNNs in regression may be found in [20].

In our experiments, the registration-based method made errors complementary to that of the CNN-based method. Thus, combining the two approaches may improve Dice scores. The choice of where to include the geometric information in the network is a research topic in itself. For example, one may add information on the top via centroid distances or have a more intricate way of adding geometric information as a priori inside the network at some level. In addition, one may even consider combining classifications obtained by both methods via an ensemble learning to improve performances for large category classification (large number of classes). With limited validation data, unsupervised/semi-supervised learning as an initialization [15] in conjunction with model-driven approaches such as image registration may improve results in the future.

Even though we only chose one architecture in this chapter, the choices are many, for examples see the special issue [12]. The choices may for example vary in the number of layers or in a fusion of multiple architectures. An increase of the network depth proved effective in the recent ILSVRC challenge where the authors use as many as 152 hidden layers [13]. Once the architecture is fixed, small variations in the settings

may not necessarily yield significantly better results [21].

7. Conclusion

Deep learning methods have already improved performances in medical tasks such as classification of tumors (3/4 classes), detection of micro-bleeds, or even classification of specific structures in histological images, which is a significant step in the field. However, CNN-based methods for segmentation of brain MRI images with a large number of classes may need more training data as used in our study to outperform model-driven approaches. With the advent of big data, connecting healthcare centers may increase the availability of medical data significantly. However, it is likely that obtaining hand annotations for such data may be intractable. In such cases, to leverage the use of data, the field of deep learning in medical imaging may have to move towards combining the benefits of model-driven techniques, unsupervised learning, and semi-supervised learning. In this chapter, we have shown that model-driven and data-driven approaches can make complementary errors. This encourages using techniques, for example ensemble learning, to combine the two approaches.

1. Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.
2. C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.
3. A. Brebisson and G. Montana. Deep neural networks for anatomical brain segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–28, 2015.
4. T. Brosch, L. Tang, Y. Yoo, D. Li, A. Traboulsee, and R. Tam. Deep 3D convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation. *IEEE Transactions on Medical Imaging*, 35(6), 2016.
5. D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Mitosis detection in breast cancer histology images using deep neural networks. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI 2013)*, pages 411–418. Springer, 2013.
6. P. Coupé, J. Manjón, V. Fonov, J. Pruessner, M. Robles, and D. Collins. Patch-based segmentation using expert priors: application to hippocampus and ventricle segmentation. *NeuroImage*, 54(2):940–954, 2011.
7. A. de Brébisson and G. Montana. Deep neural networks for anatomical brain segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2015)*, pages 20–28, 2015.
8. Q. Dou, H. Chen, L. Yu, L. Zhao, J. Qin, D. Wang, V. Mok, L. Shi, and P. A. Heng. Automatic detection of cerebral microbleeds from MRI images via 3D convolutional neural networks. *IEEE Transactions on Medical Imaging*, 35(6), 2016.
9. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.
10. B. Fischl, D. Salat, E. Busa, M. Albert, M. Dieterich, C. Haselgrove, A. van der Kouwe, R. Killiany, D. Kennedy, S. Klaveness, A. Montillo, N. Makris, B. Rosen, and A. Dale. Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. *Neuron*, 33:341–355, 2002.
11. I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. Book in preparation for MIT Press, 2016.
12. H. Greenspan, B. van Ginneken, and R. Summers. Special issue on deep learning in medical imaging. *IEEE Transactions on Medical Imaging*, 35(4), 2016.
13. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, 2016.
14. C. J. Jack, R. Petersen, Y. Xu, P. O'Brien, G. Smith, R. Ivnik, B. Boeve, E. Tangalos, and E. Kokmen. Rates of hippocampal atrophy correlate with change in clinical status in aging and AD. *Neurology*, 55(4):484–489, 2000.
15. M. Kallenberg, K. Petersen, M. Nielsen, A. Ng, P. Diao, C. Igel, C. Vachon, K. Holland, R. R. Winkel, N. Karssemeijer, and M. Lillholm. Unsupervised deep learning applied to breast density segmentation and mammographic risk scoring. *IEEE Transactions on Medical Imaging*, 35(6), 2016.
16. N. Krüger, P. Janssen, S. Kalkan, M. Lappe, A. Leonardis, J. Piater, A. J. Rodriguez-Sanchez, and L. Wiskott. Deep hierarchies in the primate visual cortex: What can we learn for computer vision? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1847–1871, 2013.
17. Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015.
18. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
19. D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner. Open access series of imaging studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. *Journal of Cognitive Neuroscience*, 19(9):1498–1507, 2007.
20. S. Miao, W. J. Zhang, and R. Liao. A cnn regression approach for real-time 2D/3D registration. *IEEE Transactions on Medical Imaging*, 35:1352–1363, 2016.
21. P. Moeskops, M. A. Viergever, A. M. Mendrik, L. S. de Vries, M. J. N. L. Benders, and I. Išgum. Automatic segmentation of MR brain images with a convolutional neural network. *IEEE Transactions on Medical Imaging*, 35(5), 2016.
22. A. Pai, S. Sommer, L. Sørensen, S. Darkner, J. Sporring, and M. Nielsen. Kernel bundle diffeomor-

phic image registration using stationary velocity fields and wendland basis functions. *IEEE Transactions on Medical Imaging*, 35(6), 2016.
23. S. Pereira, A. Pinto, V. Alves, and C. A. Silva. Brain tumor segmentation using convolutional neural network in MRI images. *IEEE Transactions on Medical Imaging*, 35(6), 2016.
24. A. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, , and M. Nielsen. Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 8150 of *LNCS*, pages 246–253. Springer, 2013.
25. H. R. Roth, L. Lu, J. Liu, J. Yao, K. M. Cherry, L. Kim, and R. M. Summers. Improving computer-aided detection using convolutional neural networks and random view aggregation. *IEEE Transactions on Medical Imaging*, 35(6), 2016.
26. H. R. Roth, L. Lu, A. Seff, K. M. Cherry, J. Hoffman, S. Wang, J. Liu, E. Turkbey, and R. M. Summers. A new 2.5 d representation for lymph node detection using random sets of deep convolutional neural network observations. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, LNCS, pages 520–527. Springer, 2014.
27. J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.
28. A. A. Setio, F. Ciompi, G. Litjens, P. Gerke, C. Jacobs, S. van Riel, M. Winkler Wille, M. Naqibullah, C. Sanchez, and B. van Ginneken. Pulmonary nodule detection in CT images: False positive reduction using multi-view convolutional neural networks. *IEEE Transactions on Medical Imaging*, 35(6), 2016.
29. K. Sirinukunwattana, S. Raza, Y. Tsang, D. Snead, I. Cree, and N. Rajpoot. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE Transactions on Medical Imaging*, 35(6), 2016.
30. Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, 2016.
31. M. J. J. P. van Grinsven, B. van Ginneken, C. B. Hoyng, T. Theelen, and C. I. Sánchez. Fast convolutional neural network training using selective data sampling: application to hemorrage detection in color fundus images. *IEEE Transactions on Medical Imaging*, 35(6), 2016.
32. M. Veta, P. J. van Diest, S. M. Willems, H. Wang, A. Madabhushi, A. Cruz-Roa, F. A. González, A. B. L. Larsen, J. S. Vestergaard, A. B. Dahl, D. C. Ciresan, J. Schmidhuber, A. Giusti, L. M. Gambardella, F. B. Tek, T. Walter, C. Wang, S. Kondo, B. J. Matuszewski, F. Precioso, V. Snell, J. Kittler, T. E. de Campos, A. M. Khan, N. M. Rajpoot, E. Arkoumani, M. M. Lacle, M. A. Viergever, and J. P. W. Pluim. Assessment of algorithms for mitosis detection in breast cancer histopathology images. *Medical Image Analysis*, 20(1):237–248, 2015.
33. M. Veta, M. Viergever, J. Pluim, N. Stathonikos, and P. J. van Diest. Grand challenge on mitosis detection,. *Medical Image Computing and Computer-Assisted Intervention (MICCAI 2013)*, 2013.
34. W. von Seelen. Informationsverarbeitung in homogenen Netzen von Neuronenmodellen. *Kybernetik*, 5(4):133–148, 1968.