



Master in Artificial Intelligence (UPC-URV-UB)

Master of Science Thesis

**Segmentation of Brain MRI Structures
with Deep Machine Learning**

Alberto Martínez González

Advisor/s: Laura Igual Muñoz

22/06/2012

Abstract

Several studies on brain Magnetic Resonance Images (MRI) show relations between neuroanatomical abnormalities of brain structures and neurological disorders, such as Attention Deficit Hyperactivity Disorder (ADHD) and Alzheimer. These abnormalities seem to be correlated with the size and shape of these structures, and there is an active field of research trying to find accurate methods for automatic MRI segmentation. In this project, we study the automatic segmentation of structures from the Basal Ganglia and we propose a new methodology based on Stacked Sparse Autoencoders (SSAE). SSAE is a strategy that belongs to the family of Deep Machine Learning and consists on a supervised learning method based on an unsupervisedly pretrained Feed-forward Neural Network. Moreover, we present two approaches based on 2D and 3D features of the images. We compare the results obtained on the different regions of interest with those achieved by other machine learning techniques such as Neural Networks and Support Vector Machines. We observed that in most cases SSAE improves those other methods. We demonstrate that the 3D features do not report better results than the 2D ones as could be thought. Furthermore, we show that SSAE provides state-of-the-art Dice Coefficient results (left, right): Caudate (90.63 ± 1.4 , 90.31 ± 1.7), Putamen (91.03 ± 1.4 , 90.82 ± 1.4), Pallidus (85.11 ± 1.8 , 83.47 ± 2.2), Accumbens (74.26 ± 4.4 , 74.46 ± 4.6).

Acknowledgments

I would like to thank the following people: To my parents for their EXTREME support. To Laura Igual, for her patience and advice. To Alfredo Vellido, for being the person who infused in me the passion for Computational Neuroscience and Deep Learning. To Maribel Gutierrez, for being such an outstanding and supportive worker at the FIB. To my friends, for not letting me forget about the sun. And finally, to Andrew Ng, for changing forever the concept of University, and being such a great professor.

Contents

1	Introduction	1
2	Problem Definition	2
2.1	Basal Ganglia	2
2.2	Anatomy of the Basal Ganglia	3
2.3	Functionality of the Basal Ganglia	4
2.4	Consequences of disorders in Basal Ganglia	5
2.5	Medical Imaging Techniques	5
2.6	Basal Ganglia in MRIs	6
3	State of the Art	7
3.1	Atlas-Based Methods	8
3.2	Machine Learning	9
3.3	Probabilistic Graphical Models	10
3.4	Deformable Models	11
4	Methodology	12
4.1	Neural Networks	13
4.2	SoftMax Model	16
4.3	Sparse Stacked Autoencoders	17
5	Technical Development	20
5.1	Development platforms	20
5.2	Data Acquisition	20
5.3	Data Preprocessing	21
5.4	Feature Extraction	22
5.5	Data Training	24
5.6	Quality assessment of voxel classification	25
5.7	SSAE Parameter Setting	26
5.8	Neural Network Parameter Setting	30
5.9	SVM Parameter Setting	32
6	Experimental Results	33
6.1	Cross-Validation Experiments	33
6.1.1	SoftMax	33
6.1.2	SSAE	34
6.1.3	PCA + SSAE	34
6.1.4	Neural Network	36
6.1.5	SVM	38
6.2	Summary of Cross-Validation Results	39
6.3	Test with Pathological Group	40
7	Discussion	41

1 Introduction

Image segmentation is the branch of computer vision whose purpose is to divide an image into meaningful regions. This process might be understood as a classification problem, where a subset of pixels which share a common visual property within an image or sequence of images are classified together under the same label.

Specifically, medical image segmentation is the task of localizing one or several different biological and anatomical regions of interest (ROIs) in the body of a particular subject, and which are made visible through different imaging techniques such as X-rays, Ultrasounds, Computed Tomography scans (CT) or Magnetic Resonance Images (MRIs).

Segmentation techniques are essential for medical diagnosis, as they allow to quantify changes in volume, shape, structure, intensity, location, composition, or any other measure that might be useful in terms of detecting malfunctioning biological processes, pathological risks or any other medical condition. Segmentation techniques can also be crucial in finding markers for medical conditions such as developmental or neurological disorders, helping to diagnose any health problem, and an endless number of clinical applications.

In this project, we will focus on brain MRI segmentation. Specifically we will define a novel approach for segmenting the main anatomical structures that conform the region called Basal Ganglia (Caudate Nucleus, Putamen, Globus Pallidus and Nucleus Accumbens).

Basal Ganglia is part of the Telencephalon, the higher developed prosencephalic derivative in primates. It is formed by various nuclei with diverse functionality, being control movement or routine learning the main ones. Changes in the morphology of these regions have been associated with a number of psychiatric and neurological disorders, including Attention Deficit Hyperactivity Disorder (ADHD), depression, fetal alcohol syndrome, schizophrenia, Alzheimer's, Parkinson's and Huntington's disease ([5], [13], [59]).

It is important to develop methodologies to help better understanding of these complex disorders, monitor their progression and evaluate response to treatment over time. This requires accurate, reliable and validated methods to segment the ROIs. Although manual segmentation of brain MRI remains the gold standard for region identification, its impractical for its application in large studies due to the time it requires, and anyway its reliability is not complete, since there is an inter-rater bias in the volumes obtained when comparing the segmentations produced by different expert neurologists. This bias could avoid an accurate statistical analysis of data.

For all these reasons, the development of reliable and robust fully automatic segmen-

tation methods is mandatory. Nowadays, there is an active field of research in this area, not only for the bigger and more obvious regions, but particularly trying to improve the segmentation of smaller regions such as the Nucleus Accumbens or the Globus Pallidus that do not yet report good results in the literature ([8], [7], [9], [3] and [49]).

A broad spectrum of methods has been applied for this purpose including methods such as Deformable Models or Atlas-Based Methods, although there is still place for improvement. Machine Learning has also been used with techniques such as Neural Networks, Support Vector Machines or Self-Organizing Maps, but the new trend that exists in the field using deep architectures has not yet reached this branch of research. The purpose of this project will be to make a step in this direction by applying Stacked Sparse Auto-encoders (SSAE) to the brain MRI segmentation problem and comparing its performance with that of other classical Machine Learning models.

SSAE is a supervised learning method based in the Feed-forward Neural Network. There are two main differences with Neural Networks. First, the use of a new regularized cost function that enforces an sparse coding of the input within the hidden layers. Second, a layer by layer unsupervised pretraining phase called Greedy Layer-wise Training, in which the method learns by itself the statistical properties of the training set. The information learned in this pretraining is later used to initialize the supervised training, instead of making a random initialization of the weights. This provides the model with an informed starting point from which to find faster and more reliably the global minimum of the cost function.

The obtained results will show that SSAE improves the other methods, and provides state-of-the-art results.

2 Problem Definition

2.1 Basal Ganglia

Basal Ganglia denotes the set of sub-cortical nuclei which are Telencephalon nuclear derivatives. They are highly interconnected but comprise two functional blocks [5]:

- A ventral system formed by limbic system structures such as Nucleus Accumbens, Amygdala and Substantia Nigra.
- A sensory-motor system, the Striatum, formed mainly by the Caudate, Putamen and Globus Pallidus.

These nuclei do not receive direct information from sensory and motor peripheral systems, but use Substantia Nigra and Pallidus as the input-output points that receive

massive afferents from the cortex and Thalamic Nuclei.

2.2 Anatomy of the Basal Ganglia

The Lentiform Nucleus is formed by the **Globus Pallidus**, which is composed by the external (GPe) and internal Globus Pallidus (GPi) that are separated by a thin layer of white matter, and the **Putamen**.

Rostral to the foremost part of the Putamen, the **Caudate Nucleus** is located parallel to the Lateral Ventricle, and separated from the Diencephalon by the Optostriate Body, and from the Lentiform Nucleus by the Internal Capsule. The Caudate is divided in:

- Head, bigger and more rostral part.
- Body, parallel to the Lateral Ventricle.
- Tail, which ends with a connection with the Amygdala after following the direction of the temporal horn of the ventricle.

The **Nucleus Accumbens** is located where the head of the Caudate and the anterior portion of the Putamen meet, in a way that there is no visible anatomical separation from these structures in MR images, although they have a different functionality. All four regions mentioned until now will be our Regions Of Interest (**ROIS**), that is, the regions that will be studied in this project. They will be studied in both right and left hemispheres, and a 3D rendering of them all can be seen in Figure 1 for better understanding.

There are other regions in the Basal Ganglia, as we said, the Amygdala is joined to the end of the tail of the Caudate, and is basically formed by the basolateral and corticomедial portions.

Finally, the Substantia Nigra lies in the midbrain, dorsal to the cerebral peduncle, and it is divided into a very dense cellular region called Pars Compacta, and a less dense one called Pars Reticulata. The Substantia Nigra Pars Compacta has dopaminergic cells, while the Sustantia Nigra Pars Reticulata has big cells that project outside of the Basal Ganglia, just like the Pallidus.

We do not consider neither the Amygdala nor the Substantia Nigra for segmentation, since for this project we want to focus in the study of the structures belonging to the sensorimotor system. The limbic structures have also their representation in the Nucleus Accumbens, a region that is also interesting for the purpose of this project because of its size.

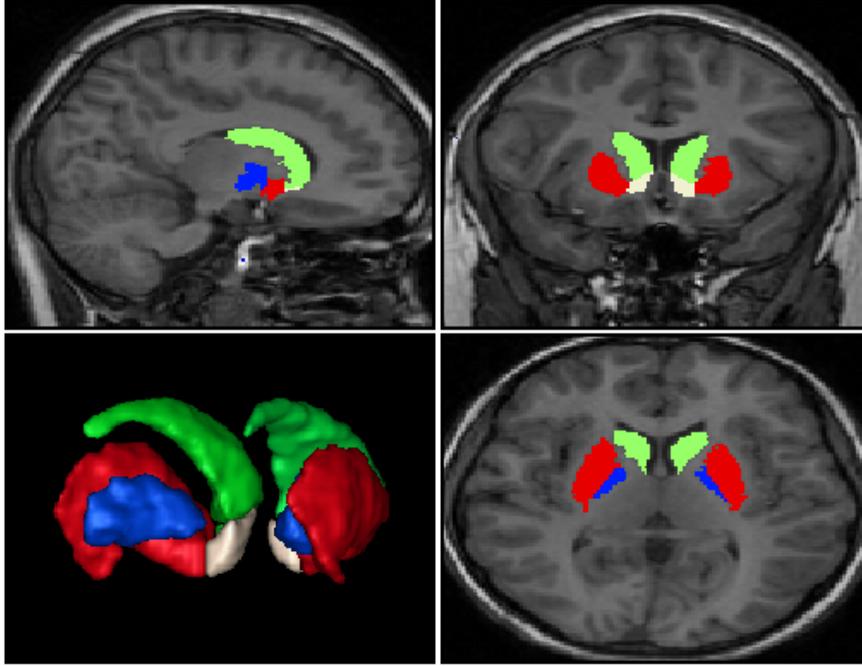


Figure 1: Posterior lateralized view of a 3D rendering of the manual segmentation for the eight ROIs (four left and four right) as they appear in one of the subjects in study, along with an orthographic view of the same regions in their corresponding brain MRI. In green the Caudate Nucleus, in red the Putamen, in blue the Globus Pallidus and in white the Nucleus Accumbens.

2.3 Functionality of the Basal Ganglia

There are in the present time several hypotheses [5] about the function Basal Ganglia have in the Nervous System, and their role in movement development:

- Basal Ganglia do not start movement, but contribute to the automatic execution of sequences of them. To this means, Basal Ganglia should contain necessary algorithms embedded in them.
- They adjust the inhibitory output of GPI, so that the performed movement diminishes or is potentiated.
- They allow or inhibit movement depending on whether they are desired or not.
- They are responsible of automatic execution of previously learned movement sequences.

Anyway, it seems very clear their central role in control and management of body movements when they have been preprocessed in other structures. This is demonstrated by observing their connection with the rest of the brain, that their neurons spike in relation with other movements, and that their lesion has symptomatic motor dysfunction.

There are several studies suggesting that Basal Ganglia are a main part in learning of tasks or sequential behavior.

Some neurons in the Striatum and Substantia Nigra spike when receiving a reward or instructive signals, just a few seconds before the movement is done, so it can be deduced that they are implied in reinforcement learning. This even helps neurons in the Substantia Nigra to be able to predict when a behavioral event will take place.

2.4 Consequences of disorders in Basal Ganglia

A subtle imbalance in neurotransmitters in Basal Ganglia may produce several disorders, such as slow motions and stiffness, or very fast and out of control motions, whether the lesion was produced in the Pallidus or in the Subthalamic nucleus. It can also lead to inability to learn complex behaviors.

Moreover, Basal Ganglia has been found responsible of depression, Obsessive-Compulsive Disorder, Attention Deficit Hyperactivity Disorder, Schizophrenia, Parkinson, Huntington disease, or Tourette Syndrome ([5], [13], [59]).

2.5 Medical Imaging Techniques

In order to allow physicians to diagnose properly all neurological conditions stated above, several different visualization techniques have been developed, as in-vivo observation is too complex in some cases, and even impossible in the vast majority.

Techniques such as CT or X-Rays were the standard for years, but the intensive energy levels at which they operate and the kind of radiation in which they are based might be harmful in high doses, and they are actually not safe even with small exposure times for population segments such as children and pregnant women. On the other hand, technologies such as ultrasounds are also non invasive and are not aggressive as they operate at low energy levels, but the resolution of the images obtained is very poor, and also the physical structure of several biological regions impose extra limitations to the results that might be obtained.

Fortunately, by early 1990s, the technology to implement MRIs was mature enough to start replacing such old school techniques. As explained by [58], the basic functioning is as follows: A strong magnetic field is created around the subject, and then brief pulses of radiofrequency radiation are emitted from a transmission coil around the area being studied. This energy is absorbed by the biological tissues while emitted, and when the pulse ends it is retransmitted at different rates by the different tissues (magnetic relaxation). This re-emission is detected, and so the different tissues are distinguished and the final image is constructed representing those tissues correspondingly with different

gray levels.

Depending on which physical emission decay is measured, different kinds of images will be obtained (T1 weighted, T2 weighted and T*2 weighted are the most typical structural MRIs), and if you resonate at different frequencies other kinds of molecules can be detected, and so different kinds of MRIs are produced (diffusion MRIs, functional MRIs, etc.)

After this process is done, it is time to analyze the data obtained. For structural MRIs, the interest relies in analyzing the different anatomical structures that are visible and conform the gray matter (such as the Cortex or the Basal Ganglia), the white matter (myelinated axons of the neurons interconnecting the different parts of the brain), and other less critical tissues.

2.6 Basal Ganglia in MRIs

Even having MRIs with the highest possible resolution, the segmentation of the structures in the sub-cortical regions present many difficulties.

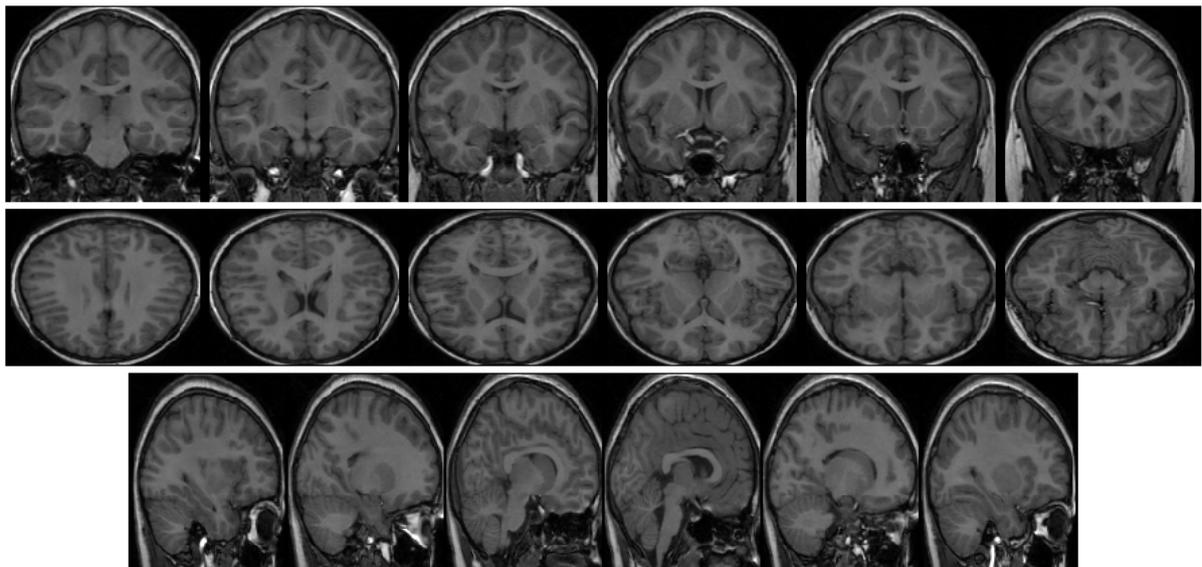


Figure 2: Intra-subject equidistant slices in the main three axes, showing the area in which the Basal Ganglia is located.

The relative low contrast of the voxels within a specific structure and those in the surrounding structures, as can be appreciated in Figure 2, and some times an undefined boundary shared with them, as is the case between the Caudate and Nucleus Accumbens, may cause traditional automatic segmentation approaches to fail.

But not only automatic methods may fail. Although, ideally, we would desire human experts to make perfect manual segmentations, there are also many inconsistencies when

comparing the manual segmentations made by different experts on the same MRI set. When comparing the overlap between two manual segmentations of the same MRIs, it can be shown that there is an inter-rater correlation [23] of only around the 80% for the Basal Ganglia structures, as will be indicated with greater detail in section 5.2. Because we are using manual segmentations as the ground truth, this inter-rater correlation will be our practical accuracy limit.

Although manual segmentation of the different ROIs by an expert neurologist is the ideal setting for analyzing the biological information hidden in the MRIs, this process is highly time consuming, so efficient automatic processes would be desirable to get faster results and let neurologists spend their expertise in other less mechanical tasks.

Furthermore, the brain is a biological tissue, and as such it suffers from a noteworthy gravitational pull, which affects it in a non uniform way due to its overall structure and composition. Not only that, but the area next to the skull suffers a different deformation than the area in the central core depending on the orientation of the head. While the MRIs are produced, the subjects may have the head slightly tilted in infinite number of directions. Also, as the images are obtained slide by slide and they take some time, small unintentional movements may alter not only the perceived signals, but the resolution between the different layers, which could produce several problematic artifacts in the final MRI. All these problems may be solved by a process called registration, which basically consists in producing affine or non affine transformations of the matrix containing the information of the pixel intensities (as they are really three-dimensional information, they receive the name of **voxels**) of each MRI, so that the final result is a new set of MRIs aligned with each other and in the same reference coordinates and with similar shapes and sizes. Extended information about MRI registration can be found in [38].

3 State of the Art

Different approaches have been adopted in order to obtain a fully-automatic segmentation of sub-cortical structures. We can differentiate four types of algorithms:

- Atlas-Based methods
- Machine Learning techniques
- Probabilistic Graphical Models
- Deformable Models

In this section we review all these categories, along with the main papers that conform the state-of-the-art in the field of automatic brain MRI segmentation.

3.1 Atlas-Based Methods

Atlas-based models rely on comparing the images under study with a pre-computed anatomical *atlas* of the brain.

In the context of this work and the referenced ones, an atlas is defined as the pairing of a structural MRI and a corresponding manual segmentation. A critical underlying assumption of these models is that it is possible to find a deformation that aligns the atlas with the target image using *label propagation*. Label propagation is the process of applying to the manual segmentations the same transformation applied to the original MRI in a way that manual segmentations line up with the target objects of interest.

Atlas-based algorithms were originally based on a single mean atlas as in [33], but as research advanced, they evolved into Multi-Atlas strategies. For example, in [25] authors segmented 67 brain structures using a non-rigid registration approach and *label fusion* of 29 atlases. Label fusion refers to the process of calculating the majority vote for the voxels in the manual segmentations of the atlases to obtain a final target segmentation. The same research group in [26] used this time 48 MRIs and a similar approach changing just the registration technique used.

In [2], although authors used 275 subjects, for each new segmentation they selected only the more similar atlases to the MRI they want to segment (around 15 gives the best results to them, although they reported results with a different quantities), and they made a label fusion that was used as the final prediction. In [3] authors fixed the number of atlases to 20 and used the same approach to segment a bigger set of sub-cortical structures.

CAUSE07 [61] was a challenge dedicated exclusively to the automatic segmentation of the Caudate Nucleus. The authors in [63], who were the winners of the CAUSE07, selected the most similar Atlases from the set of 37 Atlases provided for the challenge, and only registered a subset of them, selecting the amount automatically depending in some heuristics that describe if further improvement is expected in using extra Atlases. After this subset had been chosen and registered, they started segmentation by describing local similarity measures among the atlases and the target MRI, propagate only the parts of the original labels that best fit the target, and finally fused them in a final prediction.

In [36], authors also used an atlas-based approach. Concretely, they started with an automatic segmentation on 37 volumetric structures over 24 MRIs, which was achieved by the use of an affine registration and intensity normalization, then cropped the data to only the bounding box around the ROIs in each hemisphere, and then processed the resulting data with the Freesurfer software library [22] to get an automatic segmentation. Parallel to this they managed to construct a Segmentation Confidence Map (SCM), which is a probability map stating how accurately it is expected to behave the original

automatic segmentation method in each particular voxel of any new target. They weight the prediction made by Freesurfer with this map, obtaining their final segmentation.

3.2 Machine Learning

Machine Learning techniques have been long used in the realms of MRI analysis almost from the creation of this medical imaging modality. Neural Networks [51], K-Nearest Neighbors [15], Hopfield Neural Networks [43] (which later lead to Boltzmann Machines, the precursors of modern Deep Learning Systems), Kohonen’s Self-Organizing Maps [66] or Support Vector Machines [1], are among the methods used not only for segmentation of brain anatomical structures, but also for tumors, injuries, and also for automatic diagnosis.

Specifically for segmentation of the Basal Ganglia, authors in [45] used the classical Artificial Neural Networks (ANN) to segment the Caudate and Putamen over 30 manually segmented MRIs. They used a different model for each ROI, segmenting them independently from each other, but using the same three-layer configuration with 42 inputs that correspond to a spherical 3D region around each voxel, the frequency with which the region was found in the search space of the training set, and the 3D coordinates.

In a more complex approach, authors in [46] also used ANN, but they preprocessed the data in an interesting way. Their approach was to calculate several Geometric Moment Invariants (GMI), which are features that are theoretically not sensitive to particular deformations such as rotation, translation and scaling in their basic form, (but that could also be defined to be invariant to more complex transformations as described in [34]). First, they produced a registration of all the images in the training set. Then, for each voxel they calculated eleven GMIs with eight different scales each, and then for each scale they formed a vector adding in it the following information: the eleven GMIs, the intensities of nineteen voxels forming a particular constellation of six voxels along each orthogonal direction plus the central voxel, and the coordinates. They fed each of the eight feature sets to a different ANN which were fully trained. Finally, a new ANN was trained with the eight predictions of the previous phase, again the nineteen intensities and the coordinates, and they fully trained that final network once more. Finally, a cleaning process was initiated to the prediction, eliminating outlier predictions and filling the improperly predicted gaps.

With a simpler approach than the previous study, in [52] authors report better results than any of the previous studies, that is, a Dice Coefficient around the 90% for the Caudate and Putamen. They compared the use of ANN with SVM, as compared with more classical approaches such as single-atlas segmentation and probability based segmentation. What they did was to first produce a general non-linear registration to the

25 MRIs they used to segment Caudate and Putamen among other structures, and then for each ROI, they produced in a voxel-wise manner a vector of 25 features including 3 spherical coordinates, 9 intensities along the largest gradient, 12 intensities along the orthogonal coordinates and the probability of being part of that particular ROI. They used those vectors to train the methods and compare the results.

3.3 Probabilistic Graphical Models

Probabilistic Graphical Models are the models that use the strength of Probability Theory to study the relationships of different variables which are represented as a graph. The most used architecture within the Probabilistic Graphical Model framework is the Markov Random Fields, which is the common name to refer those directed graphical models that might contain cycles. They can be found in studies such as [64], [11] and [62]. We can also find in use Markov dependence tree [68], [69] or Hidden Markov models [30]

The classical Expectation Maximization (EM) [19] algorithm has also been used in STAPLE [65], which is a method that considers a collection of segmentations and computes a probabilistic estimate of the true segmentation, estimating simultaneously the expected accuracy.

In [55], authors described a Gaussian Mixture Model (GMM), which is a generative model that tries to estimate the label map associated with the test image via Maximum-A-Posteriori estimation. In [31], they compared two versions of this method using different parameters versus a Multi-atlas based method [25] and STAPLE [65] resulting in the GMM outperforming the others.

Graph-cuts [37] have also been used in several studies. This method considers all the voxels as vertices connected to each other by edges whose weights must be calculated, and then the basic principle is to cut the vertices until a minimum graph is obtained. In [20], authors proposed a 3D brain MRI segmentation technique based on this technique, but with automatic initialization of the voxel probabilities using Gaussian Mixture Model (GMM) and a 3D instead of 2D segmentation. In [32], also participants of the CAUSE07 challenge, authors presented CaudateCut, a method specifically designed to segment the Caudate Nucleus adapting Graph Cut by defining new energy function to exploit the intensity and geometry information, adding supervised energies based on contextual brain structures, and also reinforcing boundary detection using a multi-scale edgeness measure.

In [10], a comparison of Single and Multi-Atlas versus STAPLE and MRF was performed on MRIs depicting mouse brains, resulting on the Multi-Atlas and STAPLE approaches beating their counterparts.

3.4 Deformable Models

Although several approaches have been used with more classical deformable models such as Active Contour Models in [42] or [57], the deformable model approaches are nowadays normally based on two basic models developed by Tim Cootes, Active Shape Models (ASM) [18] and Active Appearance Models (AAM) [17]. An extensive review of the state of the art may be found at [27].

The ASM model is trained from manually drawn contours (surfaces in 3D) in training images, which are delineated in terms of **landmarks**. Landmarks are a fixed number of points each of which is supposed to represent the exact same characteristic across all the images in the training set (for instance the leftmost corner of an eye or a mouth when representing faces). The ASM model finds the main variations of each individual landmark in the training data first registering all images to a common frame by scaling, rotating and translating them, and then using Principal Component Analysis (PCA [50]). This enables the model to learn the typical shape and typical variability of the ROI being segmented. When this process is finished, an iterative search process starts in which an initial contour is deformed until the best local texture match is found for each of the landmarks, but this deformation is only produced in ways which are consistent with the variability of shapes found in the training set. This approach was taken in several studies such as [54] or [60] to segment the brain ventricles and cardiac MRIs.

On the other hand, the AAM model uses the same process as ASM to develop a model of the shape, but after that, it also generates in a similar fashion a model of the appearance, representing the intensities associated to the shape. PCA is used to find the mean shape and main variations of the training data to the mean shape. Both the Shape and Appearance Model are combined again with PCA obtaining the final model. Finally, a search process is done in which iteratively, the difference between a new image and one synthesized by the model is minimized. This is exactly the approach taken in [8], but after the AAM produced a preliminary segmentation, a final step of boundary reclassification was performed based on voxel intensity. Other studies also followed this approach with further improvements that usually included further statistical analysis, such as in [4], [40] and [49].

Nevertheless, in [7] a comparison of four methods was done: two Shape and Appearance models, AAM [8] and BAM [49], a Multi-Atlas approach [2] and a Probabilistic Graphical EMS model based on [62]. This study reported that the Multi-Atlas approach managed a much better accuracy, followed closely by the Probabilistic Graphical Approach, but another later study by the same author [9], managed to improve the Deformable Model approach to report state-of-the-art results.

4 Methodology

The purpose of this section is to review Machine Learning techniques which can be applied to the problem of segmenting structures in the Basal Ganglia from MRI scans. Nevertheless, there is a gap missing in the state-of-the-art, as no Deep architectures seem to have been fully explored yet for the problem of brain MRI segmentation. We try to make a step towards this direction in the framework of this project, and we propose the use of a method based on Neural Networks called Sparse Stacked Autoencoders.

Deep Neural Networks are networks which have multiple hidden layers. This fact allows to compactly represent a larger set of functions than the set allowed by shallower structures.

Although it has been found that if we have two layers of neurons [24], or RBF units [12], we have an universal approximator, the number of units you need to represent a function could be exponential in the input size. To show the existence of this problem, [24] came up with a set of functions that can be represented in a very compact way with k layers, but require exponential size with $k-1$ layers.

Experimental results in [12] suggested that backpropagation starting from random initialization gets stuck in apparent local minima or plateaus, and that as more layers are added, it becomes more difficult to obtain solutions that perform better than the solutions obtained for networks with 1 or 2 hidden layers. This may be due to a problem related with the diffusion of gradients, because when using backpropagation, the gradients that are propagated to previous layers diminish in magnitude as the depth of the network increases, so in the end the first layers are unable to learn when the depth increases. And that not to mention that the deeper the architecture, the higher the non-convexity of the function to optimize, which implies that the risk to end up in a local minimum is much higher.

Deep Belief Networks [29] was the first successful approach to solve these problems. The limitations of Backpropagation learning can be overcome by using a new approach to Multilayer Neural Networks, that is to apply a Greedy Layer-wise unsupervised pre-training where the network learns by itself the statistics underlying the training set, and then using those learned parameters to "fine-tune" the network with a more classical backpropagation approach.

Although several models have been developed since the first Deep Network model was proposed, we will focus on **Stacked Sparse Autoencoders** ([47], [12] and [21]), which can be understood as an extra twist on the classical Feedforward Neural Network.

4.1 Neural Networks

Logistic Regression

The Logistic Regression Model is a supervised learning model which represents the basic unit conforming the standard Neural Network model, when the sigmoid function we will describe below is used as the activation function. Under this circumstance, Logistic Regression is also known by the name of Perceptron or Neuron.

Let (X, Y) define the training set of m feature vectors with $X = \{x^{(1)}, \dots, x^{(m)}\}$ and $Y = \{y^{(1)}, \dots, y^{(m)}\}$, where $x^{(i)}$ is the feature vector representing the data with the label $y^{(i)} \in \{0, 1\}$. Given an input feature vector x which we want to classify, the model will calculate $h_{W,b}(x)$ defined as

$$h_{W,b}(x) = f\left(\sum_{j=1}^n W_j x_j + b\right) = f(W^T x + b), \quad (1)$$

where n is the number of features in x , W is a vector of n weights, b is the bias term (what would be the weight of the intercept term), and where $f(z)$ is the sigmoid function

$$f(z) = \frac{1}{1 + \exp(-z)}. \quad (2)$$

The purpose of calculating $h_{W,b}(x)$ is that the obtained value approximates the correct binary label that should be assigned to x . This is achieved by a previous learning process in which the correct values of W and b should be learned from the training set (X, Y) where each example $x^{(i)}$ has assigned a correct label $y^{(i)}$, and which should be statistically representative of all the possible values expected for values of x not in the training set.

The learning process is achieved feeding the training set to the function $h_{W,b}(x)$, and then using some cost minimization algorithm, where the cost function we want to minimize (so that we can later compare this model with SoftMax regression) is

$$J(W, b) = \frac{1}{m} \sum_{i=1}^m [-y^{(i)} \log(h_{W,b}(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_{W,b}(x^{(i)}))]. \quad (3)$$

The gradient of W is a vector of the same length as W , where the j^{th} element is defined as follows:

$$\nabla_{W_j} J(W, b) = \frac{\partial J(W, b)}{\partial W_j} = \frac{1}{m} \sum_{i=1}^m (h_{W,b}(x^{(i)}) - y^{(i)}) x_j^{(i)}, \quad \forall j = 0, 1, \dots, n, \quad (4)$$

and the gradient of b is the following value

$$\nabla_b J(W, b) = \frac{\partial J(W, b)}{\partial b} = \frac{1}{m} \sum_{i=1}^m (h_{W,b}(x^{(i)}) - y^{(i)}) . \quad (5)$$

These values are used to minimize iteratively the prediction error thanks to some optimization algorithm such as backpropagation that we will describe later in this section.

Neural Networks

A Neural Network (NN) is a model in which many neurons are interconnected in such a way that there are no loops (otherwise it would be called a Recursive Neural Network), and neurons are organized into layers. Neurons in the same layer are fully connected to the neurons in the previous layer, except for the first layer, because this layer is not formed by neurons but by the vector $x^{(i)}$ that will be the input to the model.

The way in which this model learns to predict the label $y^{(i)}$ associated to the input $x^{(i)}$ is by calculating the function $h_{W,b}(x) = a^{(n)} = f(z^{(n)})$, where n is the number of layers, b is a matrix formed by $n - 1$ vectors storing the bias terms for the s neurons in each layer, and W is a vector of $n - 1$ matrices each of which is formed by s vectors, each one representing the weight of one of the neurons in one of the layers.

In order to calculate $a^{(n)}$ and $z^{(n)}$, we need to calculate, for each layer l starting with $l = 1$ and knowing that $a^{(1)} = x$,

$$z^{(l+1)} = W^{(l)} a^{(l)} + b^{(l)} , \quad (6)$$

$$a^{(l+1)} = f(z^{(l+1)}) . \quad (7)$$

Calculating the value of $h_{W,b}(x)$ is called a feedforward pass. To train the NN, the first thing to do is to initialize the weights W and the bias terms b . This should be done using random values near zero. Otherwise, all the neurons could end up firing the same activations and not converging to the solution.

Once we have produced a feedforward pass, we need to calculate the cost function. We define the cost function of a single example as

$$J(W, b; x, y) = \frac{1}{2} \|y - h_{W,b}(x)\|^2 , \quad (8)$$

that is, half of the squared distance from the prediction to the ground truth. For the whole training set we will use

$$J(W, b) = \frac{1}{m} \sum_{i=1}^m J(W, b; x^{(i)}, y^{(i)}) + \frac{\lambda}{2} \sum_{l=1}^{n-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ij}^{(l)})^2 , \quad (9)$$

where m is the number of examples and λ is the weight decay parameter. This parameter λ helps to prevent overfitting by penalizing the cost when the weights grow too much.

Now that we have a function that measures the cost of all predictions with a particular set of weights, we need a way to update those weights so that, in next iteration, the cost will be reduced and the training may converge to a minimum, hopefully the global one. This update value is:

$$\begin{aligned}\Delta W &= \frac{\partial}{\partial W^{(l)}} J(W, b) = \left[\frac{1}{m} \sum_{i=1}^m \nabla_{W^{(l)}} J(W, b; x^{(i)}, y^{(i)}) \right] + \lambda W^{(l)} \\ &= \left[\frac{1}{m} \sum_{i=1}^m \frac{\partial}{\partial W^{(l)}} J(W, b; x^{(i)}, y^{(i)}) \right] + \lambda W^{(l)},\end{aligned}\tag{10}$$

$$\begin{aligned}\Delta b &= \frac{\partial}{\partial b^{(l)}} J(W, b) = \frac{1}{m} \sum_{i=1}^m \nabla_{b^{(l)}} J(W, b; x^{(i)}, y^{(i)}) \\ &= \frac{1}{m} \sum_{i=1}^m \frac{\partial}{\partial b^{(l)}} J(W, b; x^{(i)}, y^{(i)}).\end{aligned}\tag{11}$$

So the first step is to calculate $\nabla_{W^{(l)}} J(W, b; x, y)$ and $\nabla_{b^{(l)}} J(W, b; x, y)$ for each example independently, and this will be done with the backpropagation algorithm.

Backpropagation Algorithm

Backpropagation is the algorithm that allows to calculate the factors by which each weight should be updated in order to minimize the error produced between the prediction and the ground truth given a set of weights W and a bias term b . It proceeds as follows:

- Perform a feedforward pass, that is, calculate the final activations $h_{W,b}(x) = a^{(n)}$, where n is the number of layers, and denoting that $a^{(n)}$ are the activations of the last layer. This will give us a vector of predictions achieved by the actual weights θ . Moreover, store all the intermediate $z^{(l)}$ and $a^{(l)}$ for each layer l for later use.
- For each final activation $a_i^{(n)}$ with $i = 1, \dots, l$, calculate the penalization term

$$\delta_i^{(n)} = \frac{\partial}{\partial z^{(n)}} \frac{1}{2} \|y - h_{W,b}(x)\|^2 = -(y_i - a_i^{(n)}) \cdot f'(z_i^{(n)}).\tag{12}$$

This factor indicates how different the prediction of the model is from the ground truth.

- Propagate the penalization term to the previous layers by calculating for each node i in layer l except the first layer (the input does not need to be corrected)

$$\delta_i^{(l)} = ((W_i^{(l)})^T \delta_i^{(l+1)}) \cdot f'(z_i^{(l)}) . \quad (13)$$

- Finally, compute the partial derivatives

$$\nabla_{W^{(l)}} J(W, b; x, y) = \delta^{(l+1)} (a^{(l)})^T , \quad (14)$$

$$\nabla_{b^{(l)}} J(W, b; x, y) = \delta^{(l+1)} . \quad (15)$$

Now, we can calculate ΔW and Δb with the formulas in the previous section. These partial derivatives should now be used to properly update the old weights with some optimization technique such as gradient descent, conjugate gradient or L-BFGS algorithm [44]. As L-BFGS algorithm has been shown to give good results [41], will be the option chosen for this project.

4.2 SoftMax Model

SoftMax model generalizes the Logistic Regression model for multi-class classification. Multi-class classification is useful when the different classes are mutually exclusive and label y has k possible values, $y^{(i)} \in 1, 2, \dots, k$, and not only two possible outcomes.

For convenience, we will use θ to denote the weights and the bias term $\theta = (b, W^T)^T$. For this to be possible, we need to add a new feature, $x_0^{(i)}$, to each example $x^{(i)}$, where $x_0^{(i)} = 1$ always. For the purpose of this project, θ will denote specifically the weights for the SoftMax model and not others.

The purpose of this method is to estimate the probability of an input $x^{(i)}$ being part of each one of the k classes $y^{(i)}$ could take, that is, calculate $p(y^{(i)} = j | x) \forall j = 1, \dots, k$.

To achieve that, the hypothesis function SoftMax model uses the following one:

$$h_{\theta}(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1 | x^{(i)}; \theta) \\ p(y^{(i)} = 2 | x^{(i)}; \theta) \\ \vdots \\ p(y^{(i)} = k | x^{(i)}; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix} . \quad (16)$$

As our cost function we will use the following:

$$J(\theta) = \frac{1}{m} \left[\sum_{i=1}^m \sum_{j=1}^k (y^{(i)} == j) \log \left(\frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^k e^{\theta_l^T x^{(i)}}} \right) \right] + \frac{\lambda}{2} \sum_{i=1}^k \sum_{j=1}^n \theta_{ij}^2 , \quad (17)$$

where the expression, $y^{(i)} == j$, will be 1 in case $y^{(i)} = j$ and 0 otherwise. Note that if $k = 2$ this would be the same cost function as the one we defined for Logistic Regression, except for the last weight decay term, because this term is:

$$h(x) = \frac{1}{e^{\vec{0}^T x^{(i)}} + e^{(\theta_2 - \theta_1)^T x^{(i)}}} \begin{bmatrix} e^{\vec{0}^T x^{(i)}} \\ e^{(\theta_2 - \theta_1)^T x^{(i)}} \end{bmatrix} = \begin{bmatrix} \frac{1}{1 + e^{(\theta_2 - \theta_1)^T x^{(i)}}} \\ 1 - \frac{1}{1 + e^{(\theta_2 - \theta_1)^T x^{(i)}}} \end{bmatrix}. \quad (18)$$

So for the parameter set $\theta' = (\theta_2 - \theta_1)$, we would be calculating the same exact probability as with the Logistic Regression model (Equation 1).

It can be shown that with the regularization term now the Hessian is invertible as long as $\lambda > 0$, and so the cost function $J(\theta)$ is now strictly convex. Thus, a unique solution is guaranteed, and any minimization algorithm is assured to converge to a global minimum.

To apply those minimization algorithms we will need to compute:

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m} \sum_{i=1}^m [x^{(i)} ((y^{(i)} == j) - p(y^{(i)} = j | x^{(i)}; \theta))] + \lambda \theta_j. \quad (19)$$

4.3 Sparse Stacked Autoencoders

Autoencoders

An Autoencoder Neural Network, as described in [28], is an unsupervised learning algorithm that tries to approximate the target values $y^{(i)}$ to be equal to the inputs $x^{(i)}$, that is, an approximation to the identity function. This is done by an architecture identical to a NN with only one hidden layer that has to learn the function

$$h_{i,W,b}(x) = a_i^{(l)} = f\left(\sum_{j=1}^n W_{ij}^{(l-1)} a_j^{(l-1)} + b_i^{(l-1)}\right) = \hat{x}_i \approx x_i, \quad (20)$$

where $W_{ij}^{(l)}$ is the weight for the neuron i in the hidden layer, $b_i^{(l)}$ is the bias term for the neuron i in the hidden layer (the bias term corresponds again to the intercept term), n is the number of units in the hidden layer, and a_j is the activation of neuron j in the hidden layer ($l=2$). The value of a_j for layer l is calculated as follows:

$$a_j^{(l)} = f\left(\sum_{k=1}^m W_{jk}^{(l-1)} x_k + b_j^{(l-1)}\right), \quad (21)$$

where m is the number of input units, $W_{ik}^{(l-1)}$ and $b_i^{(l-1)}$ are the weight and bias for the input unit k , and where $f(z)$ is the activation function, which again will be the sigmoid function.

Trivial as the identity function may seem, the interesting point of this approximation is that it allows to discover easily the structure in the data by placing constraints to the hidden layer.

The only constraint that can be applied to Autoencoders as they are defined is altering the number of hidden units. This constraint would have an effect similar to that of other feature selection techniques, by learning a low-dimensional representation of the original feature set in case the number of hidden units was smaller than the number of inputs.

Sparse Autoencoders

Sparse Autoencoders ([12] and [53]) represent a step further than normal Autoencoders, as they have a sparsity constraint imposed on the hidden units. The intuition behind this approach is that we intend that the number of active units (units with an activation value equal to one) for every prediction is low, forcing the Autoencoder to achieve a more efficient image compression of every input x .

First we redefine the activation term of layer l , $a_j^{(l)}$, as $a_j^{(l)}(x)$ to denote the activation of this hidden unit when the network is given a specific input x . Further, we define

$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^m [a_j^{(l)}(x^i)], \quad (22)$$

as the average activation of hidden unit j over the training set. Our task is to ensure that $\hat{\rho}_j \approx \rho$, where ρ is a sparsity parameter that should be set to a value close to zero to ensure that the Autoencoder will learn to code the input x in an sparse manner, using only a small number of neurons.

$$\sum_{j=1}^{s2} \text{KL}(\rho_j \parallel \hat{\rho}_j) = \sum_{j=1}^{s2} \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j}, \quad (23)$$

where KL is the Kullback-Leibler divergence [39] between a Bernoulli random variable with mean ρ and a Bernoulli random variable with mean $\hat{\rho}_j$.

So, the cost function is now

$$J_{sparse}(W, b) = J(W, b) + \beta \sum_{j=1}^{s2} \text{KL}(\rho_j \parallel \hat{\rho}_j), \quad (24)$$

where β is a variable set to control the weight of the sparsity penalty, and $J(W, b)$ is the same as for the Neural Network model.

In order to train this model we also use the backpropagation algorithm as described in the NNs, except that the derivatives for the second layer should now be calculated as

$$\delta_i^{(2)} = (((W^{(2)})^T \delta^{(3)}) + \beta(-\frac{\rho}{\hat{\rho}_j} + \frac{1-\rho}{1-\hat{\rho}_j})) \cdot f'(z_i^{(2)}), \quad (25)$$

and then, the same process of optimization should be followed.

Sparse Stacked Autoencoders

Intuitively, Sparse Stacked Autoencoders (SSA [47], [12] and [21]) is a model with layers just as a NN, but in which each layer is a Sparse Autoencoder, except for the last layer that is a SoftMax model.

The way to train this is first train each layer individually in an unsupervised manner (with the input being equal to the output), and use the weights obtained to transform the input data of that model into activation values, and use those activations as the input for the new layer. This process should be repeated for as many layers as the model has, until the SoftMax model is trained in a supervised way using the labels assigned to each input x .

After this "preprocessing" step is achieved, the network should be "fine-tuned", using the calculated weights as initial weights. The network is trained as a whole with the backpropagation approach described for the NN, except for some considerations.

First, a forward pass should be computed, where the cost function should be calculated with the formula described for the SoftMax model. Then, for backpropagation purposes, the gradient of the SoftMax model should be computed independently in a first stage with the gradient, $\nabla_{\theta_j} J(\theta)$, we described earlier in Equation 19; note that now $x^{(i)}$ as expressed in Equation 19 corresponds to the activation of the last Sparse Autoencoder. Finally, the last Sparse Autoencoder are treated as the last layer from the point of view of the backpropagation algorithm. $\delta^{(n)}$ should be computed as

$$\delta_i^{(n)} = (\theta_i^T (y^{(i)} - h_{\theta}(x^{(i)})) \cdot f'(z_i^{(l)}), \quad (26)$$

where each y should be expressed as a binary vector with length k . Each element of this vector expresses if it belongs or not to that particular class. The rest of the derivatives are calculated as

$$\delta_i^{(l)} = ((W_i^{(l)})^T \delta_i^{(l+1)}) \cdot f'(z_i^{(l)}), \quad (27)$$

and the rest of the Backpropagation process is completed just as was described for the NN model. The L-BFGS algorithm is then used to update the weights of the entire network.

5 Technical Development

5.1 Development platforms

The main platform used in this project was Matlab, and all software was highly optimized by vectorizing all operations. The optimized internal routines for matrix transformations that Matlab uses became very handy also in the process of feature construction, as neighbors were found in batches by simple matrix transformations.

SPM8 [6] toolbox was used in order to manipulate the original MRIs as, for instance, to transform them into the 3D matrices used by all the routines created for feature construction.

Neural Network and Stacked Sparse Autoencoder models were entirely implemented from scratch for this project following the implementation instructions explained in [48]. For the L-BFGS algorithm the code by [56] was used. For computing the step direction this implementation of L-BFGS algorithm calculates limited-memory BFGS updates with Shanno-Phua scaling.

For SVM, the library LibSVM [14] was used in a one-against-all multi-label classification approach.

All programs were executed in a machine equipped with an I7 processor and 6GB of RAM memory.

5.2 Data Acquisition

The data we used for this study was extracted from the public dataset that can be accessed at [35], which was manually segmented by the authors of [23]. Specifically, it comprises 103 MRIs from four diagnostic groups: healthy control, Bipolar Disorder without Psychosis, Bipolar Disorder with Psychosis and Schizophrenic Spectrum. The subjects are children and adolescents aged from 6 to 17 years, both female and male. Exclusion criteria included: presence of major sensorimotor handicaps; full-scale IQ<70; presence of documented learning disabilities; history of claustrophobia, autism, anorexia, bulimia nerviosa, alcohol or drug dependence or abuse; active medical or neurological disease; presence of metal fragments or implants; history of electroconvulsive therapy; and current pregnancy or lactation.

The images were acquired at the McLean Hospital Brain Imaging Center on a 1.5 Tesla General Electric Signal Scanner. Structural acquisitions included a conventional T1-weighted sagittal scout series, a proton density/T2-weighted interleaved double-echo axial series, and a three-dimensional inversion recovery-prepped spoiled glass coronal series.

About manual delineation, first the images were positionally normalized into the standard orientation of the Talairach coordinate space, and then a bias field correction as described in [67] was applied. After that, MRIs were segmented with a semiautomated intensity contour algorithm for external border definition and signal intensity histogram distributions for delineation of gray-white borders. This kind of algorithms needs the rater to indicate which area to segment, and allows him to have the last decision about the final result, so that is the reason the final segmentation is considered manual.

The Inter-rater Intra-class Correlation Coefficients (ICCs), which measure how good is the manual segmentation of one rater with respect to another different rater, were the ROIs we are studying received the following ICC (left, right): Caudate (0.95, 0.93), Putamen (0.80, 0.78), Globus Pallidus (0.84, 0.77) and Nucleus Accumbens (0.79, 0.57). These measures serve us to establish an approximate upper bound of human performance to compare our results with.

For more information about the acquisition technical details or the segmentation criteria, we refer to [23].

5.3 Data Preprocessing

It is a common practice to apply some sort of registration technique to the MRIs in order to make them as similar as possible to a common MRI template. However, the considered data was already positionally normalized to be aligned and centered in the Basal Ganglia, thus, no further registration was performed in a hope that our method would be strong enough to detect by itself differences in shape along different individuals.

First thing was done to the data was a format change, in which images were transformed from the original NIFTI format to a 3D matrix with each of the axial, coronal and transverse planes represented in each one of the matrix dimensions. Then all images with a resolution different than 256x256x128 voxels were discarded, leaving only 81 valid MRIs. Those were then divided into a training set of 22 healthy control MRIs and a test set with the remaining 59 diagnostic MRIs. The purpose of making this division is to treat the diagnostic group as outliers, and so have the method trained independently of any possible variance specific to any particular disease. Also, it will allow us to make a more informed decision about how does the method generalize to diseased subjects in general, being certain that the method did not overfit to a particular disease.

A sample slice of each of the 22 healthy controls can be seen in Figure 3. In it, it can be appreciated the strong inter-subject variability in shape and size of all regions due to, among other reasons, the differences in age and development stage of the different subjects.

Although the original atlases were manually segmented for structures in all the brain,

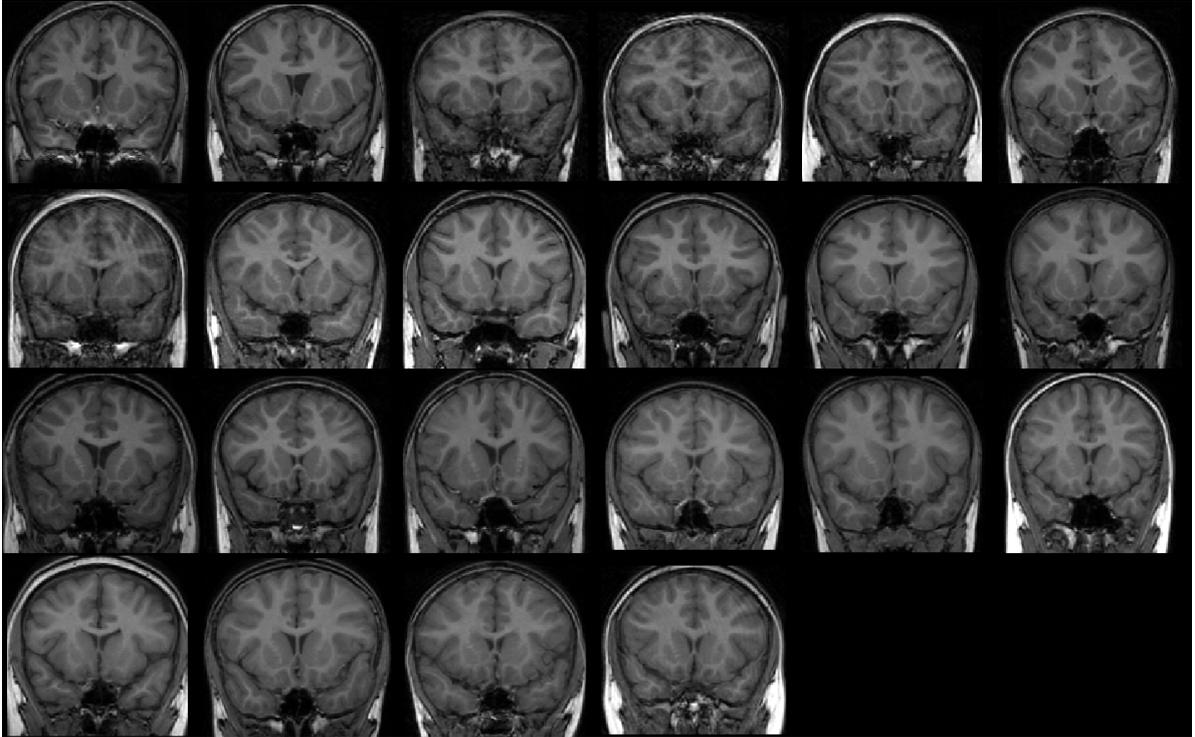


Figure 3: One coronal slice of each of the 22 healthy control MRIs that we use as a training set, where the inter-subject variability in shape and size may be observed.

we are only interested for this study in 4 regions of the Basal Ganglia, all in both hemispheres. That give us a total of 8 ROIs to study. All the remaining tags are reclassified into a common tag called "No ROI", and is studied as an extra region.

5.4 Feature Extraction

A probability map is calculated where the probability of each individual pixel being part of each one of the 8 ROIs is calculated by computing the frequency, that is, counting the number of times each pixel shows up in each structure in the manual segmentations, and dividing that amount by the number of MRIs in the training set, as illustrated in Figure 4.

In order to reduce the number of examples that should be fed to the prediction algorithm, all feature sets are pruned. The pruning criterion is done in order to keep only those examples with probability of belonging to any of the ROIs larger than zero. We compute which pixels have zero probability of being part of any of the ROIs, and use that information as a probability mask to study only the probable points. Figure 5 illustrates this idea.

All methods in the study are applied in a voxel-wise manner, that is, segmentation is achieved one voxel at a time. For this approach to be successful, each training example

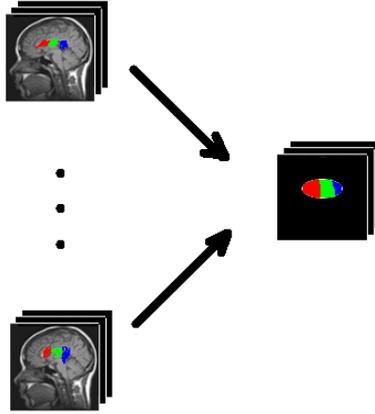


Figure 4: Process of obtaining the probability map

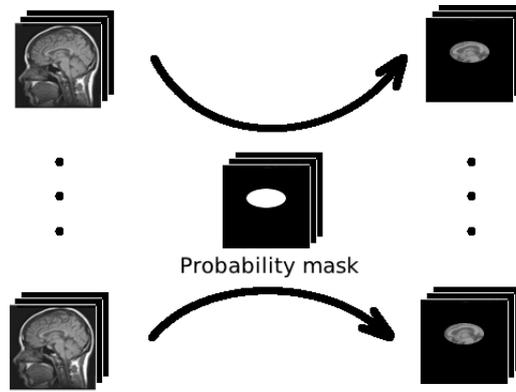


Figure 5: Process of using the probability mask in order to prune the MRIs voxels

contains information about the intensity of the voxel to be segmented, Cartesian coordinates of that voxel, probability of that voxel to belong to each of the eight ROIs under study and intensity of the voxels in the closest neighborhood. To decide how big the considered neighborhood should be, two approaches were taken. A 2D approach where voxels were chosen only in one of the three planes, and a 3D approach, in which cubes around the voxel were chosen with different scales.

In the 2D case, patches of 3x3, 5x5, 7x7, 9x9 and 11x11 pixels were chosen along each of the three main planes, giving a total of fifteen different feature sets. In the 3D case, cubes of size 3x3x3, 5x5x5 and 7x7x7 were used, producing three extra feature sets. After the features were formed, points were selected only from within the corresponding probability mask area to conform the training sets. In Figure 6, we show the process of feature construction.

All variables were normalized to mean zero and unit standard deviation. After all feature sets were constructed, they were all passed through a Principal Component Analysis (PCA) with a 99% of total variance preserved.

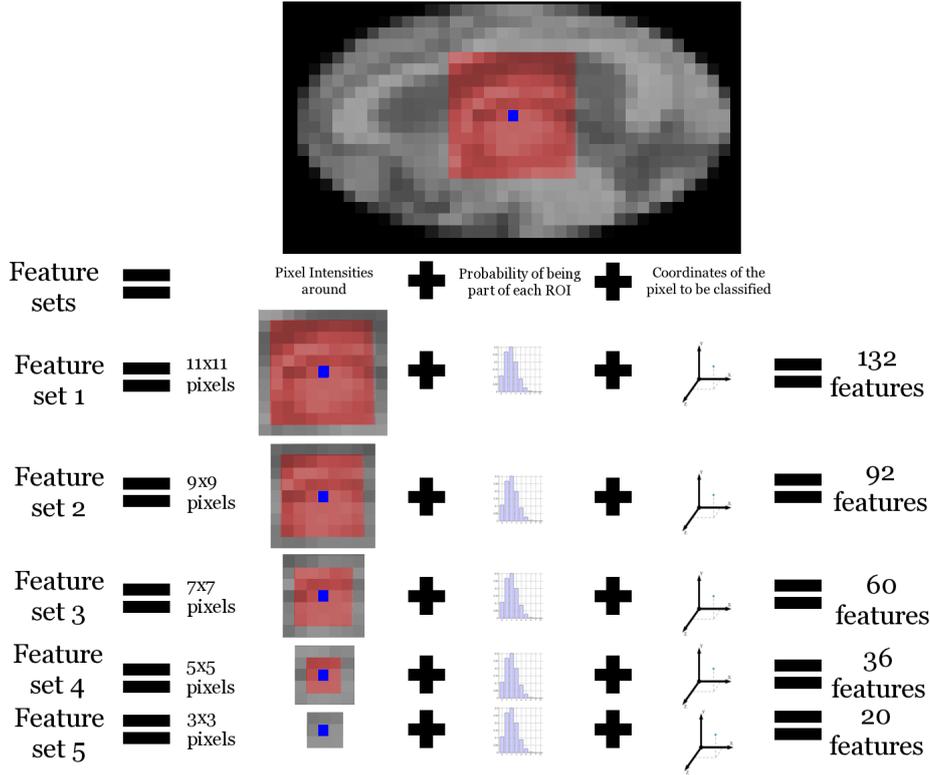


Figure 6: Feature construction process for one of the 2D planes. The rest of the features are constructed in a similar manner.

5.5 Data Training

Our purpose in this study is to determine the efficiency of the Sparse Stacked Autoencoder (SSAE) approach with respect to other Machine Learning approaches. In order to do that, the same feature sets, described in Figure 7, were used to train eight different approaches. Those are:

1. A SoftMax method only for the 2D features.
2. The SSAE model only for the 2D features.
3. SSAE for the 2D features with PCA preprocessing.
4. SSAE for the 3D features with PCA preprocessing.
5. Neural Network with one hidden layer for the 3D features with PCA preprocessing.
6. Neural Network with two hidden layers for the 3D features with PCA preprocessing.
7. Support Vector Machine (SVM) model with Radial Basis Kernel only for a 20% of the 3D features with PCA preprocessing.

8. SSAE for only for a 20% of the 3D features with PCA preprocessing.

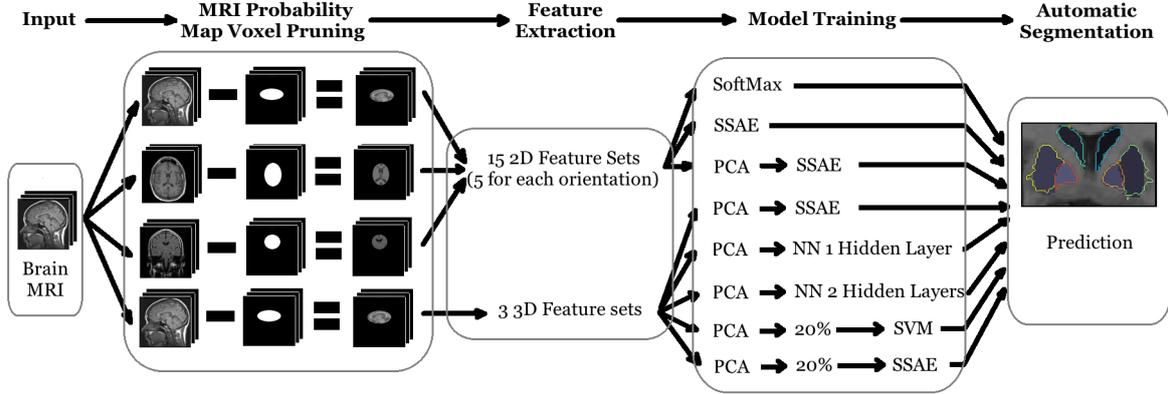


Figure 7: Depicts the whole process of feature construction

In this way, SSAE can be trained both for the 2D and the 3D feature sets, and its performance can be compared with that reported by all the other methods without having to train them all with all the feature sets. Moreover, as SVM demonstrated to be several orders of magnitude slower than SSAE, using only a subset of the feature sets is enough to assess the performance of both methods in our data. Furthermore, using several different feature extraction protocols makes the final results more consistent, as decisions are not based only in one particular set of features. Finally, it allows us to explore several strategies for getting a better final model.

5.6 Quality assessment of voxel classification

Since the data used for brain segmentation is volumetric, when studying one particular ROI the number of voxels it occupies is by far smaller than the total amount of voxels. This skewness in the nature of the data may lead to erroneous results. If the measure to assess the quality of the segmentation is a normal accuracy in terms of true positives compared to the total number of voxels, a method could simply predict no voxel belong to it and the accuracy would be almost 100%. To avoid this phenomenon, other quality measures are needed.

In this study, except for the parameter setting phase in which the normal accuracy is used, we apply a popular metric called **Overlap Ratio** (OR) defined as:

$$\text{OR} = \frac{|h_{\theta}(x) \cap y|}{|h_{\theta}(x) \cup y|}, \quad (28)$$

where $h_{\theta}(x)$ denotes the hypothesis of the classifier and y denotes the ground truth labels associated to the images x . This measure can also be expressed in terms of true positives

(tp, the number of voxels correctly predicted), false positives (fp, the number of voxels wrongly classified as part of the ROI) and false negatives (fn, number of voxels wrongly classified outside the ROI) as follows:

$$\text{OR} = \frac{\text{tp}}{\text{tp} + \text{fp} + \text{fn}} . \quad (29)$$

Another of the most extended measurement for quality assessment in the literature on brain segmentation is the **Dice Coefficient**. It is calculated as follows:

$$\text{Dice} = \frac{2|h_{\theta}(x) \cap y|}{|h_{\theta}(x)| + |y|} , \quad (30)$$

where, again, $h_{\theta}(x)$ denotes the hypothesis of the classifier and y denotes the ground truth labels. It can be shown that the Dice Coefficient corresponds exactly to the measure known in Statistics and Machine Learning as F_1 Score, defined in terms of Precision and Recall as:

$$F_1 \text{ Score} = \frac{2 \text{ Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} , \quad (31)$$

where Precision is defined as

$$\text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}} , \quad (32)$$

and Recall is defined as

$$\text{Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}} . \quad (33)$$

All these measurements range from 0 to 1, depending on the number of labels that do or do not overlap, and are proportional to the number of labels that do.

Since Dice coefficient can be expressed in terms of OR, using the following formula:

$$\text{Dice} = \frac{2 \text{ OR}}{1 + \text{OR}} , \quad (34)$$

most of our results are expressed with the OR coefficient, and only the final results are expressed also with the Dice Coefficient to facilitate comparison with other future works and previous works in the literature.

5.7 SSAE Parameter Setting

First, SSAE model was trained with different sparsity penalties β and sparsity parameters ρ , just to see the effect such variables would have in the final overlap on a fixed validation

set. In this way, the weight decay parameter λ and the number of hidden layers l were temporarily settled to a constant value, so that we could visualize better the effect of changing the other two. These values were settled to $\lambda = 0.01$, and $n = 100$ hidden units for each of the $l = 2$ hidden layers of Sparse Autoencoders used. After the other parameters were optimized, an optimization process was also applied to the previously fixed parameters, until a good final tuned up set of parameters was obtained.

One of the fundamental assumption of machine learning, in general, is that the fixed training set you have collated will have enough information about the data domain that you want to represent, so it will make correct predictions when confronting unseen new examples. Also, there is empirical evidence [16] that the more data you feed the model with, the better your results will get. Using these two assumptions, we can make a third one. One could feed to the model only on a subset of your training set that represents well enough the variability of the input space, and after testing different sets of model parameters with it. The best set will only get better results when applied with the full training set. Of course, we cannot be totally certain that the parameters chosen will be the optimal ones, because we cannot ensure that the results would get better at the same rate for every parameter configuration when adding more training examples. Although that is true, at least we know that with this approach we will choose a parameter set that defines well the statistics behind the kind of data we are seeking to define.

For these trainings to be fast, choose randomly a sample with the 5% of the training examples, doing all the process for only one fixed training set. As we are using 22 MRIs in total for our training set, and 2 of them are used for validation purposes, a 5% sample of the remaining 20 MRIs contains as many examples as those we would get from a single MRI, so we can expect it to contain a representative enough part of the variability of the data to give us a set of parameters that will generalize well. Let us explain next the settings for 2D and 3D features.

2D Features

For the 2D features, the approach taken was to use only one of the orientations for the parameter tuning, in particular the transverse one. In this way, only one third of the 2D features was used, but as all three orientations have the same five patch sizes, we believe we will obtain anyway a good parameter set for the fifteen 2D feature sets.

First, different β s and ρ s were tried in order to classify the outputs of the SSAE either into one of the eight ROIs, or as not part of any ROI. We obtained the results in Figure 8, where the values in the Z coordinate correspond to the accuracy measured as the number of voxels correctly classified divided by the total number of voxels, and the other two coordinates that represent ρ and β , and are expressed in a logarithmic scale from zero to

one.

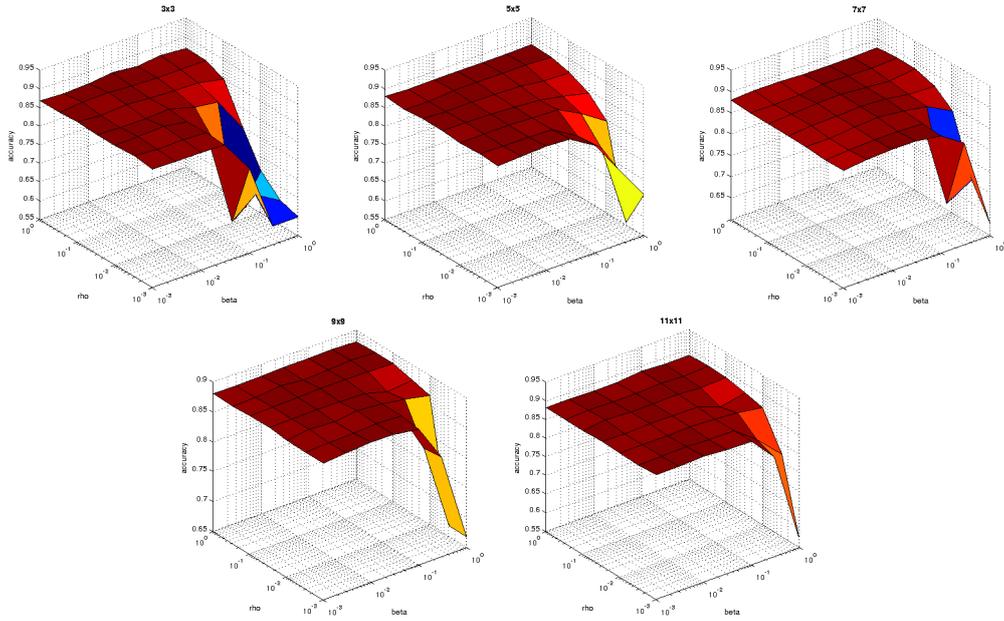


Figure 8: Parameter setting for SSAE with different β s and ρ s with all the 2D sets with the transverse orientation.

Note that for all of the feature sets the same variables have almost the same effect, that is, it is better to use a big ρ and a small β , but basically all feature sets could use the same values for these parameters, because there is a big plateau for most of the configurations that is equivalent in all feature sets.

Now, we train with another set of parameters, this time trying to find a good weight decay parameter λ . The results obtained are shown in Figure 9. Once again there seem to be a set of values that work equally well for all the feature sets, and even two of them seemed to be unaffected by changes in it.

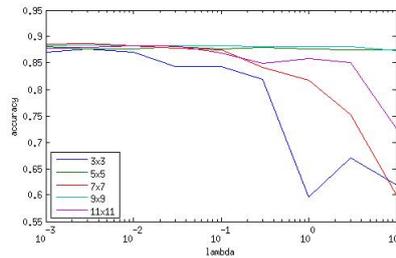


Figure 9: Parameter setting for SSAE with different λ s with all the 2D sets.

Finally, different configurations were used for the number of hidden units of the individual Sparse Autoencoder layers conforming the SSAE model. The results obtained were diverse, so in order to make them comprehensive, in Figure 10, we show the mean of the accuracies obtained by all the different architectures.

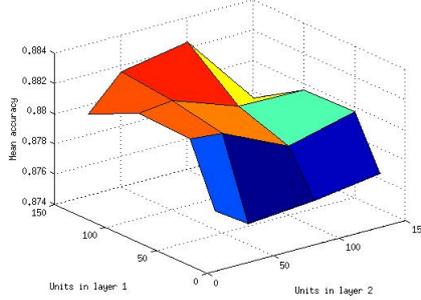


Figure 10: Parameter setting for SSAE with different hidden layer size, where the mean for all the 2D sets is shown.

We see that the general trend is that to use more neurons in the first layer than in the second one is best, but using too little or too many is equally bad for the final accuracy. Although we see that the best configuration uses 150 hidden units for the first layer, we actually use less units in both layers due to computational constraints.

After all the trainings were concluded, the configuration chosen was the one used for the 5x5 feature set, as it should work fine for all of the feature sets due to the results obtained above. This configuration is: 100 units for the first hidden layer and 50 for the second (a configuration that seems to be a good one although not optimal) $\lambda = 3e^{-3}$, $\beta = 1e^{-2}$ and $\rho = 1e^{-1}$, and we are using that configuration for all 2D feature sets.

3D Features

For the three 3D feature sets we found the parameters that independently worked best for each of them. As there are six parameters to tune up, we group them in couples and plot the results of training them in a grid way. The results for the different ρ s and β s can be found in Figure 11, those for different λ s related to different maximum number of iterations can be seen in Figure 12, and for different number of units in the Sparse Autoencoder layers in Figure 13.

We see that in the case of the 3D features with PCA, in the 7x7x7 case it works best without sparsity term, although the other two feature sets use a $\rho = 1e^{-1}$, with $\beta = 1e^{-3}$ for the 3x3x3 case and $\beta = 3e^{-1}$ for the other.

About the number of iterations, it seems that the 3x3x3 case needs only 200, but the other two feature sets need even less, and with just 100 it is enough to get the optimal results. About the weight decay term, the 3x3x3 feature set seems to work best without it, but the other two feature sets use it, with a $\lambda = 1e^{-3}$.

Finally, the number of units used is 100 in the first layer and 50 in the second for the 3x3x3 case, 150 in each for the 5x5x5 case and 100 for each layer in the remaining case.

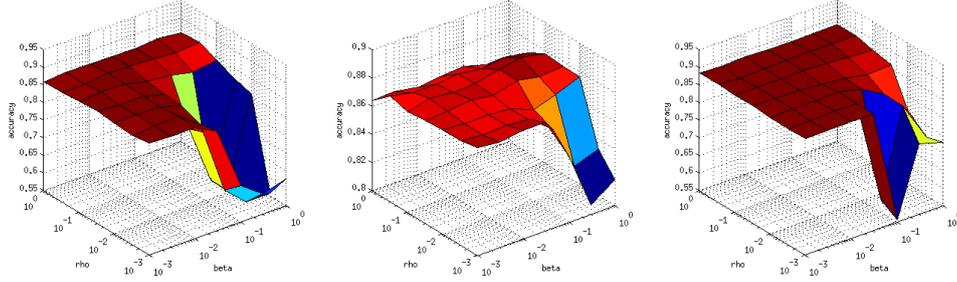


Figure 11: Parameter setting for the SSAE model with different ρ s and β s with the 3D feature sets. From left to right you can see the 3x3x3, 5x5x5 and 7x7x7 case.

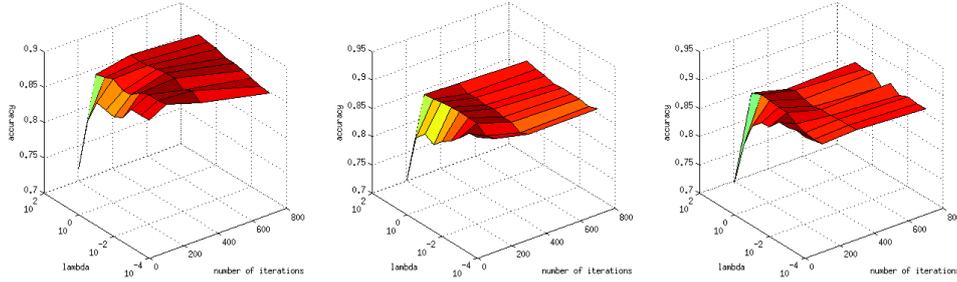


Figure 12: Parameter setting for the SSAE model with different λ s and number of iterations with the 3D feature sets. From left to right you can see the 3x3x3, 5x5x5 and 7x7x7 case.

5.8 Neural Network Parameter Setting

Using the same approach as for the SSAE model, only a 5% of the data was trained with the Neural Network models for the parameter setup, but only the 3D feature sets were trained, as the purpose of training this method was just to have a way to rate the performance of SSAE. In particular, two different NN architectures were used, one with only one hidden layer, and another one with two hidden layers.

One Hidden Layer

As we are using a NN architecture that uses L-BFGS as the optimization algorithm, no learning rate is needed to be specified as it would be needed in the case of gradient descent. Still, a weight decay term λ has been added, and so we study which λ s work best for our data. The comparison of the one hidden layer NN with different λ s and different number of hidden units can be found in Figure 14. In it we can see several effects: First, for small λ s the weight decay term does not make enough effect, and so the model tends to fall into local minima. Second, too much weight decay is also undesirable, since lower than optimal values are obtained. And third, once you have enough units in the hidden

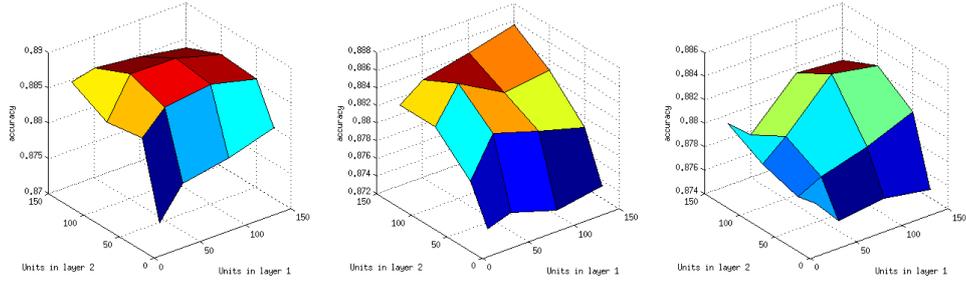


Figure 13: Parameter setting for the SSAE model with different number of units with the 3D feature sets. From left to right you can see the 3x3x3, 5x5x5 and 7x7x7 case.

layer, adding more does not make the prediction any better, as it may produce overfitting.

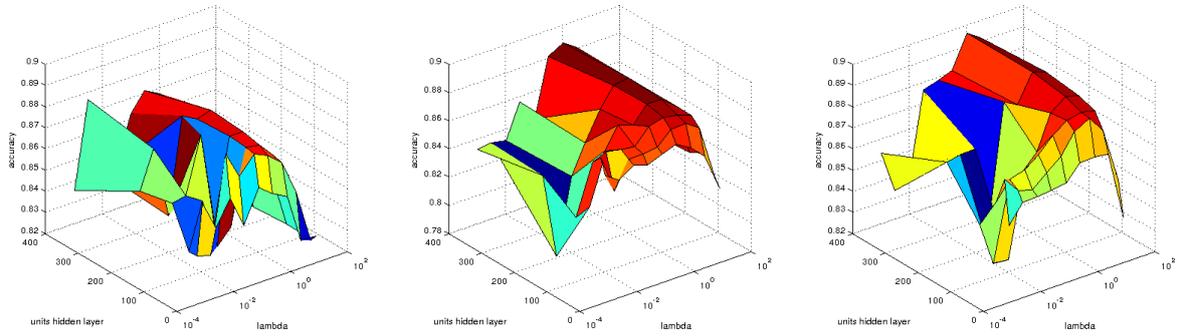


Figure 14: Parameter setting for the 2 hidden layer NN with different λ s and number of units with the 3D feature sets. From left to right you can see the 3x3x3, 5x5x5 and 7x7x7 case.

After the parameter setting process was done, both the 5x5x5 and 7x7x7 case were found to work best using 200 hidden units and $\lambda = 1$. On the other hand, the 3x3x3 case found best to use $\lambda = 1e^{-2}$ and only 100 hidden units.

Two Hidden Layers

In this case, another variable was needed to be trained, the number of units in the second hidden layer. In order to see also the effect of the number of iterations in the final accuracy, first we trained a comparison with different λ s and number of iterations that can be found in Figure 15. In it we can appreciate that 800 iterations are way too many and produce overfitting that leads to poorer results, and 400 are enough for all feature sets. About the λ s, $\lambda = 1$ works best for the 3x3x3 case and $\lambda = 3$ for the other two feature sets.

To discover the optimal number of hidden units, several configurations were tried, as can be seen in Figure 16. In the case of the 3x3x3 features the best configuration was 50

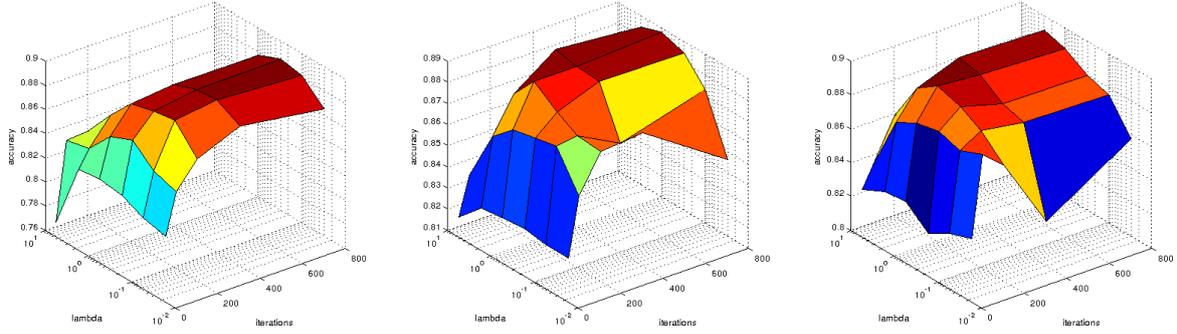


Figure 15: Parameter setting for the 2 hidden layer NN with different λ s and number of iterations with the 3D feature sets. From left to right you can see the 3x3x3, 5x5x5 and 7x7x7 case.

units for each layer, but in the other two, the best was to use 100 in the first layer, and then only 25 in the second.

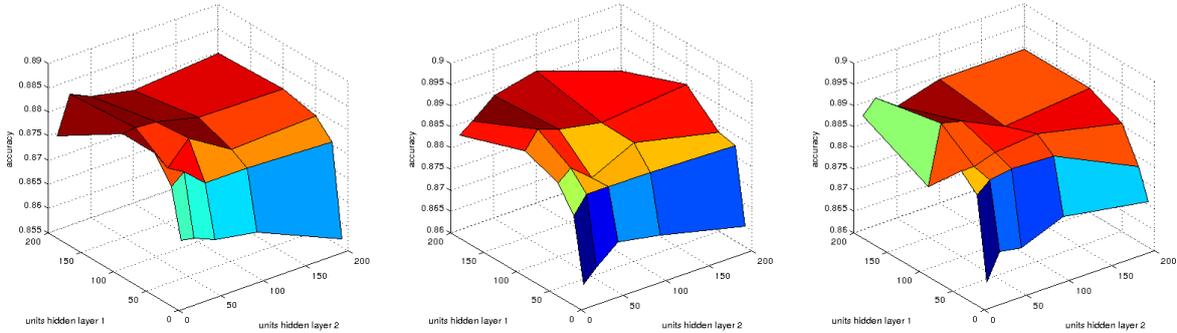


Figure 16: Parameter setting for the 2 hidden layer NN with different number of units with the 3D feature sets. From left to right you can see the 3x3x3, 5x5x5 and 7x7x7 case.

After the optimal configuration was decided, the two NNs were trained for the full training set as will be seen in the sections below.

5.9 SVM Parameter Setting

Using again the same approach as before, we train an SVM model for a 5% of the training data in order to find a good set of parameters with which to train later a cross-validation step. The only difference with the previous step is that this time we train the SVM with the 3D feature set, as this is the feature set we will use for later comparison.

As was stated earlier, we use a Radial Basis Kernel for our SVM. This makes the model a little bit more complex in the sense that there are two variables to tune up, but nevertheless it is still simpler than SSAE in that sense. This time the parameters for each feature set needed to be different due to the big differences they produced in each of the feature sets. The results of training for different C s and γ s can be seen in Figure 17,

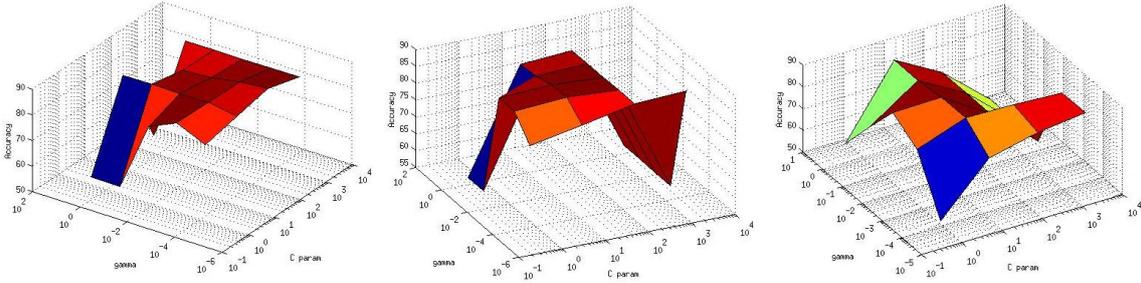


Figure 17: Parameter setting for SVM with different γ s and C s with the 3D sets. From left to right you can see the 3x3x3, 5x5x5 and 7x7x7 case.

The chosen values for the 3x3x3 case were $C = 2^3$ and $\gamma = 2^5$, for the 5x5x5 case we chose $C = 2^3$ and $\gamma = 2^{-7}$, and for the 7x7x7 case the chosen parameters were $C = 2^3$ and $\gamma = 2^{-11}$.

Once all the parameters are chosen, the cross-validation process is started.

6 Experimental Results

In order to report the following results, a cross-validation approach is used. Only 22 of the MRIs, corresponding to the healthy controls, are used as the training set, and the rest of the 59 MRIs conforming the diagnostic group are used as the final test set.

6.1 Cross-Validation Experiments

For the cross-validation of the training set, 11 partitions are produced to it with 2 MRIs in each. Then each one of the partitions is used as validation set, using the remaining 20 MRIs for training purposes.

6.1.1 SoftMax

As this is the fastest of all the methods, the variable tuning is done directly in the last phase, by really cross-validating the method with three different values for the parameter λ , with $\lambda = [0, 1e^{-4}, 1e^{-3}]$. The purpose of calculating the SoftMax method is to have a way to assess the performance leap produced by the use of the hidden Sparse Autoencoder layers in the SSAE model with respect to using the last SoftMax model alone. So, we do not calculate it also for the 3D features.

Results obtained can be seen in Figure 18 expressed with the Overlap Ratio (OR), were it can be seen that the best results are those obtained with $\lambda = 0$, and the other two configurations are worst in every case.

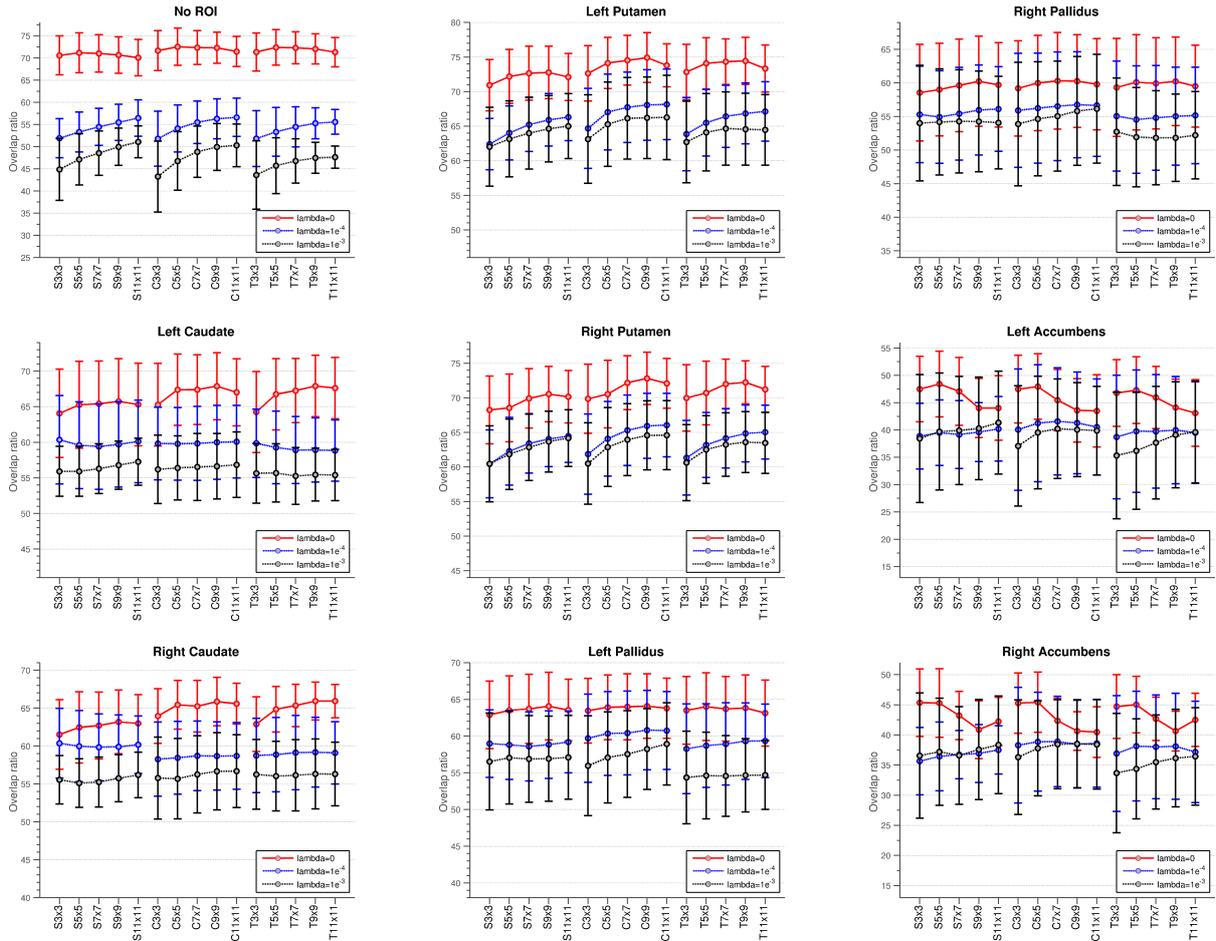


Figure 18: SoftMax OR results with $\lambda = [0, 1e^{-4}, 1e^{-3}]$ for the 8 structures and the NoROI tag.

6.1.2 SSAE

In order to assess how much gain do the extra hidden layers produce in the final results, cross-validated results are obtained again for the 2D feature sets using the SSAE algorithm. As we have described above in Section 4.3, the last layer that composes its structure is a SoftMax algorithm, so our comparison is legitimate. Results can be seen in Figure 19.

We can appreciate a generalized gain with respect of using only the SoftMax model.

6.1.3 PCA + SSAE

Now that we know that SSAE produces a significant gain with respect to SoftMax classification, let's see how much gain do we get with respect to other Machine Learning techniques with no common structure. As we will compare the results with the SVM model, and during the parameter tuning phase it has demonstrated to be much slower

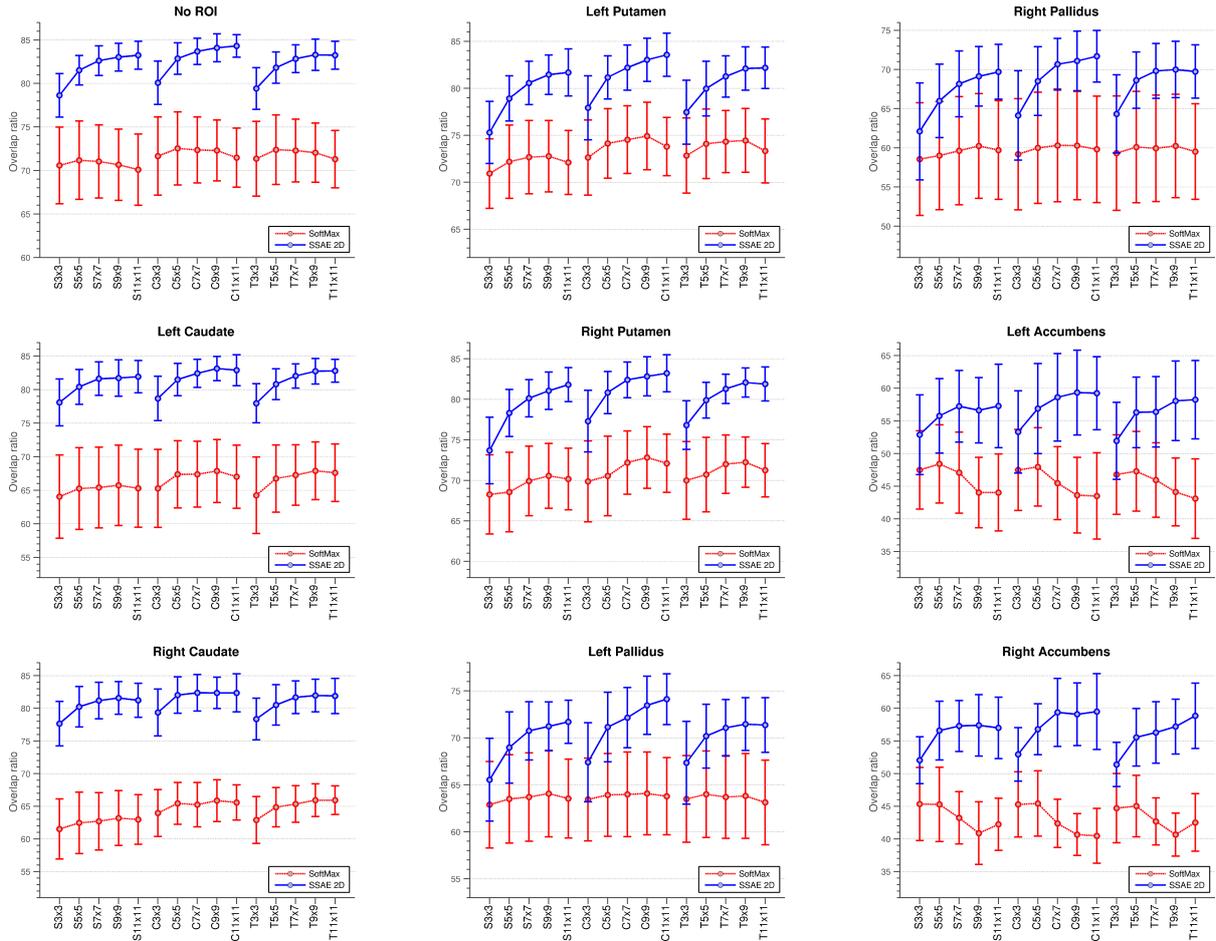


Figure 19: SSAE OR results with the 2D feature sets, compared with the best results obtained with SoftMax alone with the same feature sets.

than SSAE, we produce now an intermediate step of dimensionality reduction with PCA.

First, we apply PCA to the 2D feature sets in order to assess how much loss of classification power does it induce. Results are shown in Figure 20. Note that the 3x3 case has been skipped, as it reported the worst results without PCA.

It can be seen that, compared to the case of SSAE without PCA, results are worst in general around 1% in their Overlap. Now that we know approximately the kind of loss we can expect, we apply PCA to the 3D feature sets. The purpose of this is that, as they are fewer and can be trained in a faster manner, they will provide a good comparison when training on this data an SVM.

Now we are in position to compare the performance of the SSAE method with the performance of the NN and SVM models.

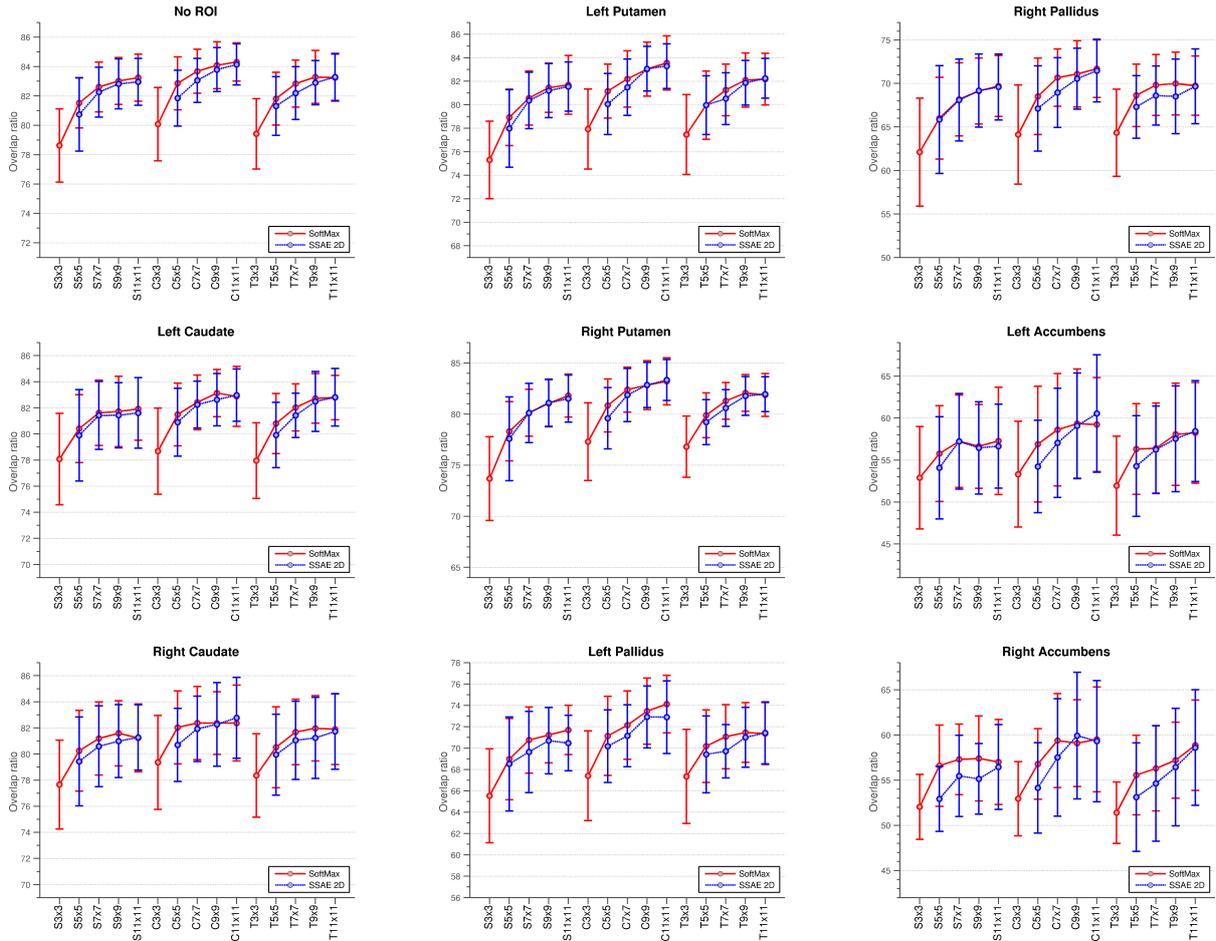


Figure 20: SS AE OR results for the 2D feature sets with and without PCA preprocessing.

6.1.4 Neural Network

We have performed cross-validation with two Neural Network architectures (with one and two hidden layers) and with the SS AE model using 3D data and we have found some interesting effects. First of all, the Overlap coefficient obtained with only one layer is worst than using the two layer approach, as can be noticed in Figure 21. This results are the expected ones, as we know that some studies ([24] and [12]) demonstrate the fact that using deeper architectures one can expect to represent more compactly a broader variety of functions. This is demonstrated by the fact that we are using in general a smaller number of neurons in the case with two hidden layers than in the case with only one hidden layer, and the latter reports a worst generalization.

Second, we can appreciate that the case with only one layer is simply not good at all for the smaller structures in our ROIs which are the Nucleus Accubens and the Putamen. The variability that is shown in the results is sometimes bigger than the mean Overlap Ratio. This means that for some of the validation sets the model misses almost

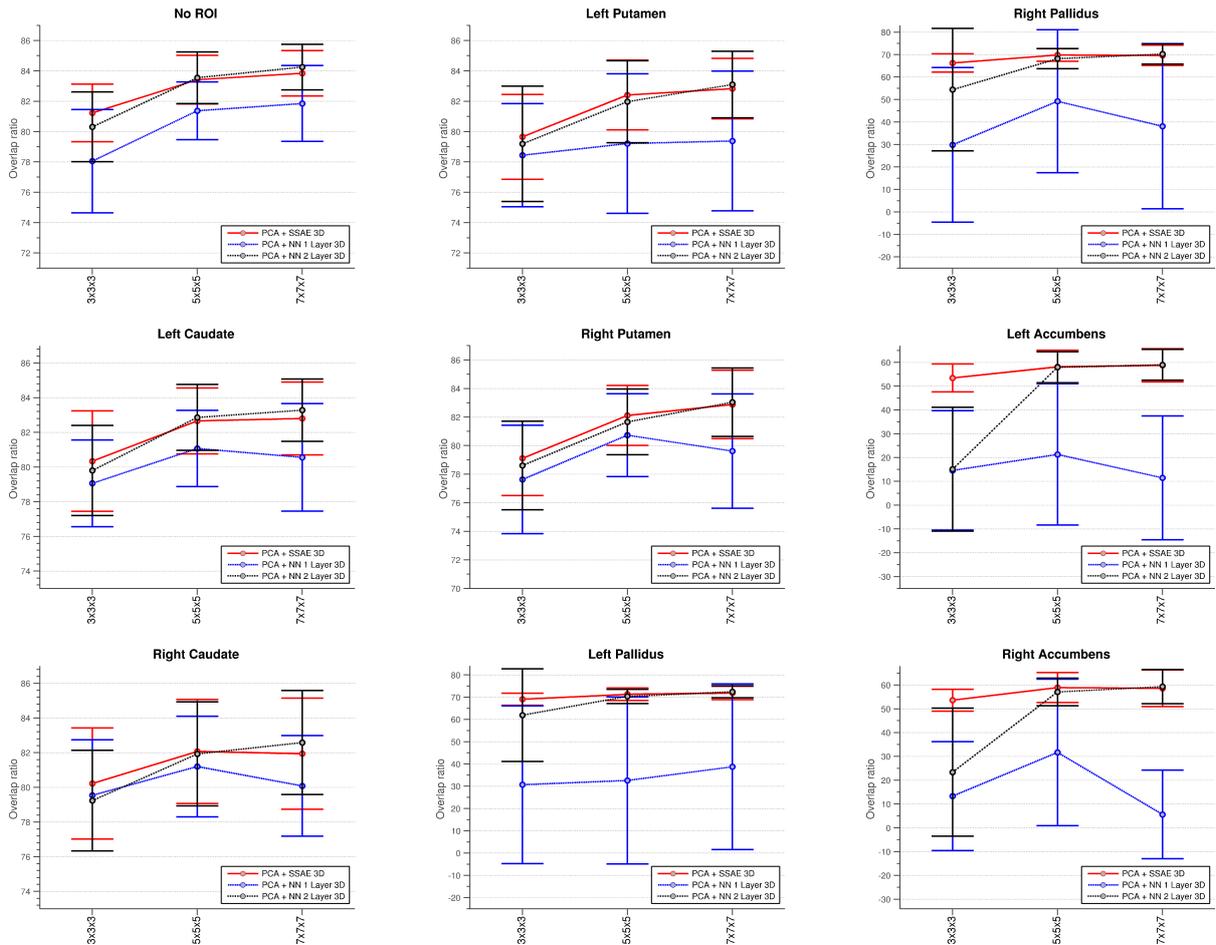


Figure 21: SSAE OR results for all 2D feature sets after applying PCA to them, compared with the results obtained for the Neural Nets with one and two hidden layers using the 3D features.

completely every voxel of the structure, which makes this particular architecture totally undesirable for segmentation purposes.

Finally, we see that for the 7x7x7 set the 2 hidden layer NN model reports slightly better results on average than the SSAE model in any feature set, although the variability of both makes undecided which one works best. Nevertheless, when there is not much information such as is the case of the 3x3x3 feature set, the two layer NN has the same problem that the one layer NN architecture, it has an enormous variability and we can understand by this fact that for some predictions it does not manage to identify the structures with smaller size. Comparatively, SSAE model manages to make a good job in the 3x3x3 feature set, even though not optimal, but still represents an improvement. This is the reason we will prefer the SSAE model over the NN with one or two layers.

6.1.5 SVM

Due to time constraints, the cross-validation of the SVM model has only been done with a 20% of the data chosen randomly from the original datasets. Even in that way, each training for the 7x7x7 case lasted on average 1.11 ± 0.5 hours, for 5x5x5 case it lasted on average 1.72 ± 0.51 hours, and for 3x3x3 features it lasted on average 7.06 ± 2.5 hours. This represents about seven times more than the SSAE or the NN when these models are trained with the complete training set. For comparison purposes, the SSAE was trained once more with just the 20% of the data. Results can be seen in Figure 22.

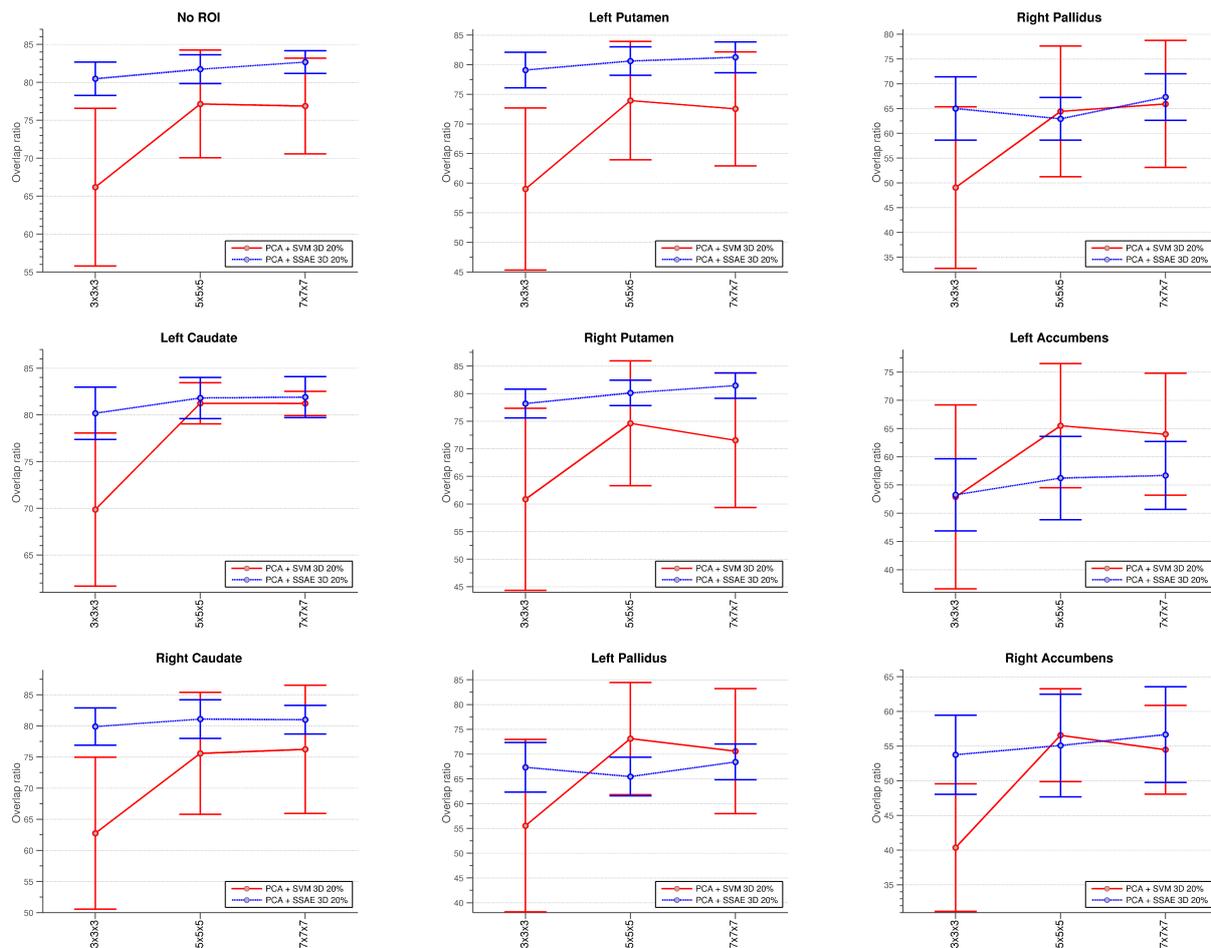


Figure 22: SSAE OR results for the 20% of the data in the 3D features compared with the results obtained by the SVM model for the same data.

We can appreciate two facts when comparing the results. First, results are considerably less accurate with SVM than with SSAE for the bigger structures. Second, although for the smaller structures SVM gets better results in the structures of the left hemisphere of the brain, it suffers from a much bigger variability in these datasets than SSAE and it does not manage to beat SSAE in the right hemisphere. If we join those facts to the fact that SVM takes considerably more time in reporting results, choosing SSAE over SVM

seem an obvious option.

It is interesting to notice also that SSAE manages, with only 20% of the data, Overlap Rates of only around 3% less than with the full training set, with seems as an indication of how good this model really is in the situations in which there is not much information.

6.2 Summary of Cross-Validation Results

In order to facilitate comparison with future works and previous works in the literature, a summary of the best results obtained for every method can be found in Table 1 expressed with the Overlap Ratio. In this table, last two column express the average of the Overlaps and the average of the standard deviations obtained.

	NoROI	LCaudate	RCaudate	LPutamen	RPutamen	LPallidus	RPallidus	LAccumb	RAccumb	AverageMean	AverageStd
SoftMax 2D (C5x5)	72.53±4.2	67.38±5.0	65.44±3.2	74.14±3.7	70.54±4.9	63.92±4.4	59.99±7.1	47.95±6.0	45.43±5.0	63.04	4.8
SSAE 2D (C11x11)	84.31±1.3	82.88±2.3	82.37±2.9	83.56±2.3	83.21±2.3	74.12±2.7	71.69±3.3	59.24±5.6	59.51±5.8	75.65	3.2
SSAE+PCA 2D (C11x11)	84.14±1.4	82.97±2.0	82.77±3.1	83.29±1.9	83.34±2.0	72.89±3.4	71.47±3.6	60.55±7.0	59.31±6.7	75.64	3.5
SSAE+PCA 3D (7x7x7)	83.84±1.5	82.80±2.1	81.94±3.2	82.83±2.0	82.88±2.4	71.87±3.0	69.63±4.5	58.75±7.0	58.58±7.7	74.79	3.7
NN1+PCA 3D (5x5x5)	81.37±1.9	81.07±2.2	81.20±2.9	79.21±4.6	80.73±2.9	32.59±37.5	49.23±31.8	21.29±29.7	31.67±30.8	59.82	16.0
NN2+PCA 3D (7x7x7)	84.25±1.5	83.28±1.8	82.58±3.0	83.10±2.2	83.04±2.4	72.41±2.7	70.24±4.5	58.93±6.5	59.31±7.2	75.24	3.5
SVM+PCA 3D 20% (5x5x5)	77.15±7.1	81.24±2.2	75.58±9.8	73.95±10.0	74.63±11.3	73.12±11.3	64.42±13.2	65.51±11.0	56.57±6.7	71.35	9.2
SSAE+PCA 3D 20% (7x7x7)	82.67±1.5	81.90±2.2	80.98±2.3	81.26±2.6	81.47±2.3	68.41±3.6	67.31±4.7	56.69±6.0	56.66±6.9	73.04	3.6

Table 1: Summary of best results obtained with every method expressed with the Overlap Ratio

We see once again that the SSAE with the 2D feature sets shows on average the best results. Furthermore, we see that the 2D results are better than the 3D ones. Also, we see that the one layer Neural Network has the worst performance, and that the SVM has the second greater variability. Moreover, we see the surprisingly good results and surprisingly low deviation shown on average by the SSAE with only the 20% of the 3D PCA preprocessed data, it indicates that it is a very robust method that manages to learn better than the other methods studied. The same results for the best configurations of the SSAE method can also be found, expressed with the Dice Coefficient, in Table 2 .

	NoROI	LCaudate	RCaudate	LPutamen	RPutamen	LPallidus	RPallidus	LAccumb	RAccumb
SSAE 2D (C11x11)	91.48±0.8	90.63±1.4	90.31±1.7	91.03±1.4	90.82±1.4	85.11±1.8	83.47±2.2	74.26±4.4	74.46±4.6
SSAE+PCA 2D (C11x11)	91.38±0.8	90.68±1.2	90.55±1.9	90.87±1.1	90.90±1.2	84.28±2.3	83.32±2.5	75.22±5.4	74.25±5.4

Table 2: Summary of best results obtained with the SSAE method expressed with the Dice Coefficient

It is interesting to observe that the SVM method actually got the best results for the Accumbens using only the 20% of the data. Nevertheless, for all the reasons stated above,

and due to the time requirements of SVM as compared to SSAE, we will still prefer the later.

6.3 Test with Pathological Group

Now that we have selected SSAE as the best method, we present the final results of how it generalizes over new unseen data. For this purpose, we choose the feature set that reported the best results in cross-validation, which was the 2D coronal section with 11x11 patches. We train the SSAE method with the 22 healthy controls, and then test it with the three different diagnostic sets we have available: 27 MRIs of Bipolar without psychosis subjects, 17 MRIs of Bipolar with psychosis and 12 MRIs of Schizophrenic subjects. The results can be seen in the Table 3.

	NoROI	LCaudate	RCaudate	LPutamen	RPutamen	LPallidus	RPallidus	LAccumb	RAccumb
Bipolar	84.28	80.87	79.48	82.30	82.53	71.18	72.40	60.48	58.42
Bipolar + psi	85.10	80.85	80.54	82.59	82.84	73.17	72.53	59.14	58.28
Schizophrenia	84.58	81.37	79.38	82.11	82.73	73.21	72.56	57.32	58.15

Table 3: Test results expressed in OR using the SSAE method with 2D coronal 11x11 features for the different diagnostic groups.

For compatibility purposes, in Table 4 you can find the same results expressed with the Dice Coefficient.

	NoROI	LCaudate	RCaudate	LPutamen	RPutamen	LPallidus	RPallidus	LAccumb	RAccumb
Bipolar	0.9147	0.8942	0.8857	0.9029	0.9043	0.8316	0.8399	0.7537	0.7375
Bipolar + psi	0.9195	0.8941	0.8922	0.9047	0.9062	0.8451	0.8408	0.7433	0.7364
Schizophrenia	0.9164	0.8973	0.8850	0.9017	0.9055	0.8454	0.8410	0.7287	0.7354

Table 4: Test results expressed with the Dice Coefficient using the SSAE method with 2D coronal 11x11 features for the different diagnostic groups.

As we can see, SSAE is a very robust method and gives us the expected results also in pathological cases. In order to assess the actual segmentations, Figure 23 shows qualitative results of the segmentation produced by the SSAE method in one of the subjects from the pathological group. We can appreciate that, although it has some trouble detecting the borders, it almost perfectly detects the central region of the ROIs. This effect was expected, since if we were comparing two manual segmentations made by either the same expert or different neurologists, we would see a similar effect, both would agree in the center, but disagree in the borders. This effect has its cause in the fact that there is not a clear frontier between regions, but instead a color intensity gradient. This problem is even worse for the Nucleus Accumbens, since, as we commented on Section 2.6, MRIs do not show where does it split from the Caudate. Nevertheless, despite this fact,

we see in the image that SSAE, although not perfectly, manages to locate and segment the Accumbens in a proper manner.

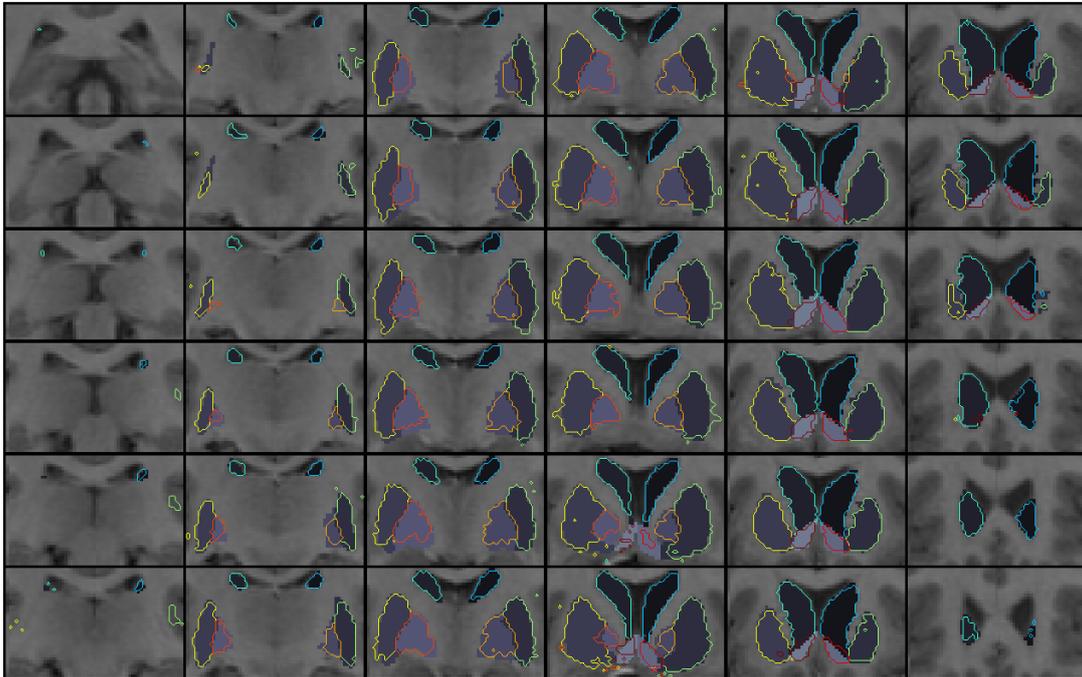


Figure 23: Segmentation produced by the best SSAE method. The contours show the automatic segmentation and the

7 Discussion

In this study, we have explored Deep Learning techniques applied to the brain MRI segmentation problem. We have seen that in general deeper structures should be preferred to shallower ones in order to extract the hidden richness of the data.

In particular, we have also seen that, on average, SSAE report the best results of all the methods tried. SSAE might be the method with the most parameters to tune up, but it also seems to be the most robust method after tuning is properly done. We have observed this fact in every situation in which there was little information, whether when the patches were small or when we were using a sub-sample of the data, SSAE consistently reported the best results with respect to the other methods.

In the benefit of NN, we might say that its results closely follow those reported for SSAE when there is enough information. About SVM with Radial Basis function, we can say it needs much less memory to operate than SSAE or NN, but it has demonstrated to be much slower in general. The fact that SSAE was entirely implemented in Matlab and LibSVM is programmed in C makes the time problem even worst.

About the selected features, we have seen that the 3D approach does not add more information than the 2D approach. This may be understood by thinking on the inter-rater variability and what does it really mean. It means that any neurologist manually segmenting the regions in the brain will not only use objective clinical information, but subjective one as well; there will be so much of it that his segmentation will differ in about a 20% with any segmentation made by any of his colleagues. What that really means is that the atlases we are using as ground truth have about a 20% of data that has nothing to do with the actual shape of the ROI, or any physical characteristic of it, but more with the particular personality of the person who manually segmented it. Since we have no data of the knowledge of that person, but only intensity values for the voxels, and since manual segmentations are done in a 2D manner, we can then expect to find enough information in a 2D slice that there is in a 3D one, as has been shown to be the case in our experiments.

We have to say that several other approaches were studied that do not appear in the memory of the project. For instance, before using the SoftMax model, simple Logistic Regression was tried in order to decide which feature sets would be used for the project. Many different configurations were tried, but since the method reported poor results in the test set, and results were not naturally converted to multi-label classification, it was difficult to justify why we were using that approach, and it was abandoned. Also, the original idea was to use a final Majority Vote technique with all the models based on 2D feature sets, but all experiments gave disappointing results and so it was not included in this study.

Furthermore, we tried to use a kind of unsupervised pretraining known as Self-Taught Learning which purpose is to make use of the loads of MRIs that can be obtained in all the open datasets that are nowadays on-line, except that they have no Atlas associated. The idea is to let the model learn in an unsupervised way the variability of the data it can expect, and then use the small manually segmented set you have to really let it fine-tune the weights. Although this approach allowed better results, it was a too low gain for the computational and temporal cost it represented, so that approach was also abandoned.

For the future work, we wish to move forward and try other structures such as Stacked Contractive Sparse Autoencoders, which are really similar to SSAE except that instead of the weight decay term and the sparsity parameter they use the Frobenius Norm of the Jacobian Matrix in a hope that it will make a similar effect, and they are actually reporting state-of-the-art results in classification experiments.

We would also try Stacked Denoising Autoencoders which basically follow the same philosophy of greedy layerwise training of Autoencoder layers, but each layer has a pre-training phase of adding Gaussian noise to the input before autoencoding it. This pre-

training allows each layer to learn not only the training set but also how it might expect new inputs to vary from the training set, and are actually also reporting state-of-the-art results.

Finally, we would like to compare the performance of these techniques with other now "classical" Deep Learning approaches, such as Deep Belief Restricted Boltzmann Machines (The model that showed to the world it was possible to go deeper in the training of networks by greedy layerwise training) or Max-Pooling Convolutional Neural Networks.

References

- [1] A. Akselrod-Ballin, M. Galun, M. J. Gomori, R. Basri, and A. Brandt. Atlas guided identification of brain structures by combining 3D segmentation and SVM classification. *Med Image Comput Comput Assist Interv*, 9(Pt 2):209–216, 2006.
- [2] P. Aljabar, R. Heckemann, A. Hammers, J. V. Hajnal, and D. Rueckert. Classifier selection strategies for label fusion using large atlas databases. *Med Image Comput Comput Assist Interv*, 10(Pt 1):523–531, 2007.
- [3] P. Aljabar, R. A. Heckemann, A. Hammers, J. V. Hajnal, and D. Rueckert. Multi-atlas based segmentation of brain images: atlas selection and its effect on accuracy. *Neuroimage*, 46(3):726–738, Jul 2009.
- [4] A Andreopoulos and J K Tsotsos. A novel algorithm for fitting 3d active appearance models: Applications to cardiac MRI segmentation. *Proceedings of the 14th Scandinavian Conference on Image Analysis*, pages 729–739, 2005.
- [5] J.A. Armengol Butron de Mjica, C.J. Catalina Herrera, E.M. Perez Villegas, F. Rubia, I. Morgado Bernal, and J.M. Delgado Garcia. *Manual del Master de Neurociencia y Biología del Conocimiento 2011*. Viguera Editores S.L., 2011.
- [6] John Ashburner, Gareth Barnes, Chun-Chuan Chen, Jean Daunizeau, Guillaume Flandin, Karl Friston, Darren Gitelman, Stefan Kiebel, James Kilner, Vladimir Litvak, Rosalyn Moran, Will Penny, Klaas Stephan, Darren Gitelman, Rik Henson, Chloe Hutton, Volkmar Glauche, Jeremie Mattout, and Christophe Phillips. *SPM8 manual*, jul 2010.
- [7] K. O. Babalola, B. Patenaude, P. Aljabar, J. Schnabel, D. Kennedy, W. Crum, S. Smith, T. Cootes, M. Jenkinson, and D. Rueckert. An evaluation of four automatic methods of segmenting the subcortical structures in the brain. *Neuroimage*, 47(4):1435–1447, Oct 2009.
- [8] K. O. Babalola, V. Petrovic, T. F. Cootes, C. J. Taylor, C. J. Twining, and A Mills. Automatic segmentation of the caudate nuclei using active appearance models. In T. Heimann, M. Styner, and B. van Ginneken, editors, *3D Segmentation In The Clinic: A Grand Challenge*, pages 57–64, 2007.

- [9] Kola Babalola and Tim Cootes. Using parts and geometry models to initialise active appearance models for automated segmentation of 3d medical images. In *Proceedings of the 2010 IEEE international conference on Biomedical imaging: from nano to Macro*, ISBI'10, pages 1069–1072, Piscataway, NJ, USA, 2010. IEEE Press.
- [10] J. Bai, T. L. Trinh, K. H. Chuang, and A. Qiu. Atlas-based automatic mouse brain image segmentation revisited: model complexity vs. image registration. *Magn Reson Imaging*, Mar 2012.
- [11] S. Bauer, C. Seiler, T. Bardyn, P. Buechler, and M. Reyes. Atlas-based segmentation of brain tumor images using a Markov random field-based tumor growth model and non-rigid registration. *Conf Proc IEEE Eng Med Biol Soc*, 2010:4080–4083, 2010.
- [12] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In Bernhard Schölkopf, John Platt, and Thomas Hoffman, editors, *Advances in Neural Information Processing Systems 19 (NIPS'06)*, pages 153–160. MIT Press, 2007.
- [13] S. Carmona, O. Vilarroya, A. Bielsa, V. Trèmols, J.C. C. Soliva, M. Rovira, J. Tomàs, C. Raheb, J.D. Gispert, S. Batlle, and A. Bulbena. Global and regional gray matter reductions in ADHD: A voxel-based morphometric study. *Neuroscience Letters*, 389(2):88–93, 2005.
- [14] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.
- [15] L. P. Clarke, R. P. Velthuizen, S. Phuphanich, J. D. Schellenberg, J. A. Arrington, and M. Silbiger. MRI: stability of three supervised segmentation techniques. *Magn Reson Imaging*, 11(1):95–106, 1993.
- [16] Adam Coates, Honglak Lee, and A. Y. Ng. An analysis of Single-Layer networks in unsupervised feature learning. *Engineering*, pages 1–9, 2010.
- [17] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *Proceedings of the European Conference on Computer Vision*, 2:484–498, 1998.
- [18] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models—their training and application. *Comput. Vis. Image Underst.*, 61(1):38–59, jan 1995.
- [19] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.
- [20] I. Despotovic, I. Segers, L. Platasa, E. Vansteenkiste, A. Pizurica, K. Deblaere, and W. Philips. Automatic 3D graph cuts for brain cortex segmentation in patients with focal cortical dysplasia. *Conf Proc IEEE Eng Med Biol Soc*, 2011:7981–7984, 2011.

- [21] Dumitru Erhan, Pierre-Antoine Manzagol, Yoshua Bengio, Samy Bengio, and Pascal Vincent. The difficulty of training deep architectures and the effect of unsupervised pre-training. In *Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 153–160, 2009.
- [22] B. Fischl, D. H. 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. M. Dale. Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. *Neuron*, 33(3):341–55+, 2002.
- [23] Jean A. Frazier, Steven M. Hodge, Janis L. Breeze, Anthony J. Giuliano, Janine E. Terry, Constance M. Moore, David N. Kennedy, Melissa P. Lopez-Larson, Verne S. Caviness, Larry J. Seidman, Benjamin Zablotzky, and Nikos Makris. Diagnostic and sex effects on limbic volumes in early-onset bipolar disorder and schizophrenia. *Schizophrenia Bulletin*, 34(1):37–46, 2008.
- [24] J. Hastad and M. Goldmann. On the power of small-depth threshold circuits. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science, SFCS '90*, pages 610–618 vol.2, Washington, DC, USA, 1990. IEEE Computer Society.
- [25] R. A. Heckemann, J. V. Hajnal, P. Aljabar, D. Rueckert, and A. Hammers. Automatic anatomical brain MRI segmentation combining label propagation and decision fusion. *Neuroimage*, 33(1):115–126, Oct 2006.
- [26] R. A. Heckemann, S. Keihaninejad, P. Aljabar, D. Rueckert, J. V. Hajnal, and A. Hammers. Improving intersubject image registration using tissue-class information benefits robustness and accuracy of multi-atlas based anatomical segmentation. *Neuroimage*, 51(1):221–227, May 2010.
- [27] Tobias Heimann and Hans-Peter Meinzer. Statistical shape models for 3d medical image segmentation: a review. *Medical Image Analysis*, 13(4):543–563, 2009.
- [28] G E Hinton and R S Zemel. Autoencoders, minimum description length and helmholtz free energy. *Advances in Neural Information Processing Systems 6*, 6(9):3–10, 1994.
- [29] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006.
- [30] A. Huang, R. Abugharbieh, and R. Tam. A novel rotationally invariant region-based hidden Markov model for efficient 3-D image segmentation. *IEEE Trans Image Process*, 19(10):2737–2748, Oct 2010.
- [31] Juan Eugenio Iglesias, Mert Rory Sabuncu, and Koen van Leemput. A generative model for multi-atlas segmentation across modalities. In *IEEE International Symposium on Biomedical Imaging ISBI 2012*, pages 888–891, Barcelona, Spain, 2012.
- [32] L. Igual, J. C. Soliva, A. Hernandez-Vela, S. Escalera, X. Jimenez, O. Vilarroya, and P. Radeva. A fully-automatic caudate nucleus segmentation of brain MRI: application in volumetric analysis of pediatric attention-deficit/hyperactivity disorder. *Biomed Eng Online*, 10:105, 2011.

- [33] D. V. Iosifescu, M. E. Shenton, S. K. Warfield, R. Kikinis, J. Dengler, F. A. Jolesz, and R. W. McCarley. An automated registration algorithm for measuring MRI subcortical brain structures. *Neuroimage*, 6(1):13–25, Jul 1997.
- [34] Flusser Jan, Zitova Barbara, and Suk Tomas. *Moments and Moment Invariants in Pattern Recognition*. Wiley Publishing, 2009.
- [35] David Kennedy and Christian Haselgrove. CANDI neuroimaging access point.
- [36] Ali R. Khan, Moo K. Chung, and Mirza Faisal Beg. Robust atlas-based brain segmentation using multi-structure confidence-weighted registration. In *Proceedings of the 12th International Conference on Medical Image Computing and Computer-Assisted Intervention: Part II*, MICCAI '09, pages 549–557, Berlin, Heidelberg, 2009. Springer-Verlag.
- [37] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? In *Proceedings of the 7th European Conference on Computer Vision-Part III*, ECCV '02, pages 65–81, London, UK, UK, 2002. Springer-Verlag.
- [38] Peter J. Kostelec and Senthil Periaswamy. Image registration for MRI. 46:161–184, 2003.
- [39] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, March 1951.
- [40] Robert Lapp, Maria Lorenzo-Valds, and Daniel Rueckert. 3d/4d cardiac segmentation using active appearance models, non-rigid registration, and the insight toolkit. In Christian Barillot, David Haynor, and Pierre Hellier, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2004*, volume 3216 of *Lecture Notes in Computer Science*, pages 419–426. Springer Berlin / Heidelberg, 2004.
- [41] Quoc V Le, Adam Coates, Bobby Prochnow, and Andrew Y Ng. On optimization methods for deep learning. *Learning*, pages 265–272, 2011.
- [42] L. Liao, T. Lin, and W. Zhang. [Brain MRI image segmentation based on active contour model using electrostatic field method]. *Sheng Wu Yi Xue Gong Cheng Xue Za Zhi*, 25(4):770–773, Aug 2008.
- [43] J. S. Lin, K. S. Cheng, and C. W. Mao. Multispectral magnetic resonance images segmentation using fuzzy Hopfield neural network. *Int. J. Biomed. Comput.*, 42(3):205–214, Aug 1996.
- [44] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Math. Program.*, 45(3):503–528, December 1989.
- [45] Vincent A. Magnotta, Dan Heckel, Nancy C. Andreasen, Ted Cizadlo, Patricia Westmoreland Corson, James C. Ehrhardt, and William T. C. Yuh. Measurement of brain structures with artificial neural networks: Two- and three-dimensional applications1. *Radiology*, 211(3):781–790, 1999.

- [46] Mostafa Jabarouti Moghaddam and Hamid Soltanian-Zadeh. Automatic segmentation of brain structures using geometric moment invariants and artificial neural networks. In Jerry L. Prince, Dzung L. Pham, and Kyle J. Myers, editors, *IPMI*, volume 5636 of *Lecture Notes in Computer Science*, pages 326–337. Springer, 2009.
- [47] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.
- [48] Andrew Ng, Jiquan Ngiam, Chuan Y. Foo, Yifan Mai, and Caroline Suen. UFLDL tutorial, 2010.
- [49] B. Patenaude, S. M. Smith, D. N. Kennedy, and M. Jenkinson. A Bayesian model of shape and appearance for subcortical brain segmentation. *Neuroimage*, 56(3):907–922, Jun 2011.
- [50] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.
- [51] D. W. Piraino, S. C. Amartur, B. J. Richmond, J. P. Schils, J. M. Thome, and P. B. Weber. Segmentation of magnetic resonance images using an artificial neural network. *Proc Annu Symp Comput Appl Med Care*, pages 470–472, 1991.
- [52] S. Powell, V. A. Magnotta, H. Johnson, V. K. Jammalamadaka, R. Pierson, and N. C. Andreasen. Registration and machine learning-based automated segmentation of subcortical and cerebellar brain structures. *Neuroimage*, 39(1):238–247, Jan 2008.
- [53] Marc’Aurelio Ranzato, Y-Lan Boureau, and Yann LeCun. Sparse feature learning for deep belief networks. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1185–1192. MIT Press, Cambridge, MA, 2008.
- [54] Mikaël Rousson, Nikos Paragios, and Rachid Deriche. Implicit active shape models for 3D segmentation in MR imaging. pages 209–216. 2004.
- [55] M. R. Sabuncu, B. T. Yeo, K. Van Leemput, B. Fischl, and P. Golland. A generative model for image segmentation based on label fusion. *IEEE Trans Med Imaging*, 29(10):1714–1729, Oct 2010.
- [56] Mark Schmidt. minfunc, a matlab function for unconstrained optimization of differentiable real-valued multivariate functions using line-search methods.
- [57] Gustavo M. Teixeira, Igor R. Pommeranzembaum, Bernardo L. Oliveira, Marcelo Lobosco, and Rodrigo Weber Dos Santos. Automatic segmentation of cardiac MRI using snakes and genetic algorithms. In *Proceedings of the 8th international conference on Computational Science, Part III, ICCS ’08*, pages 168–177, Berlin, Heidelberg, 2008. Springer-Verlag.

- [58] Paul M Thompson, Elizabeth R Sowell, Nitin Gogtay, Jay N Giedd, Christine N Vidal, Kiralee M Hayashi, Alex Leow, Rob Nicolson, Judith L Rapoport, and Arthur W Toga. Structural MRI and brain development. *International Review of Neurobiology*, 67(05):285–323, 2005.
- [59] Virginia Tremols, Anna Bielsa, Joan-Carles Soliva, Carol Raheb, Susanna Carmona, Josep Tomas, Joan-Domingo Gispert, Mariana Rovira, Jordi Fauquet, Adolf Tobeña, Antoni Bulbena, and Oscar Vilarroya. Differential abnormalities of the head and body of the caudate nucleus in attention deficit-hyperactivity disorder. *Psychiatry Res*, 163(3):270–8, 2008.
- [60] Hans C. van Assen, Mikhail G. Danilouchkine, Alejandro F. Frangi, Sebastian Ordas, Jos J. Westenberg, Johan H. Reiber, and Boudewijn P. Lelieveldt. SPASM: A 3D-ASM for segmentation of sparse and arbitrarily oriented cardiac MRI data. *Medical Image Analysis*, 10(2):286–303, April 2006.
- [61] B. van Ginneken, T. Heimann, and M. Styner. 3D Segmentation in the Clinic: A Grand Challenge. In *3D Segmentation in the Clinic: A Grand Challenge*, pages 7–15, 2007.
- [62] K. Van Leemput, F. Maes, D. Vandermeulen, and P. Suetens. Automated model-based tissue classification of MR images of the brain. *IEEE Trans Med Imaging*, 18(10):897–908, Oct 1999.
- [63] E. M. van Rikxoort, I. Isgum, Y. Arzhaeva, M. Staring, S. Klein, M. A. Viergever, J. P. Pluim, and B. van Ginneken. Adaptive local multi-atlas segmentation: application to the heart and the caudate nucleus. *Med Image Anal*, 14(1):39–49, Feb 2010.
- [64] C. Wang, O. Teboul, F. Michel, S. Essafi, and N. Paragios. 3D knowledge-based segmentation using pose-invariant higher-order graphs. *Med Image Comput Comput Assist Interv*, 13(Pt 3):189–196, 2010.
- [65] S. K. Warfield, K. H. Zou, and W. M. Wells. Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation. *IEEE Trans Med Imaging*, 23(7):903–921, Jul 2004.
- [66] A. Wismuller, F. Vietze, J. Behrends, A. Meyer-Baese, M. Reiser, and H. Ritter. Fully automated biomedical image segmentation by self-organized model adaptation. *Neural Netw*, 17(8-9):1327–1344, 2004.
- [67] A. J. Worth, N. Makris, M. R. Patti, J. M. Goodman, E. A. Hoge, V. S. Caviness, and D. N. Kennedy. Precise segmentation of the lateral ventricles and caudate nucleus in MR brain images using anatomically driven histograms. *IEEE Trans Med Imaging*, 17(2):303–310, Apr 1998.
- [68] J. Wu and A. C. Chung. Markov dependence tree-based segmentation of deep brain structures. *Med Image Comput Comput Assist Interv*, 11(Pt 2):1092–1100, 2008.

- [69] J. Wu and A. C. Chung. A novel framework for segmentation of deep brain structures based on Markov dependence tree. *Neuroimage*, 46(4):1027–1036, Jul 2009.