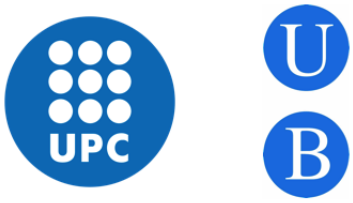

UNIVERSITAT POLITÈCNICA DE CATALUNYA
UNIVERSITAT DE BARCELONA
UNIVERSITAT ROVIRA I VIRGILI

Master in Artificial Intelligence

Master of Science Thesis

Multiple Sclerosis Lesion Segmentation using Deep Learning

Erol Kazancli

FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)
FACULTAT DE MATEMÀTIQUES Y INFORMÀTICA (UB)
ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA (URV)

Supervisor:
Laura Igual Muñoz
Department of Mathematics
and Computer Science,
Universitat de Barcelona (UB)

Co-supervisor:
Paulo Rodrigues
Neuroimaging Company - Mint Labs

June 27, 2017

Acknowledgements

I would like give special thanks to

Laura Igual Muñoz, Associate Professor,
for her supervision, knowledge and support

Pablo Villoslada, Medical Doctor,
for his expertise and knowledge

Vesna Prchkovska, PhD,
for her support and faith in the project

Paulo Rodrigues, PhD,
for his technical support

Contents

Abstract

Multiple Sclerosis (MS) is a chronic neurological disease that affects mainly people between the ages of 20 and 50. It damages the central nervous system causing demyelination (loss of myelin sheath), which results in symptoms such as loss of vision, loss of balance, slurred speech, numbness, fatigue, memory and concentration problems, coordination problems, etc. The lesions caused by demyelination are visible in MRI images in a way that the affected region is hypo-intense (darker) in T1 modality while it is hyper-intense (lighter) in T2 modality. The demarcation of these regions, which is also called MS lesion segmentation, is critical for the diagnosis, treatment and follow-up of the patients. The MS lesion segmentation is a time-consuming manual process that requires certain expertise from medical experts. Moroever, manual segmentation is subject to intra- and inter- expert variability. There have been efforts to automatize and standardize this process but since the problem of MS lesion segmentation is hard to formulate mathematically, the solutions have been limited. As the MRI data has accumulated and the knowledge in machine learning deepened, machine learning methods including deep learning has been applied to this problem, obtaining solutions that outperformed other conventional automatic methods. Especially deep learning methods have turned out to be promising, attaining human expert performance levels. In this thesis, we will apply several approaches of deep learning to the MS Segmentation problem. Our aim is to develop a solution that will help manual segmentation experts in their task and reduce the necessary time and effort in the process. We obtain an average dice score of 57.5% and a true positive rate of 59.7% for a real test dataset of 9 patients from Hospital Clinic, outperforming LST [22] on all measures we employ, which is a commonly used automatic tool in MS lesion segmentation.

1 Introduction

Multiple Sclerosis (MS) is a common neurological disease that afflicts especially the young population between the ages 20 and 50. It affects 2.3 million people worldwide and can cause symptoms such as loss of vision, loss of balance, slurred speech, numbness, fatigue, memory and concentration problems, coordination problems, etc. MS remains a very challenging disease to diagnose and treat, due to its variability in its clinical expression.

MS is characterized by lesions throughout the brain that is caused by the loss of myelin sheath around neurons in the brain, which is also known as demyelination. The lesions are generally ovoid in shape, are of varying sizes, are scattered throughout the brain and are seen mainly in the white matter, and sometimes in the gray matter of the brain. The diagnosis and prognosis of MS is currently guided by conventional structural MRI of brain and spinal cord, seeking for evidence of both dissemination in time and dissemination in space of the MS lesions.

The number and the total volume of MS lesions are indicative of the disease stage and are used to track disease progression. For this reason, the accurate segmentation of lesions is quite important for MS disease in the medical world with the purposes of correct diagnosis, adequate treatment development and prognosis follow-up. Manual segmentation by experts is the most commonly used technique of MS lesions and is still considered to produce the most accurate results although it suffers from many complications. First of all, it is subject to intra- and inter- expert variability, which means there are significant differences between two segmentations performed by two different experts (due to slightly varying definitions) or by the same expert at different times (due to fatigue or similar factors). Secondly there is a shortage of adequately trained experts given the huge amount of segmentation need. Thirdly, the segmentation task requires valuable expert time and concentration, which could be dedicated to other tasks. These drawbacks with manual segmentation makes it necessary and desirable to develop a semi-automatic segmentation method that would assist experts in the task with a reduced amount of time and intra- inter- expert variability or, in the ideal case, a fully automatic segmentation method which would obviate the need for experts and produce accurate/reproducible results.

However, the challenges to obtain accurate and reproducible automatic segmentation are numerous. First of all, there is no single standard acquisition protocol for MRI images. To give an example, the intensity range and the resolution of an MRI image may vary depending on the magnet strength or the acquisition protocol, with which the image was taken. Moreover, MRI images come with a noise introduced during acquisition. Therefore, standardization/pre-processing of the data is a tricky process and one method for all types of MRI images is a distant possibility. Secondly, lesions appear in different shapes and intensities even within a single MRI image, and they may have fuzzy borders, which are difficult to differentiate even for an expert. Thirdly, there might be other abnormal regions in the brain, similar to lesions in appearance, caused by necrosis, inflammation or other brain diseases, which would be difficult for an automatic method to differentiate from MS lesions.

In spite of very promising results obtained with supervised learning methods, usage of these techniques with MRI images has its additional challenges to the ones stated above. Firstly, the supervised learning methods require a big amount of labeled data, especially if the underlying mapping is a rather complex one, which seems to be the case for lesions. This poses a problem with MRI data, since this type of data does not abound, due to the complex and expensive nature of the acquisition process. Moreover, the data is generally kept locally in different institutions and is not made available publicly. Even more scarce is the labeled MRI data, since manual segmentation requires valuable expert knowledge and time. Secondly, the manual segmentation is not a completely reliable process, subject to inter- and intra-expert variability, which will introduce

wrongly labeled data into the training dataset. Thirdly, generally a high number of features is required to obtain a good accuracy, which makes necessary a high number of data, more complex architectures and, therefore, high processing power for the training process. Lastly, the training data is very unbalanced, given the fact that lesion regions are very few in number compared to non-lesion regions. This makes it necessary to adequately sub-sample data or design appropriate loss functions.

Recently, deep learning has been quite successful and popular in the area of Computer Vision, achieving improvements in accuracies sometimes as high as 30%. The main strength in deep learning, also differentiating it from other machine learning methods, is their automatic feature extraction capability. In general, raw data has to processed automatically or manually to extract meaningful and useful features. This process requires time and careful analysis, and includes subjectivity on the part of the expert, which might bias the results or produce erroneous results. However, in deep learning, the feature extraction is driven by data, an appropriate loss function and a learning algorithm, which removes the subjectivity, randomness and expert knowledge to a certain degree. Moreover, the features obtained are hierarchical, each layer producing more abstract features using the less abstract features obtained in the previous layer. This way feature extraction is carried out step-by-step, which is more likely to produce more complex and useful features. Another strength of deep learning is its ability to represent very complex functions, which might also be considered as a drawback since it is prone to easily overfitting, but with the correct guidance and regularization methods the overfitting can be prevented. Deep learning methods are also robust to outliers, which is very common in neuroimaging data.

There are also disadvantages to deep learning methods such as hard-to-explain nature of it and computational complexity. Visualisation of the features in methods like Convolutional Neural Networks (CNN) helps for better explainability of deep learning models. Novel methods such as CNNs/Convolutional Encoder Networks (CENs) provide weight sharing methods which eases the computational costs. In addition, improving hardware (GPUs) and the advent of cloud technologies (such as Amazon Web Services) removes the computational efficiency need to a great extent.

The problem of MS Lesion Segmentation is a difficult problem, since there is no clear or agreed-upon definition of how to segment MS Lesions on an MRI image. There are some guidelines that human experts follow but they also apply their own knowledge or experience in the field, which may lead to subjective judgments. To perform the segmentation, experts inspect the MRI images and delineate lesions in a voxel-by-voxel fashion following the guidelines, knowledge, personal experience and their preferences (to be generous or conservative). For this reason, MS lesion segmentation problem is difficult to explicitly formulate mathematically, which makes it suitable for a machine learning solution. There are some tools commonly used such as LST to help the experts alleviate their work load to some degree but they are far from reaching human performance levels. Recently deep learning methods have been applied to this problem and it has been reported in some of these studies that human level performance has been reached by these methods.

In this thesis, we will implement an MS Lesion Segmentation method with Deep Learning using a combination of different approaches stated in the state-of-the-art together with other approaches that we have come up with. In this study we collaborate with Mint Labs, a start-up specialized in image processing in neuroimaging and with experts on MS from Hospital Clinic. We performed our experiments on Amazon Web Services, which enabled us to work with complex neural networks and a high amount of data in a reasonable amount of time. The aim of this study is to achieve a method that will surpass the performance of existing methods in helping the experts in the MS Lesion Segmentation work and even make their interruption minimal.

The contribution of this work is to try different approaches with deep learning so far applied separately to MS Segmentation in a single study, to explore a new sub-sampling method to improve the learning process and to develop our own approaches to obtain better performance results. Moreover, we work on both white and gray matter, while the majority of the work in the literature is only with white matter.

The following chapters are organized as follows. Chapter 2 explains the of State-of-the-art, in which we review the neuroimaging technologies, understanding and pre-processing of MRI data, Manual MS Segmentation, Deep Learning Techniques and the existing literature on MS Segmentation using deep learning. In Chapter 3, we discuss the methodology we apply. In Chapter 4 we explain the different approaches we have implemented and the results obtained, comparing these approaches between each other and also with LST, which is a very commonly used tool for automatic MS Segmentation in the medical domain. In Chapter 5 we discuss the conclusion. In Chapter 6 we talk about the future work, suggesting what can be done to improve the achieved results.

2 State-of-the-art

In this chapter, we first review the neuroimaging technologies since it is important to know about these technologies, especially MRI imaging, to understand the study carried out in this thesis. Following the discussion about neuroimaging, we give information about MRI data and the pre-processing techniques applied to the MRI data. Later, we review the deep learning techniques, which have been successful in a wide range of fields including neuroimaging. After this section, we discuss about how MS Segmentation is done manually, based on the observations and notes in a meeting with an manual segmentation expert. As the final section we make a literature review on the deep learning techniques so far applied to the MS Segmentation problem.

2.1 Neuroimaging

There are so many conditions or diseases originating from or affecting the brain, which are related to physical, physiological, or psychological functioning of the body, that it is very critical for human life to be able to observe the brain as data itself and extract information from it. This data ideally should give us information on what abnormality there is from the norm, where it is stemming from, how it is progressing, etc. It is important to understand the brain, not only from the perspective of abnormalities, but also to be able to grasp the nature of normal human behaviours, feelings, bodily processes because a grand majority of these are brain-originated.

The brain is such a delicate organ and therefore protected by nature with such a strong case that it has been always difficult, limited and risky to analyse the brain invasively on a living human body. There has always been a need for being able to examine the brain structure and processes non-invasively. Thankfully, recent advances in science have made it possible to develop techniques to investigate the brain in a non-invasive fashion. Now we have a range of techniques which allow us to throw a different light on brain and study a different aspect of it. Here are some of the techniques commonly used in neuroscience to throw light on the structure and inner workings of the brain, which produce the big bulk of the neuroscience data:

- **EEG:** Electroencephalography (EEG) is a technique to measure and record the electrical activity of the brain. EEG measures the electrical activity of the brain over a period of time, with the help of electrodes placed on scalp, which capture voltage fluctuations within the neurons of the brain and send them as signals to a specialized computer where these signals are recorded. EEG is used to detect abnormalities in the normal electrical activity of the brain in the diagnosis of conditions or diseases like epilepsy, sleep disorders, coma, stroke, etc. It has very limited spatial resolution but very fine temporal resolution.
- **PET:** Positron Emission Tomography (PET) is a technique based on nuclear processes that measures the metabolic processes in the body. The technique works by detecting gamma-rays emitted by a tracer molecule introduced into the body, which then constructs a three dimensional image of the concentration of the molecule throughout the body. It is mainly used in oncology to detect tumors and metastases.
- **MEG:** Magnetoencephalography (MEG) is a technique to measure brain activity by detecting the magnetic field produced by the electrical current in the brain. The technique may be used to detect the abnormalities in the brain by the changes in the magnetic field or may simply be used in experimental settings to observe the magnetic/electrical activity of the brain.
- **MRI:** Magnetic Resonance Imaging (MRI) is a medical imaging technique to form pictures of the anatomy and medical processes of the body, which make use of magnetic fields, radio waves or field gradients. MRI is a very powerful technology in neuroscience allowing to investigate the details of human brain in various perspectives. It is so flexible in the sense

that one can measure the macrostructural features of the brain (such as cortex thickness or volumes of different sub-structures), microstructural features (such as diffusion within white matter tracts) or functional activations in the brain (task-driven activations or resting state properties). The scale of the MRI image data may vary depending on the modality and the resolution used. MRI can be considered under three categories, structural MRI, functional MRI and diffusion MRI.

- **sMRI:** sMRI provides the static anatomical information, such as the size and shape of the white and gray matter. In structural MRI, different structures appear in different levels of contrast. It is used to observe the structural abnormalities in the brain, such as tumors, lesions, etc. It has several modalities, such as T1, T2, PD Weighted and FLAIR (See Figure 1).

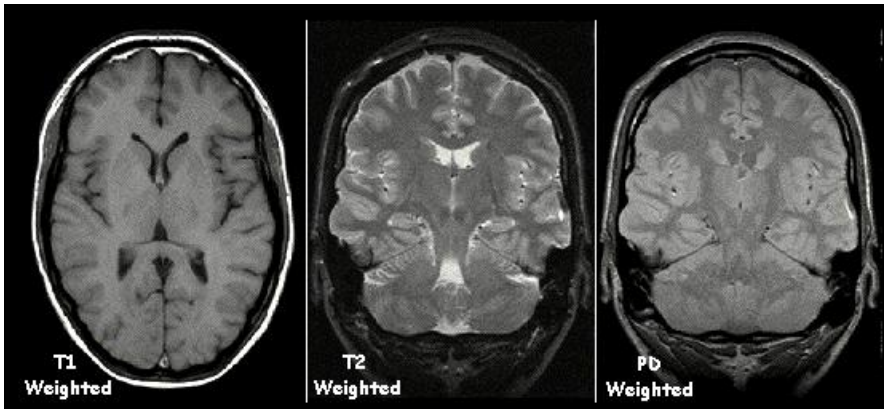

Figure 1: Different modalities of sMRI [27]

- **fMRI:** Functional MRI (fMRI) measures indirectly the brain activity by directly measuring the oxygenation level of blood throughout the brain. The main aim of this technique is to find a relationship between structure and function. The scanning is typically accompanied with a behavioural task and it is done taking many scans through time within a short time period. Thus, the resulting data is a 4D image, which is converted to a 3D contrast image.
- **Diffusion MRI:** In this scanning technique the relative motion of water through each voxel in different directions is captured. A tensor analysis is carried out to calculate the direction and the intensity of the motion through each voxel. Since it is known that water moves more easily along axons than across them, this technique mainly gives information about the connectivity throughout the brain. The abnormality in the connections may indicate a structural or functional abnormality in the brain, such as tumors.

2.2 Understanding and Pre-processing MRI Data

Neuroimaging techniques are generally based on special scanners of the brain, which turn tissue properties into volume-elements (voxels) in certain resolutions. This scan can be carried out in different orientations (radiological, neurological), in varying dimensions (2D, 3D, 4D), in different spatial and temporal resolutions. These data are generally stored in a raw binary format consisting of either 8- or 16- bit integers. Along with this raw data, there is also usually the metadata including the descriptive information such as, subject information, type of image, imaging parameters and image dimensions.

Even though MRI process is non-invasive, it can cause some level of disturbance to the patient. For instance, it requires the patient to stay still in a certain position until the scanning is complete. This poses a problem to the standardness of the data, which is specifically important for automatic techniques, because it is never possible to fix the head in a certain position without any movement. The measurements also contain some amount of noise, which may affect the quality of the data and therefore require further pre-processing.

The possible potential side effects, which are still to be investigated, is another issue in the usage of MRI techniques. The doctors look for a valid reason to require the patient to go through a scan in order not to subject him to unnecessary disturbance and danger. Moreover, the scanning process does not come cheap, require special machines, equipment, staff and therefore is prescribed with parsimony. For these reasons, the data does not abound, which is important in data mining and machine learning techniques.

Each of the MRI modalities measure different indicators and one cannot in general be converted into the other. Therefore, even though they all provide images from the brain, they contain different kinds of information on brain and need to considered in different categories. Besides the obvious differences between different modalities, there is also variation within the same modality. For one given modality, there might be different machines, different resolutions, image quality, etc and this causes problems in the standardness of the data.

Because of all this variation in brain data, it is crucial to bring them to similar distributions using some pre-processing techniques. Skull-stripping (removing the skull from images), field-bias correcting (removing the undesired signal from images), motion correction (removing the effects of the movement of the head in different images and aligning them to the same space), intensity normalization such as 0-mean unit variance standardization or histogram normalization are the common pre-processing steps applied to MRI images to obtain similar distributions. For further info see [2] and [3]

2.3 Manual MS Segmentation using MRI

To solve the problem of MS Segmentation, it is important to clearly understand the problem. For this purpose, as a supplement to the available coarse definitions in the literature, we came together with an MS segmentation expert in Hospital Clinic to understand and observe what rules or intuitions an expert is applying while performing the segmentation task. To support and clear the knowledge we gained, we also kept regularly in touch with a doctor from the same hospital and exchanged ideas. In addition to understanding the problem, we also discussed about the needs and difficulties an expert has regarding MS segmentation problem. The following information in this section is based on our notes and understandings from this meeting.

The manual segmentation is done by initially looking at the T1 image. In the white matter, periventricular area, or at the cortex area bordering white matter, if there is a hypo-intense region, i.e. a region darker-than-normal or darker than its surrounding area, then this is a "candidate" lesion. To mark it as a "definite" lesion, T2-flair image is checked. If the corresponding area in T2-flair is hyper-intense meaning that it is brighter than its surrounding area, then this is marked as an MS lesion. The border of the MS lesion is determined by the T1 hypo-intense region since the corresponding T2 hyper-intense region tends to be larger. The periventricular area and the juxtacortical area (white matter bordering cortex) are particularly prone to contain lesions. The lesions can appear in the cortex as well and are generally an extension of a lesion in the white matter. Lesions are generally ovoid in shape (or a collection of ovoids) and remain the same or get bigger in time (do not get smaller or disappear altogether). See Figure 2 for an example of manually segmented MRI.

Challenges:

- In T2-flair, a thin area surrounding ventricular region appears bright as if it were a lesion but in reality it is not. However, if a bright zone in this region spans a greater-than-normal area and is darker-than-normal in the corresponding T1 zone, then it is a lesion.
- Borders of lesions may be difficult to identify, a voxel is labeled as a lesion only if the evidence is sufficient. If there is doubt, it is not labeled as a lesion.
- Lesions around the cortex are particularly difficult to detect since they look like a cortex tissue. This can be deduced from the shape of the cortex, or in our case, with the help of correct tissue-segmentation.
- There might be darker-than-normal regions in the white matter in T1 or brighter-than-normal regions in T2-flair that do not point to a lesion. Only if these two conditions are fulfilled together than this points to a lesion.
- In the ventricular region around the vessels (that are small black hole in T1 and T2-flair), there might appear small brighter-than-normal zones in T2-flair but these are not lesions.
- The choroids plexus are inside the ventricles and frequently are calcificated and are bright in Flair sequence. Thankfully, these are not darker-than-normal in T1. Moreover, lesions do not occur inside the ventricles, so can be ignored altogether.
- Brain stem and cerebellum area tend to be noisy in MRI images (possibly due to scanning), therefore, it is difficult to detect the lesions in these areas. One spot looking like a lesion may be just a noise.

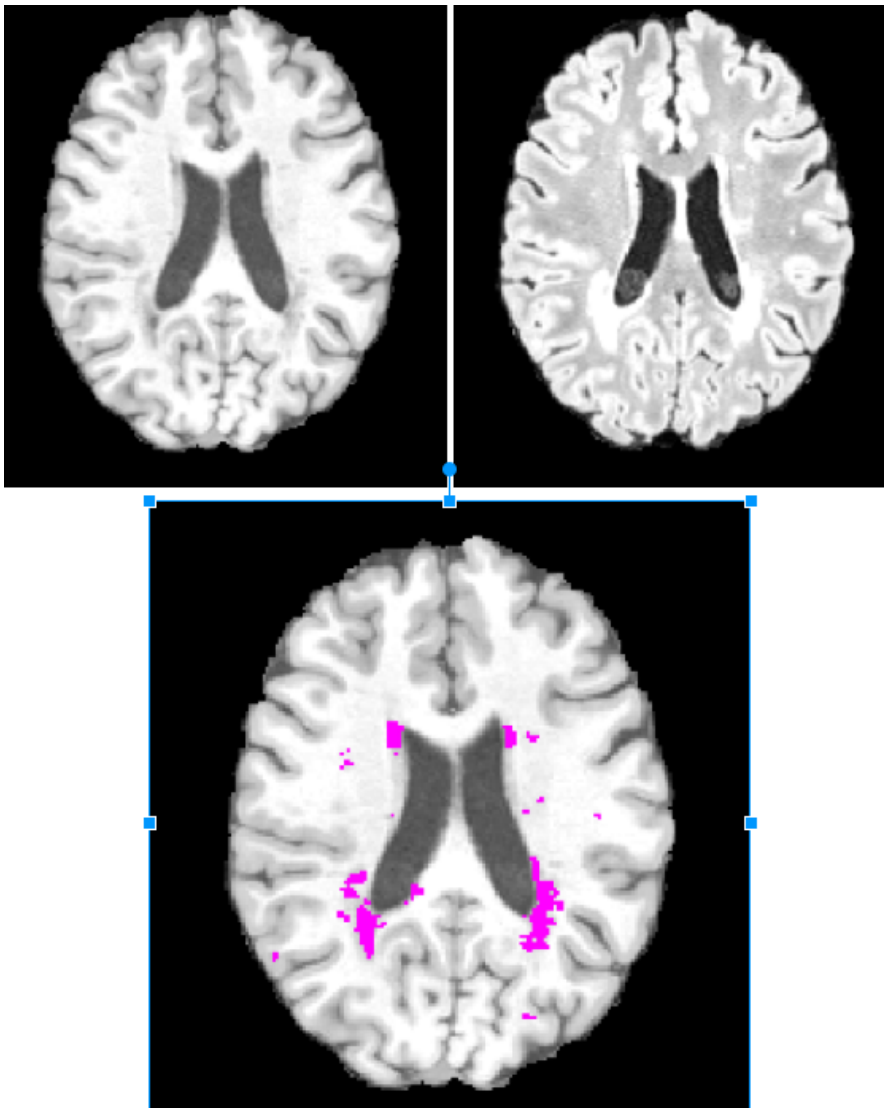

Figure 2: T1, T2 and Manual MS Segmentation

2.4 Deep Learning Techniques

In this section we review the Deep Learning techniques commonly used, namely Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Auto-encoders and Restricted Boltzmann Machines (RBMs), which constitute the backbone of the most deep learning implementations in neuroimaging.

- **Convolutional Neural Networks (CNNs):**

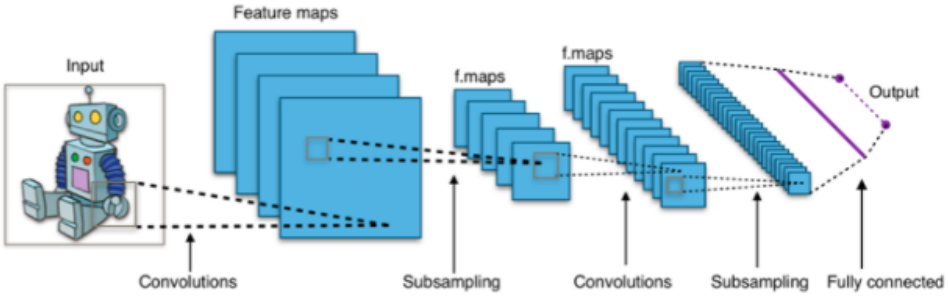

Figure 3: A typical CNN Architecture [24]

Convolutional neural networks are a special type of feed-forward neural networks inspired by human visual system. These networks have one or more convolution layers (typically as the first layers) in addition to conventional fully connected layers. These convolution layers use special convolution filters, also called kernels, which are linear operations applied to the whole data in a region-by-region fashion. The objective of using these kernels is to extract useful features from the data to be able to use them in later stages of the network (see Figure 3).

Convolutional neural networks are mainly applied to data that has a grid-like topology, such as time-series data or images. The motivation to use CNNs for such data is that the same specific convolution is applied to every region throughout the whole data so that a specific feature can be detected regardless of its position. In other words, if we are looking for a specific pattern in a time-series data or an image, this pattern can occur anywhere in the whole data and we typically do not care about its location. This repeated application of a kernel in a sliding-window fashion enables invariance to translation for a classification problem.

One other advantage of the repeated application of the kernel is that the amount of parameters to optimize is significantly decreased since the same kernel is shared among every region (also called parameter sharing). One positive consequence of this is sparse interactions, which means one neuron in one layer is only connected to some of the neurons in the next layer. These two features, namely parameter sharing and sparse connectivity, bring the advantage of less memory requirement and computational efficiency.

In a typical convolutional layer, regional data is processed by three operations in a sequence. The first is the linear kernel we discussed above. The second is a non-linear activation function, such as rectified linear unit, applied to the result of the first linear operation. The third is called pooling (subsampling), which is applied to the result of non-linear activations of neighboring regions, which produces a single result taking a summary statistic for that section such as taking the max or the average. Pooling helps to make the classification problem invariant to small translations of a feature. As can be seen, first through the kernel and second through the pooling operation, the input is substantially reduced when we come to the fully connected

layer. This is a very big advantage from a computational point of view because the following layers will be fully connected. At this point, hopefully independent and explanatory features have been extracted for better and faster classification for the task at hand. From this layer on everything works as in a standard feed-forward fully connected neural network with the extracted features as the inputs.

The convolutions can also be seen as a feature extraction process and can be applied in a succession of several convolutional layers, in which case features, increasing in complexity and abstraction, are extracted in every layer. As an example, for human detection, a kernel for edge-detection, a second kernel for corner detection and a third kernel for body parts detection can be used in succession. The important point here is the information necessary for next-layers should not be discarded in previous layers.

The training of CNNs can be made with the standard stochastic gradient descent algorithm. The kernels can be learned in a supervised fashion during the training of the whole network. However, this is computationally quite expensive. Therefore, other methods like learning the kernels in an unsupervised way or hand-designing before training them may be preferred. [15] [24] [20] [21]

- **Recurrent neural Networks (RNNs):**

A recurrent neural network is a computational graph containing recurrent connections to allow the present value of a variable to have an effect on its future value. It is applied to sequential data mainly to predict later values from former ones. As an example, it may be used for statistical language modeling to predict the next word given the previous ones. RNNs are mainly used in speech recognition, handwriting recognition, financial time-series data, generating image descriptions, etc.

RNNs are represented by recurrent graphs where some neurons have recurrent connections to themselves or to previous neurons indicating that the current value of that neuron affects the future value of that neuron or previous neurons. The recurrent connections can occur from an output neuron to itself, from an output neuron to a previous hidden neuron or from a hidden neuron to itself.

The recurrent graph can be unfolded (See Figure 4) to have an explicit representation of the whole graph in order to have separate variables for each time step. Although the recurrent graph form is more succinct, the unfolded form helps to see clearly the information flow through the graph. In the unfolded graph, the parameters are shared between the repeated parts.

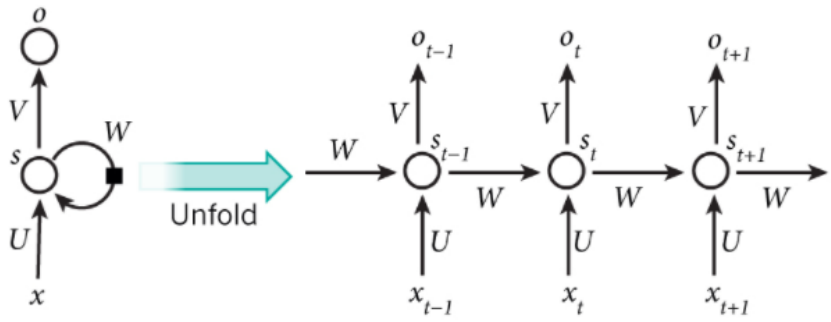

Figure 4: Recurrent and unfolded form of a RNN [18]

Unrolling the recurrent network provides another advantage such that training of the whole network can be done with the back-propagation algorithm. This usage of back-propagation in unrolled recurrent graphs is called back-propagation through time (BPTT). BPTT differs from the standard back-propagation algorithm. Since the parameters are shared through all the time steps, the current gradient calculation depends not only on the calculations in the current step but also on previous steps, which makes it necessary to sum up all the previous gradients. For this reason, RNNs having long temporal dependencies suffer from vanishing/exploding gradient problem and therefore standard RNNs are not suitable for such long-term dependencies. The RNNs having only output-to-hidden connections can also be trained using a special technique called teacher forcing. Other methods to train RNNs include Real Time Recurrent Learning, Kalman Filters and Breeder Genetic Algorithms.

Other important variations of RNNs are bidirectional RNNs, which allow future values to affect present values; and LSTMs (Long-short term memory), which have special gates for neurons to allow the passage or forgetting of the past accumulated information and do not suffer from vanishing/exploding gradient problem. [15] [18] [23] [20] [21]

- **Auto-encoders:**

Auto-encoders are a special type of feed-forward networks that produces an output that is approximately the same as the input. This idea may seem useless at first but it provides useful information about the data. The network uses two functions, an encoder and a decoder function. The encoder function takes the input and produces an internal representation, possibly less dimensional, of the input. From this representation the decoder function should be able to produce an output very similar to the original input. See Figure 5.

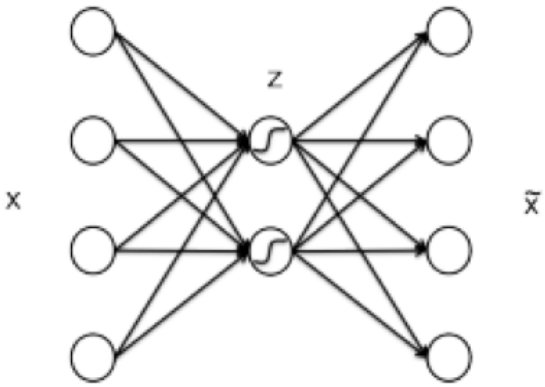

Figure 5: A typical autoencoder [21]

The internal representation contains important information about the input and this process of encoding may be seen as feature extraction/selection. If the encoder and the decoder functions are linear, this process could be likened to PCA. If they are non-linear, a stronger non-linear version of PCA may be obtained. The network can be trained by gradient-descent through back-propagation or recirculation by minimizing the reconstruction error.[15] [20] [21]

Auto-encoders can be made deep adding hidden layers. Deep auto-encoders have been experimentally shown to yield much better results than shallow auto-encoders. Auto-encoders can also be a part of other deep architectures, mainly for the objective of obtaining a good internal representation of the input. The advantage of such architectures is the auto-encoder layers can be trained independently in an unsupervised fashion with e technique called greedy layer-wise unsupervised pre-training. This technique enables the training of very deep and large neural networks with the layer-wise independent training and provides two advantages for the supervised training, a better representation of the data and a good initialization of the parameters.

Denoising auto-encoders are a special type of auto-encoders in which the input is intentionally corrupted before being fed to the network but the network is trained in a fashion to obtain the non-corrupted version of the input. [15] [20] [21]

- **Restricted Boltzmann Machines (RBMs):**

RBMs are bipartite neural networks consisting of a visible layer and a hidden layer. The visible layer contains the observed variables, x; the hidden layer contains the latent variables, h, which give rise to the observed variables. The only connections are between visible units and hidden units. RBMs are made to learn to reconstruct the observed data through hidden variables in an unsupervised fashion with several backward and forward passes between the visible and the hidden unit (see Figure 6). The aim of RBMs is to learn the joint distributions of observed and hidden variables, p(x, h). Joint probabilities are specified by a special energy function, which is designed to have a low value for plausible configurations. In unsupervised training of RBMs an algorithm called Contrastive Divergence, which is based on Gibbs Sampling, is used to obtain the gradient of the log-likelihood.

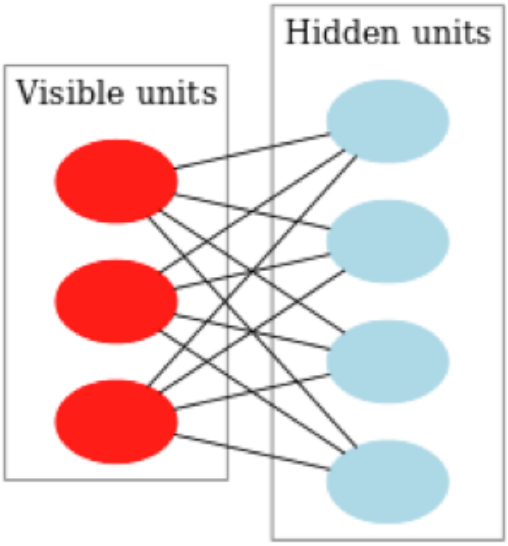

Figure 6: RBM with 3 visible units and 4 hidden units [25]

RBMs can be used in deep neural network architectures as an initial feature extraction process and such networks are generally called deep belief networks. They can be stacked together to benefit from a hierarchy of features getting more complex and abstract in every RBM layer. In such networks, RBM layers are trained in a greedy layer-wised unsupervised fashion, independent of each other. [15] [25] [20] [21]

2.5 Deep Learning Techniques in MS Lesion Segmentation

In the literature of automatic MS Segmentation with machine learning, some methods use supervised approaches with hand-crafted features or learned representations and some use unsupervised methods like clustering which aim to detect lesion voxels as outliers. Supervised models need manually or semi-automatically annotated training images, while unsupervised models do not. The examples to supervised models used in MS segmentation tasks are k-nearest neighbours, artificial neural networks, random decision forests, bayesian frameworks, random forests etc.[7]. Examples to unsupervised models are fuzzy c-means or gaussian mixture models with expectation maximization(EM)[7]. Unsupervised models suffer from non-uniformity in the image intensities and lesion intensities since this variability cannot be captured by a single global model [13]. In this respect supervised methods present an advantage, potentially being able to capture this variability with the appropriate choice of training set or features.

Recently, the interest in machine learning and especially deep learning methods is on the rise due to impressive results obtained in Computer Vision. Due to these new results and the increasing amount of data available in neuroimaging, there is growing research towards developing automatic methods based on these techniques that are reliable, reproducible and efficient.

Previous work on MS lesion segmentation have been developed using voxel-by-voxel classification (lesion vs. normal) and is done on 2D/3D patches centered on the voxel of interest to obtain a complete segmentation of the whole brain. In some methods, in addition to the local context, global context is also provided to the network to give more information about the nature of a voxel.

Convolutional Neural Networks (CNNs) are commonly used as part of the architecture due to their strong feature extraction capabilities. Restricted Boltzmann Machines (RBMs) / Auto-encoders are also exploited to obtain a good initialization of the network, which might affect the ultimate performance.

One example to deep learning methods applied applied to MS segmentation problem is a voxel-wise classifier with a 3 layer convolutional neural network (CNN) used in [8]. According to this architecture, multiple channels of 3D patches of are extracted from MRI images of modality T1, T2 and FLAIR and fed to a CNN. The network has two convolutional layers and a fully-connected network with average pooling layers in-between CNN layers (see Figure 7). For efficient training, sub-sampling methods and sparse convolutions are used. This implementation won the 2015 MS Challenge. According to the authors, dice scores comparable to inter-rater results were obtained. The main advantage of this implementation is its relative simplicity to other CNN architectures. The main drawback is the segmentation is done voxel-by-voxel, which might mean long training and testing times.

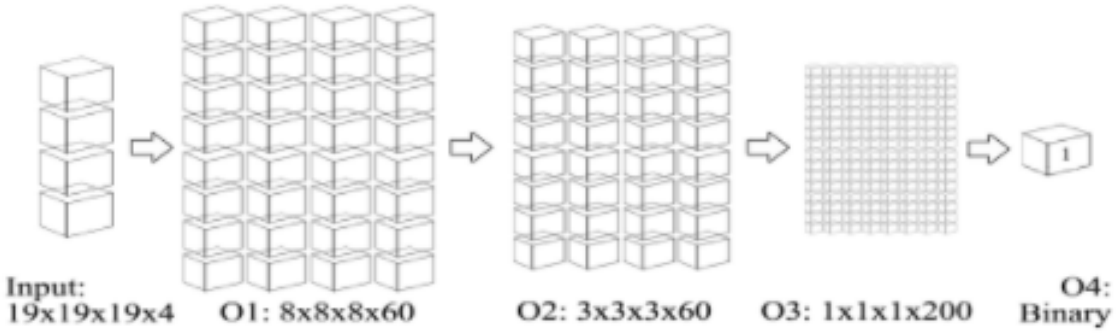

Figure 7: CNN on MS Lesion segmentation [8]

Another example of a CNN solution with 3D patches is a 3 layer convolutional neural network(CNN) presented in [9]. The network consists of 3 layers, the input layer of voxels, one convolutional layer that extracts features for each voxel, one deconvolutional layer which predicts the class of the voxel using previously extracted features. This special type of CNN, consisting of convolutional layers followed by deconvolutional layers is called Convolutional Encoder Network (CEN) (see Figure 8). During training entire MRI volumes as samples are used instead of patch-based training. The entire network is trained all at once, therefore, the learning process is guided by segmentation performance. The proposed architecture is similar to a convolutional auto-encoder. A special objective function with weighted sensitivity and specificity measures is used to deal with the unbalanced nature of the training dataset, i.e. the error measure is a weighted average of squared differences for lesion and non-lesion voxels. 3 MRI modalities (T1, T2, FLAIR) are exploited. The use of convolutional and deconvolutional layers allow the segmentation result to be of the same resolutions as the input, which is the whole brain. The main advantage of this implementation is although one training step might be more computationally intensive, the whole training time is reduced, since the whole MRI image is used as one sample in training. Testing time is also significantly lower than a patch-based testing which evaluates one voxel at a time. The main drawback is the complexity of the model, which might lead to overfitting.

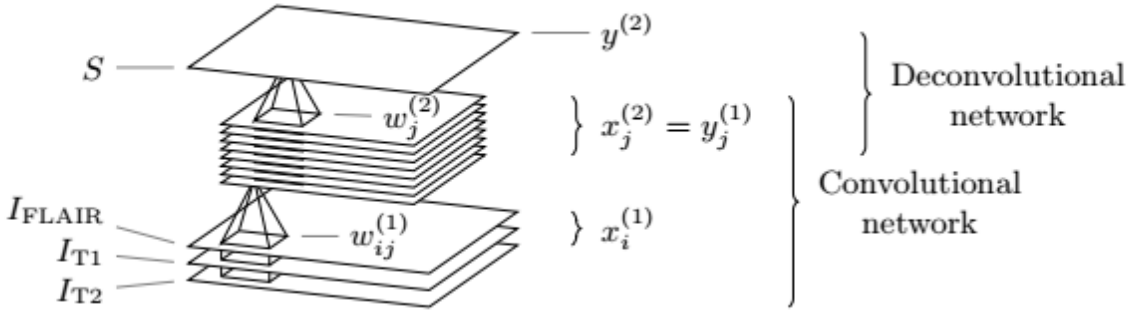

Figure 8: A Basic CEN Structure [9]

In a study one year later, the same authors proposed a similar architecture [10] with convolutional and deconvolutional pathways using the whole brain as input. Initially, a convolutional Restricted Boltzmann Machine (cRBM) is trained, with convolutional layers and pooling layers in between, to pre-train the initial weights and biases. With this initialization, another network is trained consisting of a convolutional pathway followed by a deconvolutional pathway. In the convolutional pathway, there are alternating convolutional layers and pooling layers. The low-level features obtained from the first convolutional layer of this pathway are kept to be used later. The features resulting from the second convolutional layer are fed to the deconvolutional pathway, which consists of alternating deconvolutional and unpooling layers. To the high-level features obtained after the first deconvolutional+unpooling layer are added the low-level features obtained in the previous stage, to obtain a probabilistic lesion mask. The reason to use the deconvolutional layers is to obtain a result the same size as the input, so that the segmentatıon can be done as a whole. The architecture can be seen in Figure 9. As can be seen from Figure 10 the number of convolutional layers has been varied to see the effect of increasing the depth. As the depth of the CEN is increased the segmentation of the bigger lesions gets better due to the increased size of the receptive field. The main advantage to this method is again the segmentation of the whole MRI image in a single-step. Another advantage is the high number of convolutional layers, which may produce better features. The main drawback is the complexity of the model, which might lead to overfitting.

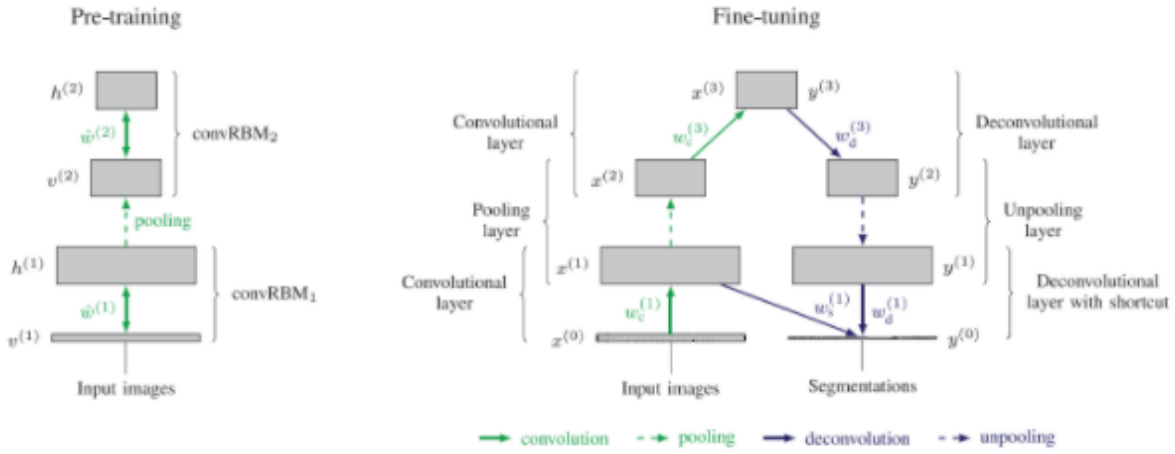

Figure 9: CEN Architecture used in [10]

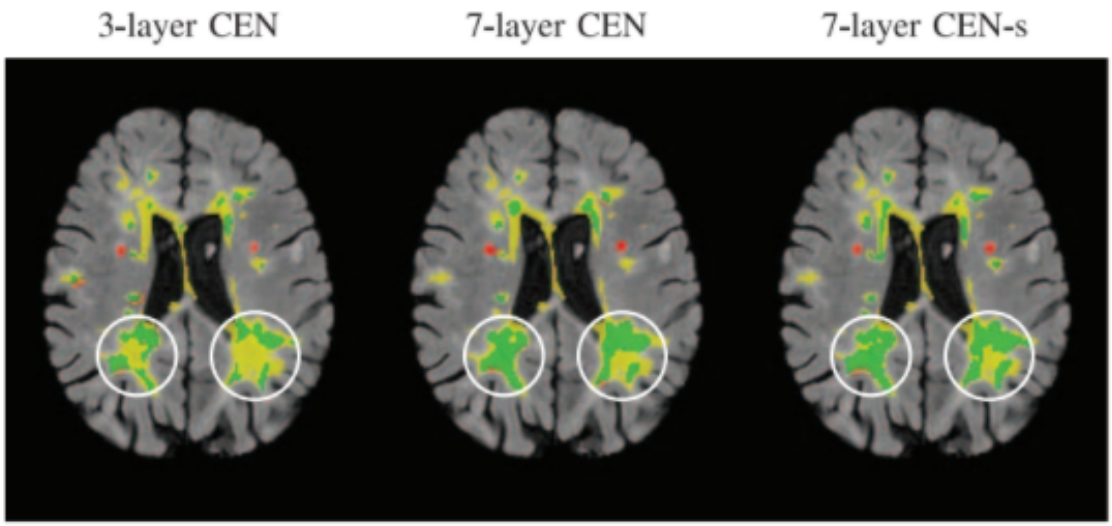

Figure 10: As the depth of the CEN is increased the segmentation of the bigger lesions gets better [10]

In [11] authors propose to integrate the anatomical location in CNN architectures to increase the accuracy, since the probability of a lesion occurring in certain regions are higher. Different from the previous architectures, 2D patches are used instead of 3D patches. Two different ways of feeding the location information to the network is implemented, the first by considering multi-scale patches and the second by adding the location features explicitly. 2D patches are sampled from 500 patients, selecting the positive and negative samples in a random fashion. The patches are selected in 3 different sizes, 32*32, 64*64 and 128*128. The first architecture they propose is a CNN with 4 convolutional layers followed by 3 fully connected layers. This architecture uses only patches of size 32*32. In a second architecture, they consider all the different scales as the input and down-sampling the 64*64 and 128*128 patches into 32*32 patches, they obtain 3-channels for each input. This architecture takes into account both the local information with the smaller patches and the global information with the bigger patches. The second architecture proposed in the same study trains 3 different convolutional branches for each input scale and then fuse them with fully-connected layers. Alternatively to this architecture where global information is taken into account, they propose also to use only the smallest patches and add explicit location features (x, y, z) at the fully-connected layer. They find that incorporating the spatial location information improves the segmentation results. The advantage to these approaches is the explicit or implicit location information added to the network, which might lead to improved accuracies. Another advantage is the reduced training and testing times due to 2D patches. However, usage of 2D patches also create a drawback, because some valuable information would be lost due to the usage of 2D patches instead of 3D patches. See figure 11 for the proposed architectures.

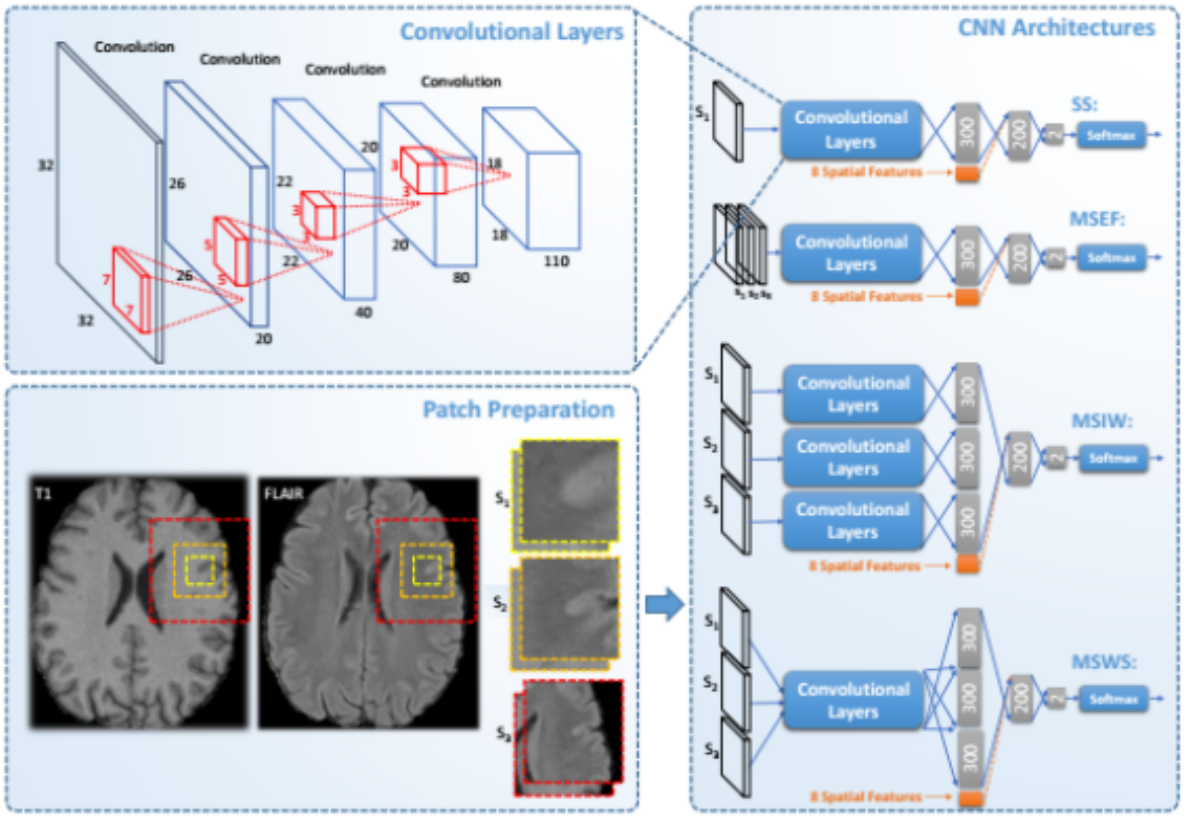

Figure 11: Different proposed architectures in the study by [11]

An alternative to the implementations suggested above could be the cascade based approach in [12]. This implementation is based on a cascade based training with two stages, using 3D patches of size 11*11*11. For the first training step a CNN with two convolutional layers each followed by max-pooling layer and a fully-connected layer followed by a soft-max layer at the end is used. The novel idea in this paper is that first a relatively "coarser" CNN is trained, with the explained architecture, which is supposed to find the "candidate" lesions. Later, using the wrongly classified negative examples from the application of the first CNN (together with all the positive ones), a "finer" CNN with the same architecture is trained, which is more sensitive, because trained with harder cases. This two-stage technique is applied during testing as well, during which the first network is used to find the "candidate" lesions, discarding the low-probability results. These candidate lesions are later fed to the second network to make finer predictions. See figure 12. This idea might also be used to select the training samples in a more smart fashion but train only one network in the end to be used during testing, which might decrease the testing time. The main advantage of this approach is the two stage evaluation, which might increase the accuracy results. The main drawback is high training and testing times.

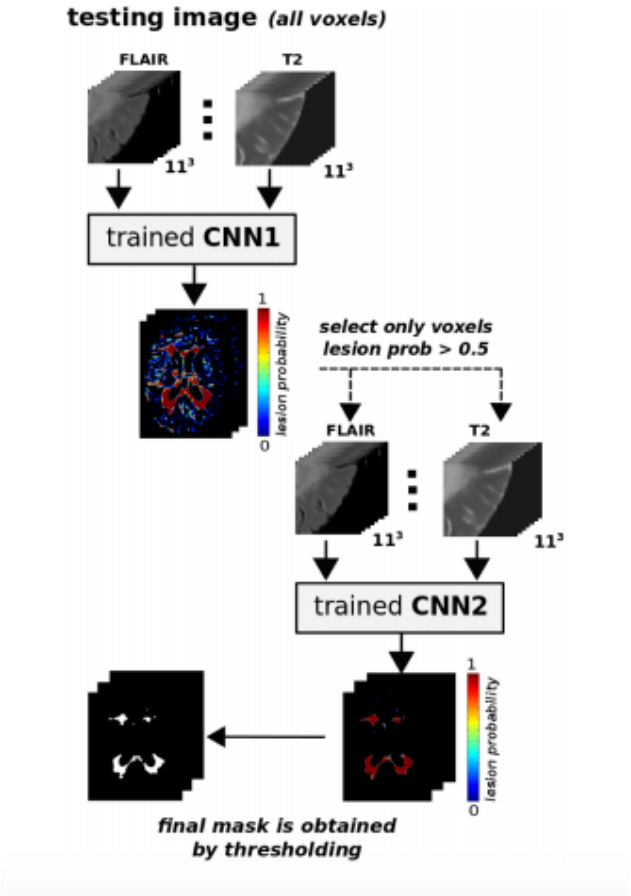

Figure 12: Testing in the proposed architecture by [12]

3 Methodology

In this section, we present our strategies for sub-sampling the training samples, determining the architecture of the deep learning methods we use and designing different approaches to improve our results.

3.1 Data and Pre-processing

We will be using the T1 modality, T2 modality, tissue segmentation and lesion-mask in NIfTI-1 format. The voxel resolution for the images is 0.86mm * 0.86mm * 0.86mm and the image size is 208 * 256 * 256. As a preprocessing of the MRI images we apply skull stripping, bias-correction, tissue-segmentation and co-registration. Additionally we apply 0-mean unit-variance normalization to the data. To know more about pre-processing see [2] and [3].

3.2 Technical Specifications

We will use an EC2 instance of type p2.xlarge of Amazon Web Services. This is a cloud service of Amazon with GPU that provides high computational power for computationally intensive processes such as deep learning. It also comes with an execution environment that contains deep learning frameworks such as Tensorflow, Caffe, Theano, Torch, etc. We will use Python as a programming language and Tensorflow as a deep learning framework to implement and run our deep learning algorithms. We will also use cloud storage provided by Amazon to store our training samples. We will read the neuroimaging files in the NIfTI-1 format with the nibabel library of python.

3.3 Sub-sampling Strategy

Since we will be doing voxel-by-voxel classification, a sample is a 3D patch centered on the voxel of interest. A positive sample is such a patch centered on a lesion voxel, while a negative sample is a patch centered on a non-lesion voxel. The data comes with a big imbalance of positive-negative samples, negative samples greatly outnumbering the positive ones, because the lesion regions generally make up a very small proportion of the whole brain. To overcome this problem, we take all the positive samples and select as many negative samples in different ways. The approaches we selected for negative sample selection are random sampling and sampling around the lesions. By random sampling we mean choosing negative samples randomly throughout the brain, without taking into account its location. By sampling around lesions we mean taking negative samples very close to the lesion areas. The latter approach produced better results in the initial experiments therefore we kept to this approach in the bigger experiments. This might be due to the fact that in the random sampling method, the selected negative samples are very similar to each other and does not represent the diversity of the negative samples. However, when we select the negative samples around the lesions, we add more variety and harder cases to the training set. In addition, we avoid the negative examples very close to (2-voxels) lesion regions, since these voxels may be in reality lesion voxels although they were not labeled as such by the expert. We select the samples from white and gray matter although the studies so far generally chose their data from only white matter since the probability to have a lesion in the white matter is far greater than having it in the gray matter.

3.4 Patch-based Classification using CNN

We start with the CNN architecture in the study [8] with some changes. According to this architecture we are using a CNN architecture with two convolutional layers and one fully connected layer. The patches are obtained from T1 and T2 images centered on the voxel of interest. The first convolutional layer uses 60 filters of 4*4*4 convolutions. The stride value for this layer is 1, therefore the output size is the same as the input size. This layer is followed by an average pooling layer that takes 2*2*2 patches of the output generated in the previous layer and produces an average

for each patch. The stride value for average pooling is 2; therefore the output number obtained is 1/8th of its original value at this stage. The second convolutional layer consists of 60 filters of 3*3*3 convolutions with a stride value of 1. After this second layer of convolutions we use average pooling of 2*2*2 patches with a stride value of 2. The number of voxels at this stage drops to 1/64th of its original value because of the two pooling layers with strides of 2. The two convolutional layers is followed by a fully connected layer. This layer consists of 200 hidden units that are fully connected to the output of the previous layer. The output layer is a softmax layer of two units, which produces a probability for the two classes, lesion and non-lesion for the center voxel. The class with a higher probability is the decision class produced.

For the convolutional layers and fully connected layer ReLU (Rectified Linear Unit) are used as activation functions. As the output layer softmax layer is used, which produces a probability for each class. To compute gradients and guide the training of the network cross-entropy (negative log likelihood) loss function is used. We use mini-batch gradient descent with a batch size of 128 examples for each training step. To adjust the convergence speed of the algorithm, an adaptive learning method, namely Adam (Adaptive Moment Estimation) is used which adapts the momentum and learning rate through the training. At the convolutional layers, batch normalizations is applied, which is known to lead to faster convergence and which serves as a type of regularization method. We apply dropout to the fully connected layer to add further regularization. In our experiments, we detected that after a value of approximately 60 epochs the training accuracy rate continues to increase slightly but we observe some decrease in the validation accuracy rate. For this reason, to prevent overfitting, we kept the number of epochs as 60 in our training. See Figure 13 from one of the training sessions.

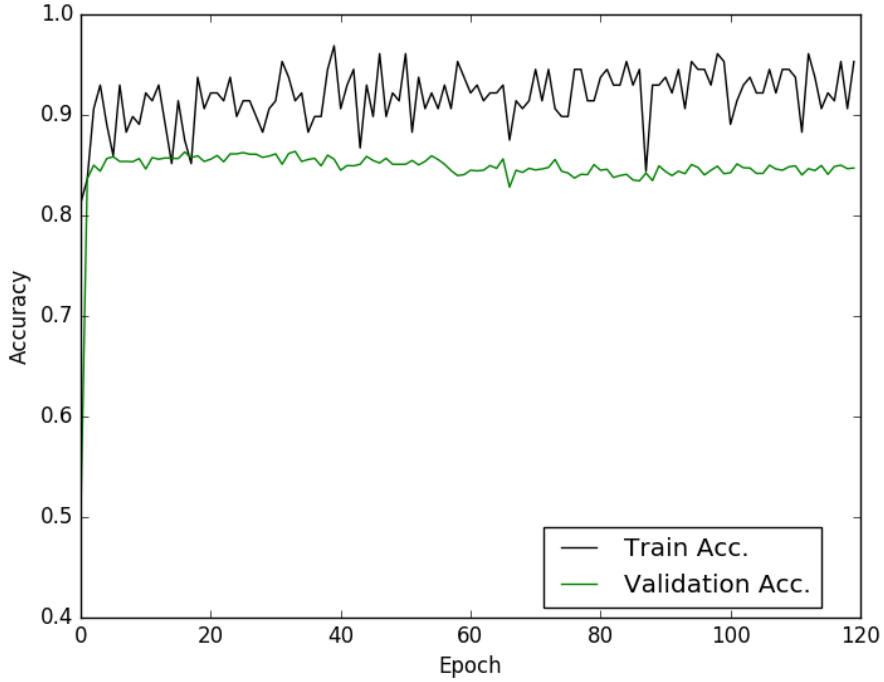

Figure 13: The evolution of Training and Validation Accuracies

In the study [8] authors use 3 modalities T1, T2, and FLAIR, but we use only two, T1 and T2 since we do not have the FLAIR modality for our patients. The patch size they use is 19*19*19 but since our images has nearly half the resolution of the images used in the study we choose our initial patch size as 11*11*11. They consider different modalities as input channels and obtain a

4D data. We chose to start separate branches from different modalities and merge them at the fully-connected layer since in our small experiments we tried both and we saw some improvement with this approach. Also our subsampling method differs from theirs, which was determined based on small experiments we carried out using different approaches and which was explained in the previous section.

In the first approach, we start without adding the location information to the network we described above. Figure 14 illustrates the first approach.

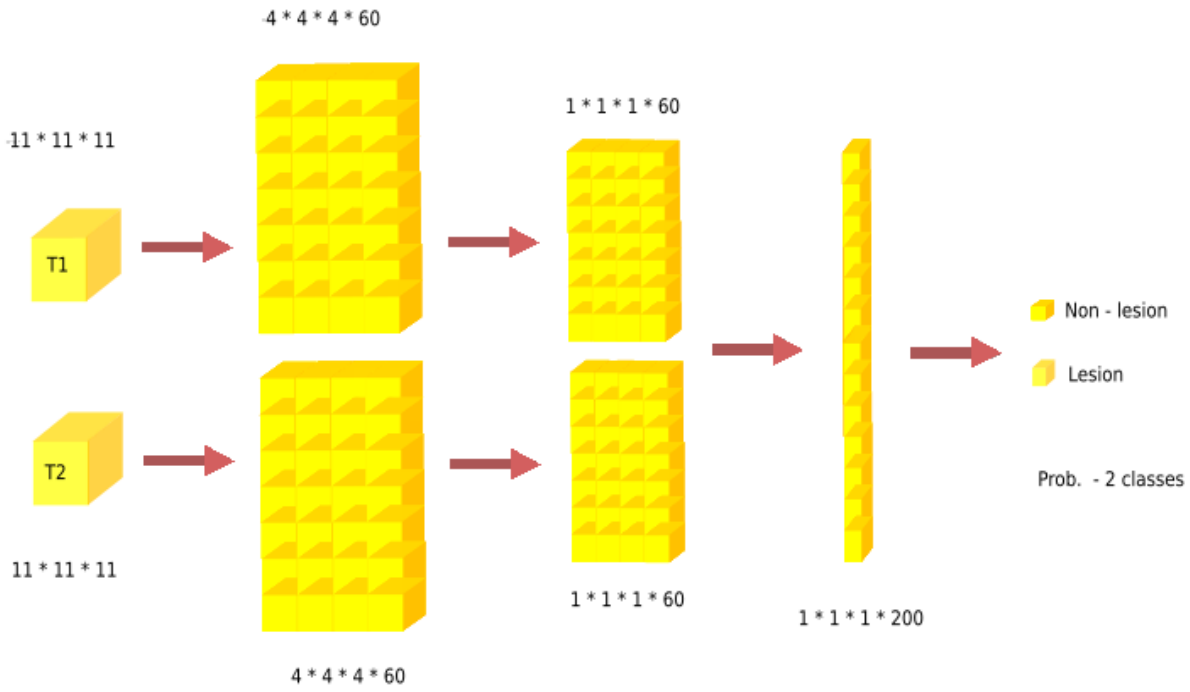

Figure 14: Approach 1: Patch size 11 * 11 * 11 - 2 classes - without location

As the second approach we test if adding the location information to the feature set helps to achieve better results. We think this information might be useful since the probability of having a lesion in certain regions of the brain might show differences. For this approach we store the x,y,z coordinates of each patch, we normalize them and add these features at the fully connected layer. The additional computational cost by adding these 3 features is negligible for training and testing. Figure 15 illustrates the second approach.

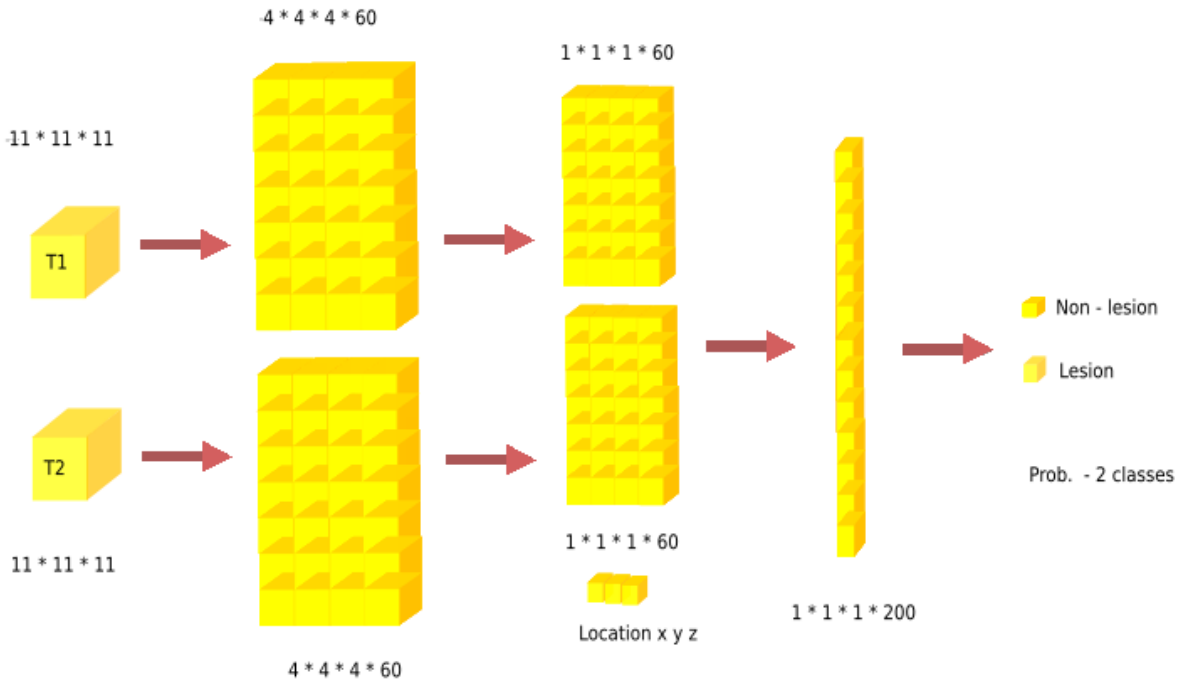

Figure 15: Approach 2: Patch size 11 * 11 * 11 - 2 classes - with location

As a third approach we increase the patch size to see if giving more information about the surrounding region increases the accuracy obtained. We increase the patch size from 11*11*11 to 19*19*19. Note that this increases the number of initial features more than five times. This means increasing the computational cost enormously. We will keep the location information for this approach and further approaches since we see a slight improvement with adding the location information. Figure 16 illustrates the third approach.

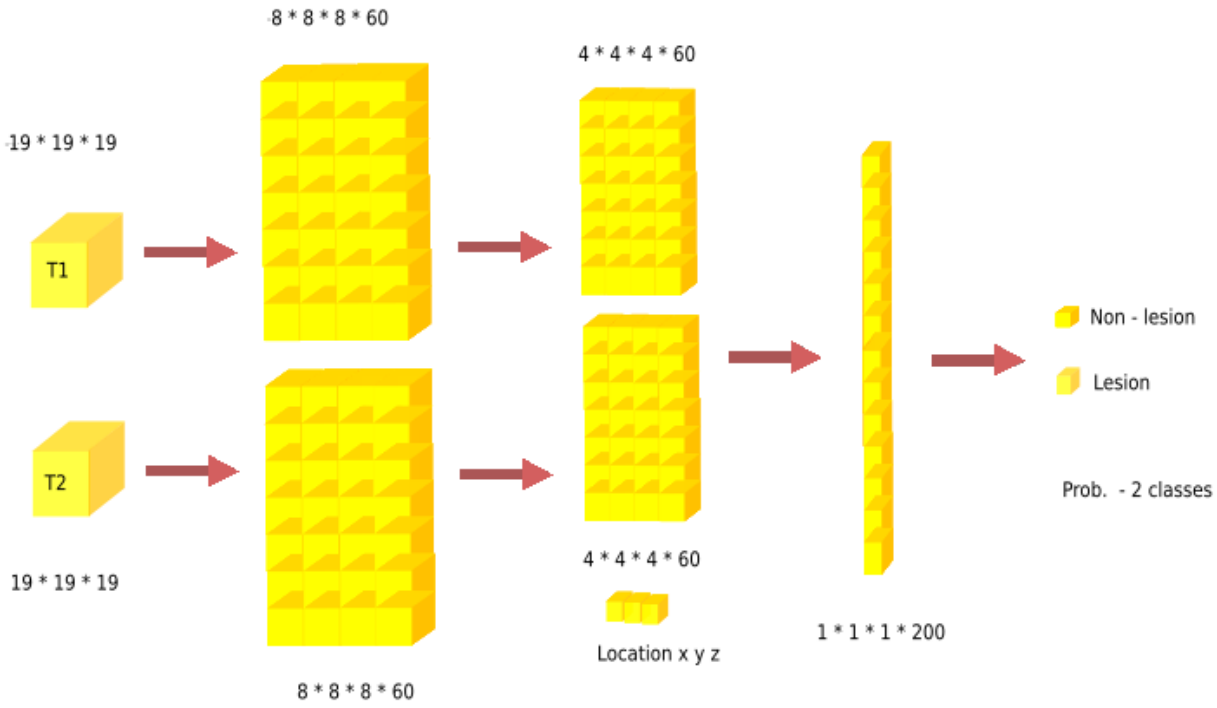

Figure 16: Approach 3: Patch size 19 * 19 * 19 - 2 classes - with location

Up to this point in our approaches, the main problem in the results obtained was the high number of false positives obtained. Even though true negative rates reached 96% and higher levels, the resulting segmentation contained a high number of false positives, even surpassing the number of true lesion voxels, due to the high number of non-lesion voxels in the brain compared to the lesion voxels. This is very undesirable for an automatic segmentation technique since the human expert will need to discard these false positives, which might even make the automatic segmentation useless. From this point on, our efforts will be focused on decreasing these false positives.

As the next and the fourth approach we switch back to 11*11*11 patch size due to its computational advantage. In this approach we implement two CNNs in a cascade fashion as explained in the study of [12]. We first implement a CNN as in the second approach (11*11*11 patch size with location information) and obtain a model. With this model obtain, we segment all the training subjects automatically, which will be used in the selection of training samples in the second stage. For the second model, we use exactly the same architecture as in the first stage but the sampling method will differ. We choose all the lesion voxels as the positive samples, which is the same as in the first stage. As for the negative samples, we choose as many non-lesion voxels in such a fashion that half of this number comes from the false negatives and half comes from the true negatives from the first stage segmentation results. The reason to choose from false negatives is to be able to remove these false negatives in the second stage and the reason to choose from true negatives is to prevent the model to forget the learning from the first model. In the original study negative samples are only selected from the false negatives, which in our case performed far from desired. The

second stage model is trained with the samples selected as just explained. In the testing stage, an initial segmentation is obtained using the first CNN model. The "candidate lesions" obtained from this first model is fed to the second model and a final segmentation is obtained with the resulting positives of the second model. The computational cost to the training of this model is more than twice the cost of a one stage model since it also includes the evaluation of all the training samples. However, once the training is obtained the computation cost for testing does not double since the samples evaluated in the second stage are a very small proportion of the whole sample set. Figure 17 illustrates the fourth approach.

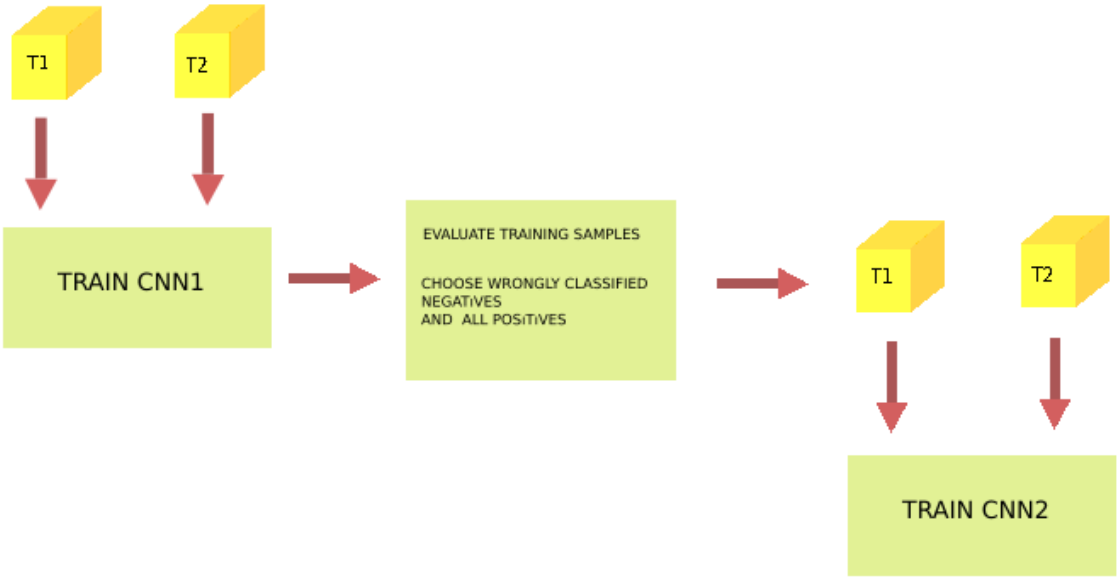

Figure 17: Approach 4: 2 Stage Training - Cascade

With the cascade implementation explained above we gain substantial improvement in the reduction of false positives at the cost of losing also some of the true positives, but the gain was considerably higher than the loss. Although we managed to decrease the number of false positives with this method, the number was still high. In the course of our experiments we realized that there is a difference in the accuracy rates between the region along the border of a lesion and the region far-from a lesion border. This observation brought to our minds to try first a 4-class model first and a 3-class model next.

As the fifth approach, we increased the number of classes from 2 (lesion, non-lesion) to 4 (lesion interior, lesion border, non-lesion border, non-lesion interior) during training to see if we obtain better results with the false positive rate. During testing, we merge the 4 classes back to 2 classes. As a result we saw some improvement with this model compared to the 2-class model. Figure 18 illustrates the fifth approach.

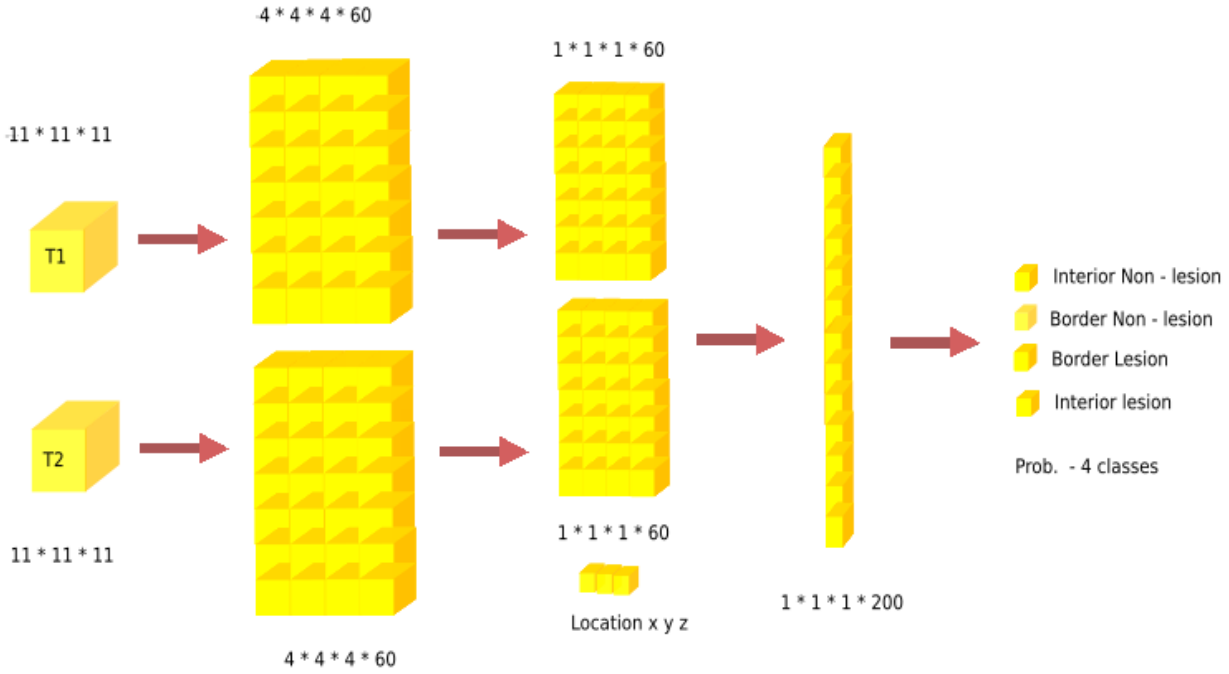

Figure 18: Approach 5: Patch size 11 * 11 * 11 - 4 classes - with location

The improvement obtained with the 4 class model led us further to continue with a 6th approach, in which we considered 3 classes (lesion, border non-lesion, interior non-lesion). Actually, the problem with the 4 class model was that the number of lesion interior voxels was quite low compared to the number of other classes and to balance the training set we had to decrease the number of samples substantially. To prevent this, we thought we could consider lesion voxels as one class, and divide the non-lesion voxels into two classes, border non-lesion, interior non-lesion. Figure 19 illustrates the sixth approach. The improvement obtained with this model was quite surprising, surpassing the results obtained from all the previous models.

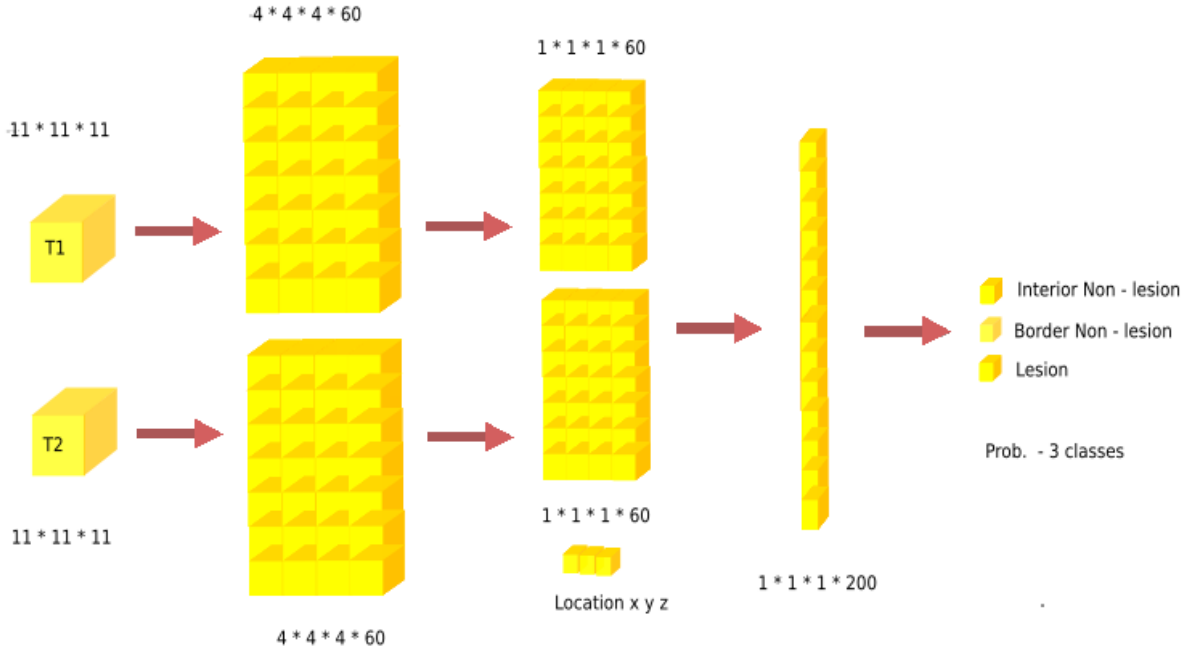

Figure 19: Approach 6: Patch size 11 * 11 * 11 - 3 classes - with location

As a last step, we wanted to improve our results with a cascade implementation of this 3 class model as a 7th approach. This is the exact cascade approached explained before in the 4th approach, but with 3 classes instead of 2. See Figure 17.

The aim of these approaches is to see if we can outperform the commonly used automatic tool, which is called LST and also see the advantages of different approaches. LST [22] is an automatic lesion segmentation tool, which is developed for MS Lesion Segmentation. It is based on Logistic Regression and provides a lesion probability map for the brain using T1 and T2 images.

4 Experiments and Results

4.1 Data

We have T1 and T2 modalities, tissue segmentation and lesion segmentations from 59 subjects. For the distribution of data to train, validation and test, we allocate 45 subjects to train, 5 subjects to validation and 9 subjects to test data. We use validation accuracy to determine the number of epochs with which to train the networks. In the experiments, after 60 epochs the validation accuracy did not improve and even started to drop, for this reason we decided to stick to this number for the training of our networks.

4.2 Validation Measures

For the evaluation of the performance of the models obtained on the test set, we use the following metrics.

Dice similarity coefficient (DSC): DSC is a statistical overlapping measure that measures the similarity between two segmentations. We consider this as the most important statistic in our set of metrics to evaluate the performance of a segmentation method. This measure is between 0 and 1, 0 meaning no similarity and 1 meaning a perfect match between segmentations. We express it as a percentage.

$$Dice = \frac{2 * TP}{2 * TP + FP + FN} * 100 \tag{1}$$

True Positive Rate (TPR): TPR is the percentage of the lesion voxels with respect to the total ground truth lesion voxels, which is also called the sensitivity. This measure is between 0 and 1, and the higher the better although it has to be considered together with other measures for the quality of the segmentation. We express it as a percentage.

$$TPR : \frac{TP}{\#lesion\_voxels\_GT} * 100 \tag{2}$$

False Discovery Rate (FDR): FDR is the percentage of false positive voxels in the output segmentation performed by the method. The measure is between 0 and 1, and low values are desired. We express it as a percentage.

$$FDR : \frac{FP}{\#lesion\_voxels\_found} * 100 \tag{3}$$

Positive Predictive Value (PPV): PPV is the precision rate of the method segmentation, meaning the ratio of the true positive voxels with respect to the output segmentation. The measure is between 0 and 1, and high values are desired. Together with FDR they add up to 1. We express it as a percentage.

$$PPV : \frac{TP}{\#lesion\_voxels\_found} * 100 \tag{4}$$

Volume Difference (VD): Volume Difference is the percentage of the absolute difference between the ground truth lesion volume and the volume of the lesions found by the automatic model with respect to the ground truth lesion volume. This measure does not give information about the overlap of the two segmentations but gives an idea about the relative volumes. The minimum and the ideal value for this measure is 0 but there is no maximum for this measure. 0

value means the lesion volumes in the method segmentation and the ground truth are the same in size, although it might mot mean a perfect overlap. We express it as a percentage.

$$VD = \frac{\|\#lesion\_voxels\_found - \#lesion\_voxels\_GT\|}{\#lesion\_voxels\_GT} * 100 \tag{5}$$

#CC lesion GT: The number of connected components in the ground truth segmentation

#CC lesion found: The number of connected components found

#CC lesion coincided: The number of connected components in the ground truth that has some overlap with the connected components in the method segmentation

where,

TP: True Positives
FP: False Positives
FN: False Negatives
#lesion voxels GT: The number of lesion voxels in the ground truth
#lesion voxels found: The number of lesion voxels found in the method segmentation
Connected Components (CC): One connected component can be explained as a group of lesion voxels connected with each other. It can be as small as 1 voxel or as big as thousands of voxels.

4.3 Results

4.3.1 Experiment 1 - Patch size 11 * 11 * 11 - Location not included

In this experiment we choose the patch size as 11 * 11 * 11 and do not include the location information. As can be seen from the table although the true positive rates are quite high, the number of false positives far outnumber the number of true positives, which makes the model far from usable. See Table 1.

SUBJECT	DSC%	VD%	TPR%	FDR%	PPV%	#lesion voxels GT	#lesion voxels found	#CC lesion found	#CC lesion coincided	#CC lesion GT
013MSVIS	42.4	204.6	85.7	71.9	28.1	13297	40507	651	123	160
050MSVIS	28.7	306.6	72.7	82.1	17.9	7453	30303	655	109	145
082MSVIS	12.6	1207.7	88.6	93.2	6.8	1450	18961	427	31	41
083MSVIS	54.8	130.9	90.6	60.8	39.2	23150	53456	451	104	151
084MSVIS	7.4	2070.6	83.6	96.1	3.9	994	21576	388	47	65
088MSVIS	5.6	2317.2	73.7	96.9	3.1	1611	38941	714	97	146
090MSVIS	48.7	150.6	85.4	65.9	34.1	16223	40650	579	150	230
091MSVIS	31.6	274.5	74.9	80.0	20.0	11357	42530	715	125	141
092MSVIS	63.0	56.2	80.8	48.3	51.7	28622	44712	472	45	51
Mean +standard deviation	32.8 +- 21.1	746.5 +- 891.2	81.8 +- 6.6	77.2 +- 16.9	22.8 +- 16.9	11573 +- 9873	36848 +- 11170	561.3 +- 128.6	92.3 +- 41.6	125.6 +- 61.4

Table 1: Experiment 1 Results

4.3.2 Experiment 2 - Patch size 11 * 11 * 11 - Location included

In this experiment we choose the patch size as 11 * 11 * 11 and include the location information to the patch at the fully-connected layer. The results on average seem not to have improved, however the fluctuation in terms of standard deviation seems less. Still the main problem is high number of false positives. See Table 2.

SUBJECT	DSC%	VD%	TPR%	FDR%	PPV%	#lesion voxels GT	#lesion voxels found	#CC lesion found	#CC lesion coincided	#CC lesion GT
013MSVIS	44.7	184.9	86.0	69.8	30.2	13297	37882	448	113	160
050MSVIS	36.6	200.5	73.3	75.6	24.4	7453	22398	545	108	145
082MSVIS	15.0	1014.3	91.0	91.8	8.2	1450	16157	428	28	41
083MSVIS	47.3	205.3	95.9	68.6	31.4	23150	70674	354	120	151
084MSVIS	7.4	2024.1	97.4	96.1	3.9	994	21114	340	45	65
088MSVIS	7.9	1602.3	71.2	95.8	4.2	1611	27424	588	90	146
090MSVIS	45.7	190.4	89.3	69.3	30.7	16223	47104	471	148	230
091MSVIS	31.0	331.3	82.4	80.9	19.1	11357	48983	565	128	141
092MSVIS	58.7	99.7	87.9	56.0	44.0	28622	57153	535	47	51
Mean +standard deviation	32.7 +- 18.7	650.3 +- 721.1	86.0 +- 9.1	78.2 +- 14.0	21.8 +- 14.0	11573 +- 9873	38765 +- 18541	474.9 +- 90.3	91.9 +- 42.2	125.6 +- 61.4

Table 2: Experiment 2 Results

4.3.3 Experiment 3 - Patch size 19 * 19 * 19

In this experiment we choose the patch size as 19 * 19 * 19. The training and testing times increase cubically, however we see a considerable improvement in the dice coefficient at the cost of losing some true positives. As a result this is a more acceptable model, since the number of false positives are not as overwhelming. See Table 3.

SUBJECT	DSC%	VD%	TPR%	FDR%	PPV%	#lesion voxels GT	#lesion voxels found	#CC lesion found	#CC lesion coincided	#CC lesion GT
013MSVIS	59.9	23.0	66.8	45.7	54.3	13297	16355	143	58	160
050MSVIS	42.2	32.8	35.3	47.5	52.5	7453	5007	94	42	145
082MSVIS	47.5	120.0	76.0	65.5	34.5	1450	3189	70	13	41
083MSVIS	60.8	78.4	84.6	52.6	47.4	23150	41306	154	84	151
084MSVIS	17.3	525.0	62.7	90.0	10.0	994	6213	121	28	65
088MSVIS	9.8	534.7	36.1	94.3	5.7	1611	10225	234	24	146
090MSVIS	59.1	28.3	67.5	47.4	52.6	16223	20819	137	74	230
091MSVIS	41.9	88.1	60.4	67.9	32.1	11357	21367	267	100	141
092MSVIS	68.0	14.4	72.9	36.2	63.8	28622	32736	145	41	51
Mean +standard deviation	45.2 +- 20.1	160.5 +- 212.3	62.5 +- 16.8	60.8 +- 20.3	39.2 +- 20.3	11573 +- 9873	17468 +- 13073	151.7 +- 62.6	51.6 +- 29.5	125.6 +- 61.4

Table 3: Experiment 3 Results

4.3.4 Experiment 4 - Patch size 11 * 11 * 11 - Cascade with 2 stage models

In this model we choose the patch size as 11 * 11 * 11 and train two models in succession. The second stage model is trained with the false positives found from the application of the first model to the training data. As compared with the single stage models (experiments 1 and 2) the dice score has improved and the number of false positives decreased substantially. Also compared with the 19 * 19 * 19 model, this model obtained similar or slightly better results, with an additional benefit of computational efficiency. See Table 4.

SUBJECT	DSC%	VD%	TPR%	FDR%	PPV%	#lesion voxels GT	#lesion voxels found	#CC lesion found	#CC lesion coincided	#CC lesion GT
013MSVIS	55.3	93.3	81.1	58.0	42.0	13297	25700	116	89	160
050MSVIS	48.2	24.0	53.9	56.5	43.5	7453	9238	73	59	145
082MSVIS	50.2	142.3	85.9	64.6	35.4	1450	3513	51	16	41
083MSVIS	52.9	156.6	94.3	63.3	36.7	23150	59401	130	118	151
084MSVIS	22.9	436.7	73.0	86.4	13.6	994	5335	95	36	65
088MSVIS	18.2	399.8	54.4	89.1	10.9	1611	8051	171	49	146
090MSVIS	56.1	103.8	85.2	58.2	41.8	16223	33056	137	103	230
091MSVIS	37.6	196.3	74.5	74.8	25.2	11357	33650	155	114	141
092MSVIS	66.6	54.8	84.9	45.2	54.8	28622	44318	96	45	51
Mean +standard deviation	45.3 +- 16.0	178.6 +- 145.7	76.4 +- 14.1	66.2 +- 14.5	33.8 +- 14.5	11573 +- 9873	24695 +- 19621	113.8 +- 38.8	69.9 +- 37.0	125.6 +- 61.4

Table 4: Experiment 4 Results

4.3.5 Experiment 5 - Patch size 11 * 11 * 11 - 4 class model

In this experiment we choose the patch size as 11 * 11 * 11 but this time we increase the number of classes from 2 (lesion, non-lesion) to 4 (lesion interior, lesion border, non-lesion border, non-lesion interior). The improvement in the dice coefficient can be seen compared to the 2 class single stage model, due to the decrease in the number of false positives. However, the results are worse than the two-stage 2 class model. See Table 5.

SUBJECT	DSC%	VD%	TPR%	FDR%	PPV%	#lesion voxels GT	#lesion voxels found	#CC lesion found	#CC lesion coincided	#CC lesion GT
013MSVIS	50.7	122.0	81.6	63.2	36.8	13297	29523	249	105	160
050MSVIS	40.1	135.0	67.2	71.4	28.6	7453	17511	318	105	145
082MSVIS	22.4	581.4	87.5	87.2	12.8	1450	9881	208	27	41
083MSVIS	47.5	199.5	94.9	68.3	31.7	23150	69344	237	114	151
084MSVIS	14.2	864.2	75.8	92.1	7.9	994	9584	223	45	65
088MSVIS	14.8	644.4	62.6	91.6	8.4	1611	11993	312	75	146
090MSVIS	51.3	132.9	85.5	63.3	36.7	16223	37782	262	132	230
091MSVIS	34.9	239.4	76.6	77.4	22.6	11357	38544	296	116	141
092MSVIS	60.1	86.1	86.1	53.8	46.2	28622	53275	209	47	51
Mean +standard deviation	37.3 +- 16.9	333.9 +- 285.5	79.8 +- 10.3	74.3 +- 13.7	25.7 +- 13.7	11573 +- 9873	30826 +- 20973	257.1 +- 42.7	85.1 +- 37.6	125.6 +- 61.4

Table 5: Experiment 5 Results

4.3.6 Experiment 6 - Patch size 11 * 11 * 11 - 3 class model

In this experiment we choose the patch size as 11 * 11 * 11 but this time we increase the number of classes from 2 (lesion, non-lesion) to 3 (lesion, non-lesion border, non-lesion interior). The considerable improvement in the dice coefficient can be seen compared to the 2 and 4 class single stage model, due to the substantial decrease in the number of false positives. The results are even considerably better than the two-stage 2 class model. This model is the best model we obtained among the models we obtained so far. Therefore we wanted to apply the two stage architecture with this approach in the next step. See Table 6.

SUBJECT	DSC%	VD%	TPR%	FDR%	PPV%	#lesion voxels GT	#lesion voxels found	#CC lesion found	#CC lesion coincided	#CC lesion GT
013MSVIS	64.6	1.7	65.1	36.0	64.0	13297	13528	184	98	160
050MSVIS	45.6	42.7	35.9	37.4	62.6	7453	4269	156	85	145
082MSVIS	60.4	39.8	72.4	48.2	51.8	1450	2027	70	16	41
083MSVIS	67.0	47.4	82.9	43.7	56.3	23150	34129	149	98	151
084MSVIS	39.4	85.0	56.1	69.7	30.3	994	1839	84	29	65
088MSVIS	32.0	98.2	47.7	75.9	24.1	1611	3193	189	75	146
090MSVIS	65.1	15.3	70.1	39.3	60.7	16223	18712	199	122	230
091MSVIS	47.9	21.6	53.1	56.4	43.6	11357	13807	212	113	141
092MSVIS	71.1	4.9	69.3	27.1	72.9	28622	27214	108	41	51
Mean +- standard deviation	54.8 +- 13.9	39.6 +- 33.8	61.4 +- 14.5	48.2 +- 16.2	51.8 +- 16.2	11573 +- 9873	13190 +- 11721	150.1 +- 51.8	75.2 +- 38.0	125.6 +- 61.4

Table 6: Experiment 6 Results

4.3.7 Experiment 7 - Patch size 11 * 11 * 11 - 3 class model - Cascade with two stages

In this experiment we applied the cascade approach to the best model we previously obtained, which is the 3 class model with patch size 11 * 11 * 11, to improve the results further. As can be seen we obtained further improvement in the dice coefficient. On average, one can note that the number of lesion voxels found are very close to the actual number. See Table 7.

SUBJECT	DSC%	VD%	TPR%	FDR%	PPV%	#lesion voxels GT	#lesion voxels found	#CC lesion found	#CC lesion coincided	#CC lesion GT
013MSVIS	65.9	3.9	64.6	32.7	67.3	13297	12773	143	95	160
050MSVIS	46.4	47.9	35.3	32.3	67.7	7453	3885	114	79	145
082MSVIS	67.3	14.7	72.2	37.0	63.0	1450	1663	15	36	41
083MSVIS	68.3	38.0	81.3	41.1	58.9	23150	31952	103	94	151
084MSVIS	47.2	38.0	56.1	59.3	40.7	994	1372	46	29	65
088MSVIS	37.5	33.3	43.8	67.2	32.8	1611	2148	122	66	146
090MSVIS	65.6	7.9	68.1	36.8	63.2	16223	17502	138	111	230
091MSVIS	48.7	6.8	50.4	52.8	47.2	11357	12131	170	105	141
092MSVIS	70.7	14.7	65.5	23.2	76.8	28622	24410	64	39	51
Mean +standard deviation	57.5 +- 12.4	22.8 +- 16.5	59.7 +- 14.6	42.5 +- 14.3	57.5 +- 14.3	11573 +- 9873	11981 +- 10986	101.7 +- 50.4	72.7 +- 31.5	125.6 +- 61.4

Table 7: Experiment 7 Results

4.3.8 LST Tool - Commonly used method in MS Lesion Segmentation

The performance results of LST [22] on our test data set can be seen in Table 8.

SUBJECT	DSC%	VD%	TPR%	FDR%	PPV%	#lesion vox-els GT	#lesion vox-els found	#CC lesion found	#CC lesion coin-cided	#CC lesion GT
013MSVIS	57.8	4.9	56.3	40.7	59.3	13297	12639	21	32	160
050MSVIS	36.5	44.5	28.4	48.9	51.1	7453	4138	23	33	145
082MSVIS	35.2	181.6	67.2	76.1	23.9	1450	4083	20	10	41
083MSVIS	61.4	32.0	71.2	46.1	53.9	23150	30550	12	43	151
084MSVIS	13.5	291.6	33.3	91.5	8.5	994	3893	34	20	65
088MSVIS	15.1	88.1	21.7	88.5	11.5	1611	3031	39	12	146
090MSVIS	56.0	10.0	58.8	46.5	53.5	16223	17841	18	37	230
091MSVIS	35.8	27.3	40.7	68.0	32.0	11357	14463	38	64	141
092MSVIS	66.7	9.9	63.4	29.7	70.3	28622	25786	23	35	51
Mean +standard deviation	42.0 +- 19.6	76.7 +- 98.0	49.0 +- 18.3	59.6 +- 22.1	40.4 +- 22.1	11573 +- 9873	12936 +- 10231	25.3 +- 9.4	31.8 +- 16.6	125.6 +- 61.4

Table 8: LST Results

4.3.9 Final Comparison

In this section we present the performance comparison between the models we obtained and also with the LST [22] method. Later in this section we discuss the behaviour of the best model we found on real MRI images.

As can be seen from the Table 9 the final models, which are the 3 class models has the best performance in all the measures except the True Positive Rate (TPR). This decrease in the TPR is understandable since as we removed the false positives we also had to sacrificed some true positives although very small in number in comparison. The best model also surpassed the LST [22] method in all the measures including the TPR.

METHOD	DSC%	VD%	TPR%	FDR%	PPV%	#lesion voxels GT	#lesion voxels found	#CC lesion found	#CC lesion coincided	#CC lesion GT
LST [22]	42.0 +- 19.6	76.7 +- 98.0	49.0 +- 18.3	59.6 +- 22.1	40.4 +- 22.1	11573 +- 9873	12936 +- 10231	25.3 +- 9.4	31.8 +- 16.6	125.6 +- 61.4
Patch11-w/o location	32.8 +- 21.1	746.5 +- 891.2	81.8 +- 6.6	77.2 +- 16.9	22.8 +- 16.9	11573 +- 9873	36848 +- 11170	561.3 +- 128.6	92.3 +- 41.6	125.6 +- 61.4
Patch 11- with location	32.7 +- 18.7	650.3 +- 721.1	**86.0 +- 9.1**	78.2 +- 14.0	21.8 +- 14.0	11573 +- 9873	38765 +- 18541	474.9 +- 90.3	91.9 +- 42.2	125.6 +- 61.4
Patch 19- with location	45.2 +- 20.1	160.5 +- 212.3	62.5 +- 16.8	60.8 +- 20.3	39.2 +- 20.3	11573 +- 9873	17468 +- 13073	151.7 +- 62.6	51.6 +- 29.5	125.6 +- 61.4
Patch 11- 2 stages	45.3 +- 16.0	178.6 +- 145.7	76.4 +- 14.1	66.2 +- 14.5	33.8 +- 14.5	11573 +- 9873	24695 +- 19621	113.8 +- 38.8	69.9 +- 37.0	125.6 +- 61.4
Patch 11- 4 class	37.3 +- 16.9	333.9 +- 285.5	79.8 +- 10.3	74.3 +- 13.7	25.7 +- 13.7	11573 +- 9873	30826 +- 20973	257.1 +- 42.7	85.1 +- 37.6	125.6 +- 61.4
Patch 11 - 3 class	54.8 +- 13.9	39.6 +- 33.8	61.4 +- 14.5	48.2 +- 16.2	51.8 +- 16.2	11573 +- 9873	13190 +- 11721	150.1 +- 51.8	75.2 +- 38.0	125.6 +- 61.4
Patch 11 - 3 class 2 stages	**57.5 +- 12.4**	**22.8 +- 16.5**	59.7 +- 14.6	**42.5 +- 14.3**	**57.5 +- 14.3**	11573 +- 9873	11981 +- 10986	101.7 +- 50.4	72.7 +- 31.5	125.6 +- 61.4

Table 9: Final comparison between models

The null hypothesis that there is no significant difference between the best model found and the LST is rejected for all the measures except VD, with a threshold p-value of 0.01. See Table 10 for t and p values. We can say the best model found provides significant improvement over LST with confidence.

measure/sig.	t	p
Dice	-4.186	0.003
VD	1.733	0.121
TPR	4.503	0.002
FPR	4.308	0.003
PPV	-4.308	0.003

Table 10: p and t values from t-test between LST and our best model

With the best model we obtained, we wanted to observe the behaviour of the segmentation algorithm on the MRI images by closely inspecting the result. As far as our human inspection could go, we observed that the model has learned the general pattern quite well. As explained before in the state-of-the-art manual segmentation section, the lesions found are in hypo-intense regions in T1 and hyper-intense regions in T2 and the borders are determined by T1 hypo-intense region, which is the correct behaviour.

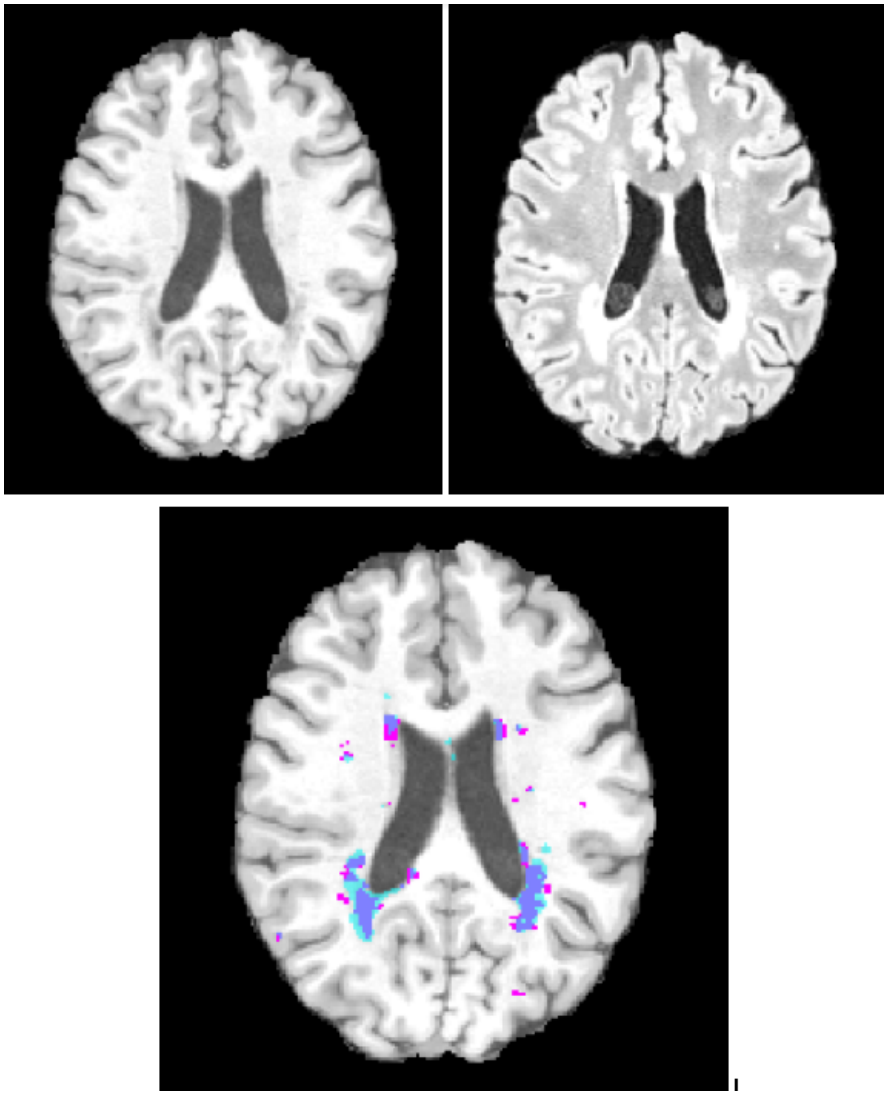

Figure 20: T1, T2 and the segmentation result with the best model

pink:false negative
blue: false positive
purple: true positive

As can be seen from Figure 20, there is an activation around the ground truth segmentation for most of the lesions and there is a high amount of overlap but there are also differences. In our opinion the differences are partly due to the subjective nature of manual MS segmentation or difficulty in obtaining the real borders, even by a human expert. The false positives and false negatives can also be seen from the figure. More qualitative results can be seen from cross-sections of the brains of each test subject in Figures from 24 to 32

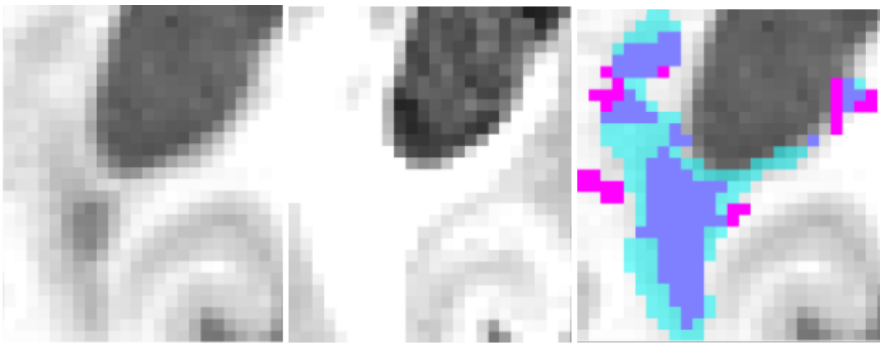

Figure 21: T1, T2 and the segmentation result close up

pink:false negative
blue: false positive
purple: true positive

As can be seen from the figure 21, which is an up-close version of a lesion region, the manual segmentation is more conservative while our model is more generous in designating a region as lesion if there is a corresponding hypo-intensity in T1 and hyper-intensity in T2. Also note that the ground truth segmentation is more jagged and dispersed while the model segmentation is rounder and more connected. This is expected since the probability of two neighbouring voxels being segmented as lesions both is high since they have a very similar neighbourhood. Based on our observations, there were also some cases in the ground-truth that was contrary to the MS lesion definition, which caused some "false" false negatives in our case and this might be explained with some special case, a human error or an error in the alignment process of the images.

The good thing about the model segmentation was that in the majority of the lesions there was some activation on the lesion or very close to the lesion although the overlapping was far from perfect. For instance, if from a cross-section, there was a ground-truth lesion which seemed not detected, when we advanced a few voxels up or down along the perpendicular axis of the cross-section, we observed a lesion detected by the model. This was mainly due to the difficulty in determining the real borders of a lesion.

From our observations, the model seems to capture the hypo-intensity in T1 and hyper-intensity in T2 technically but misses some of the intuitions, domain knowledge or the subjectivity of the human expert. This is expected since the model was trained with MRI's that were segmented by different people or at different times, which may affect the final ground truth segmentation (based on the judgment or the knowledge of the expert) and therefore the final model obtained.

Another observation we made was the number of detected lesions (or lesion voxels) increased with the number of ground truth lesions (or lesion voxels). This means the segmentation result of the model is indicative of the lesion load. This can be seen from the Figures 22 and 23. This information can be used to see the progression of the lesions in an MS patient.

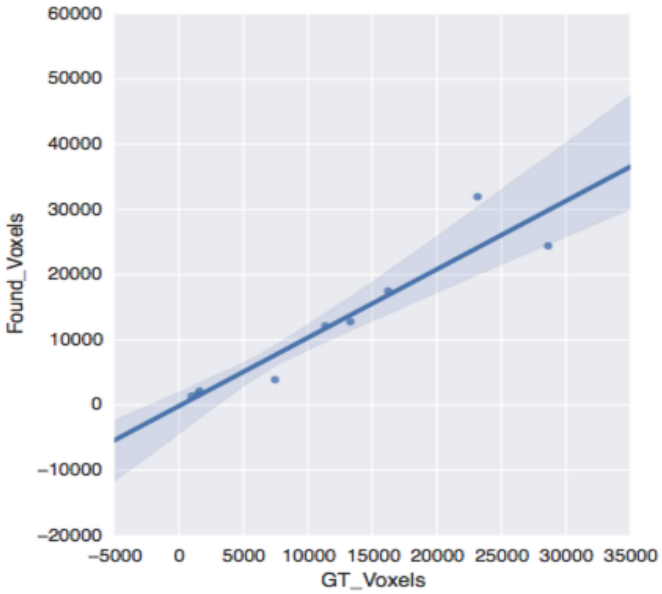

Figure 22: The number of gound truth lesions vs. the number of lesions found

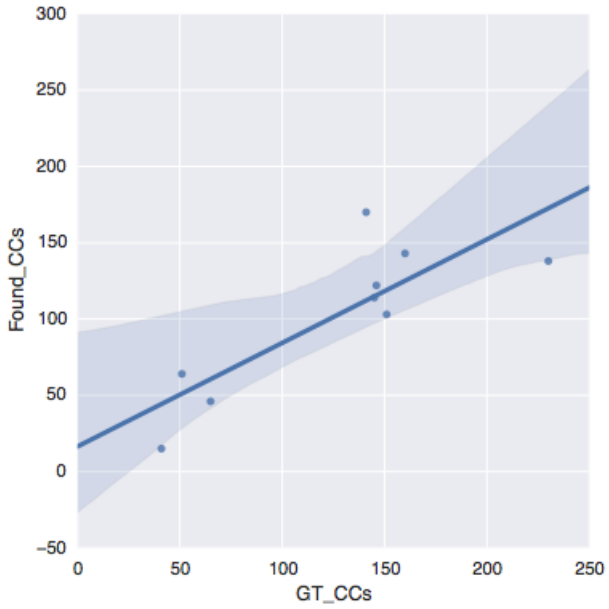

Figure 23: The number of gound truth CC's vs. the number of CC's found

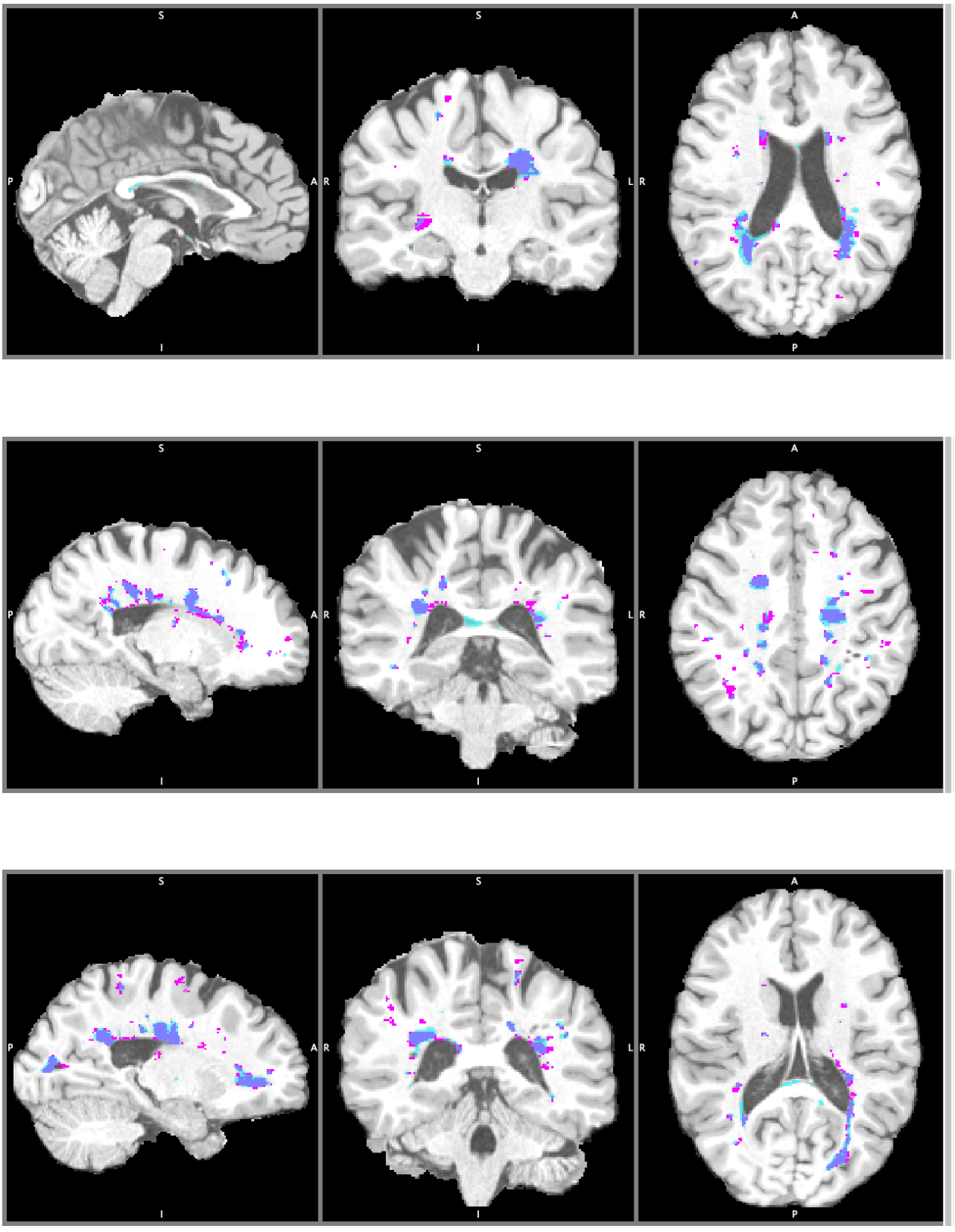

Figure 24: Qualitative Results for Subject 013MSVIS

pink:false negative
blue: false positive
purple: true positive

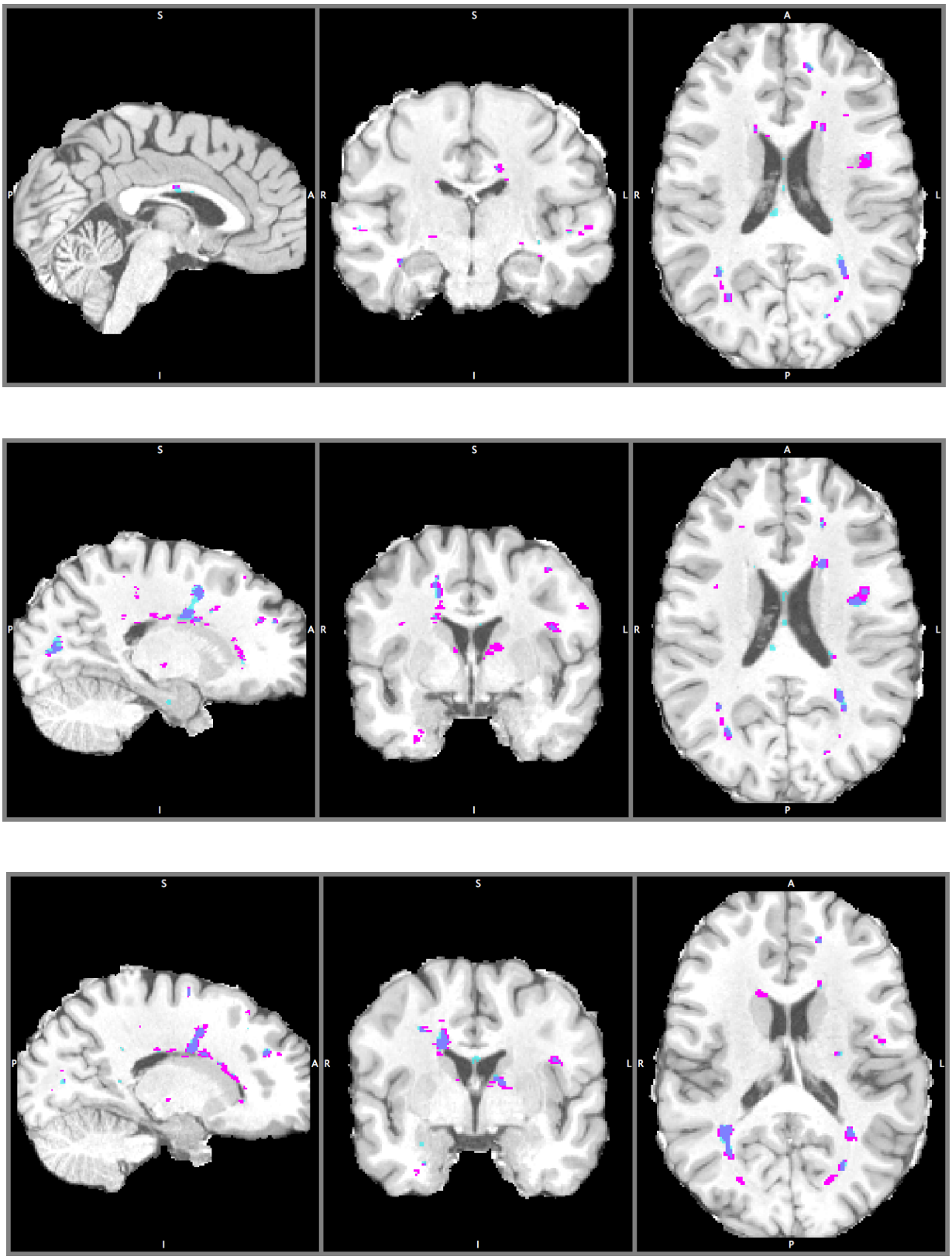

Figure 25: Qualitative Results for Subject 050MSVIS

pink:false negative
blue: false positive
purple: true positive

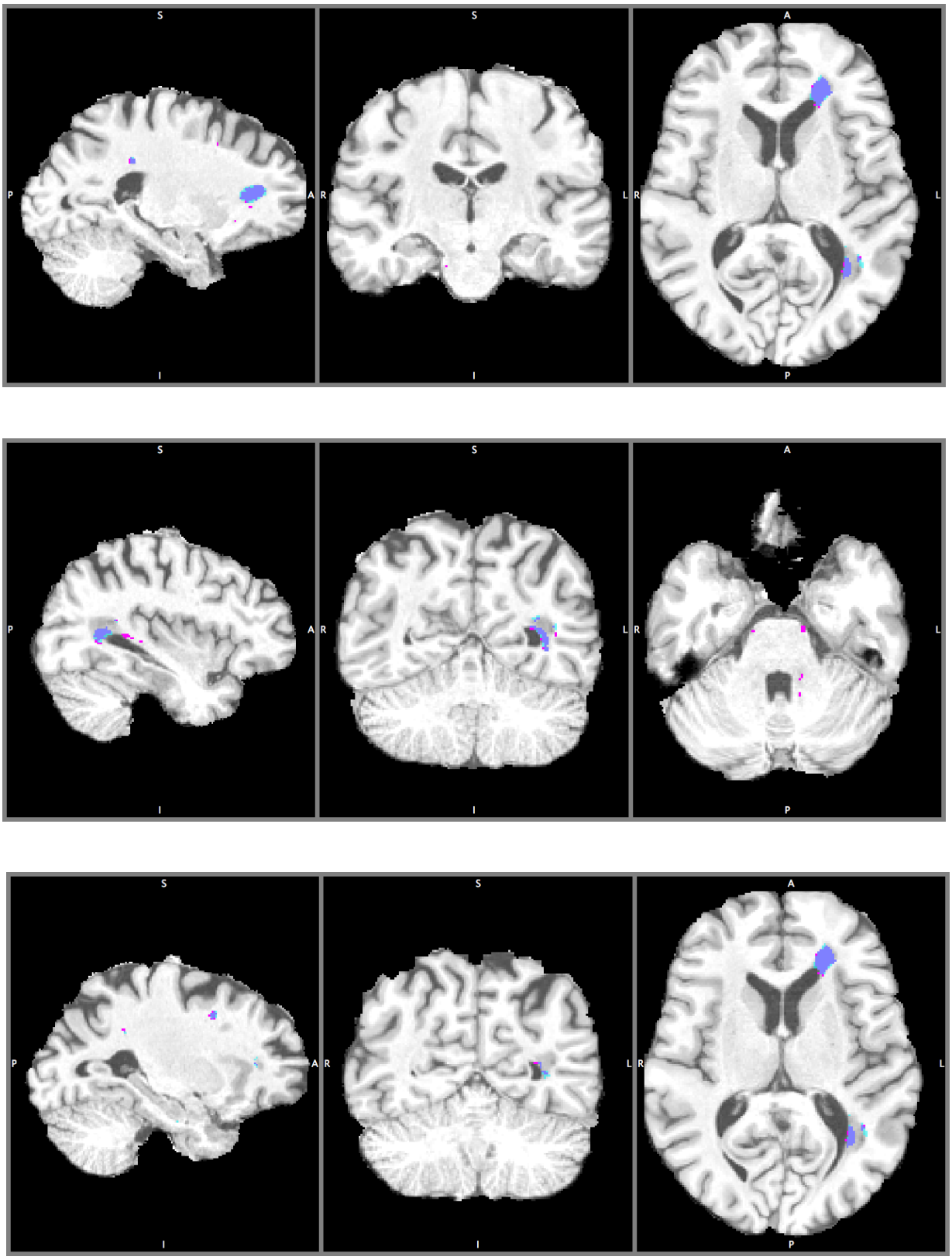

Figure 26: Qualitative Results for Subject 082MSVIS

pink:false negative
blue: false positive
purple: true positive

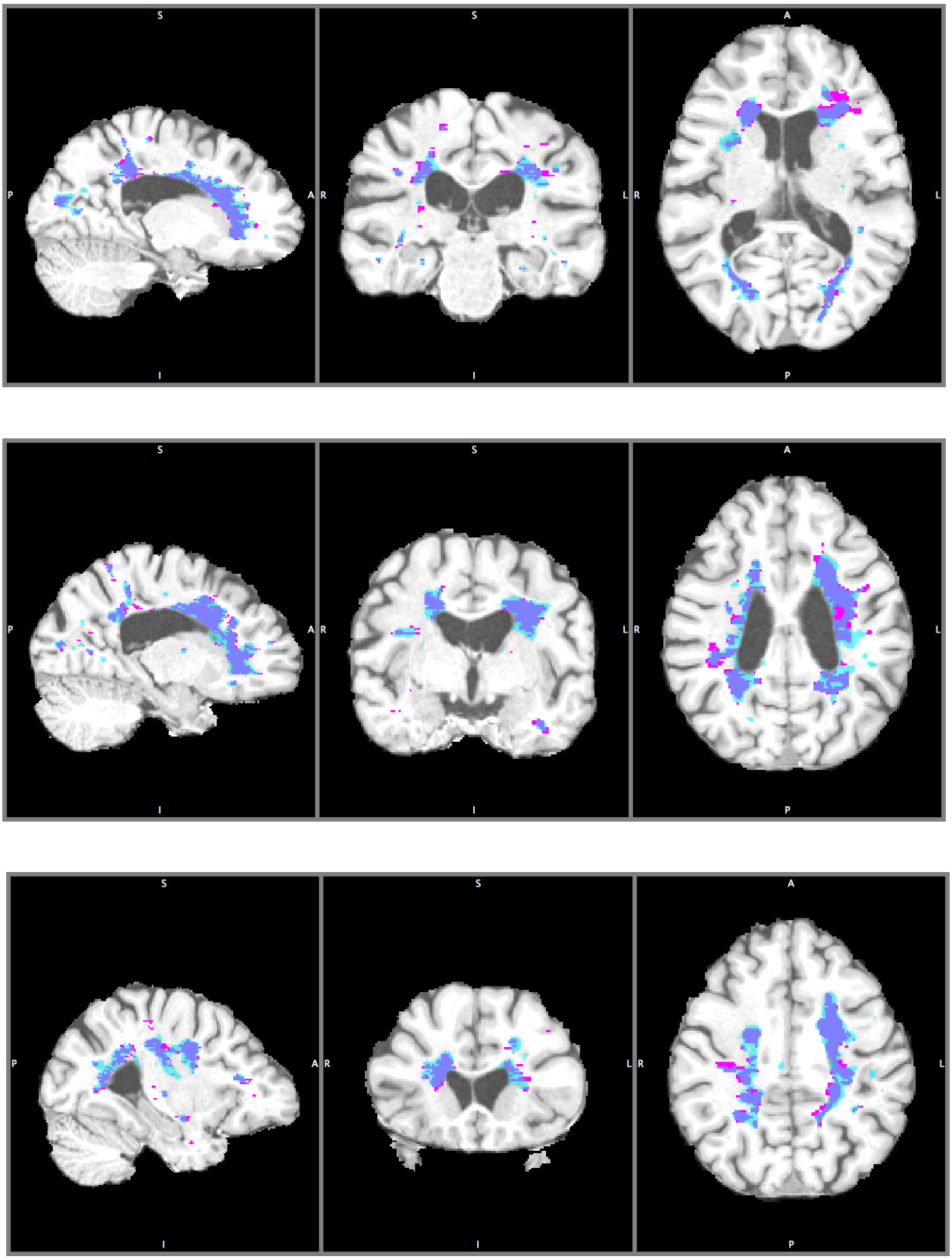

Figure 27: Qualitative Results for Subject 083MSVIS

pink:false negative
blue: false positive
purple: true positive

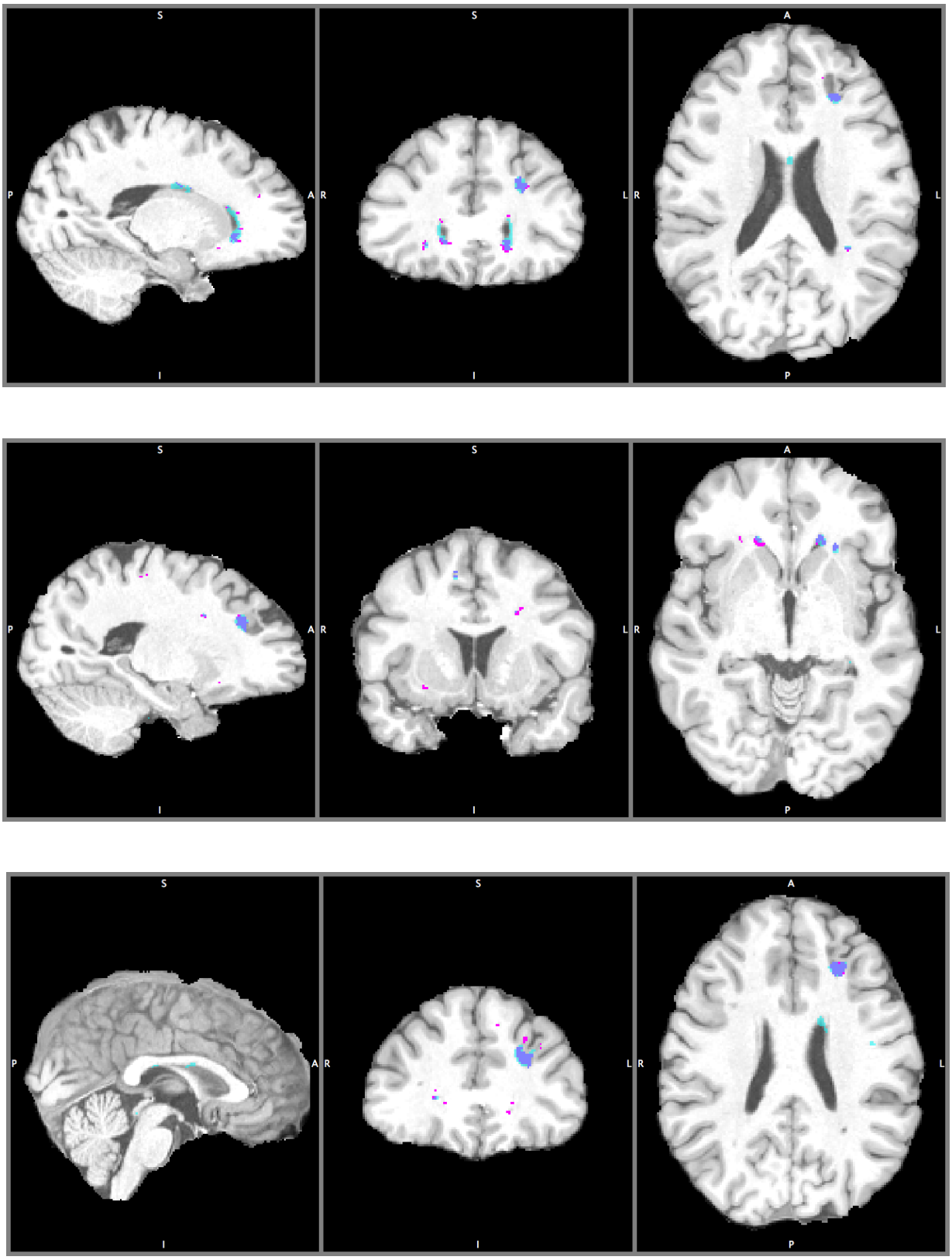

Figure 28: Qualitative Results for Subject 084MSVIS

pink:false negative
blue: false positive
purple: true positive

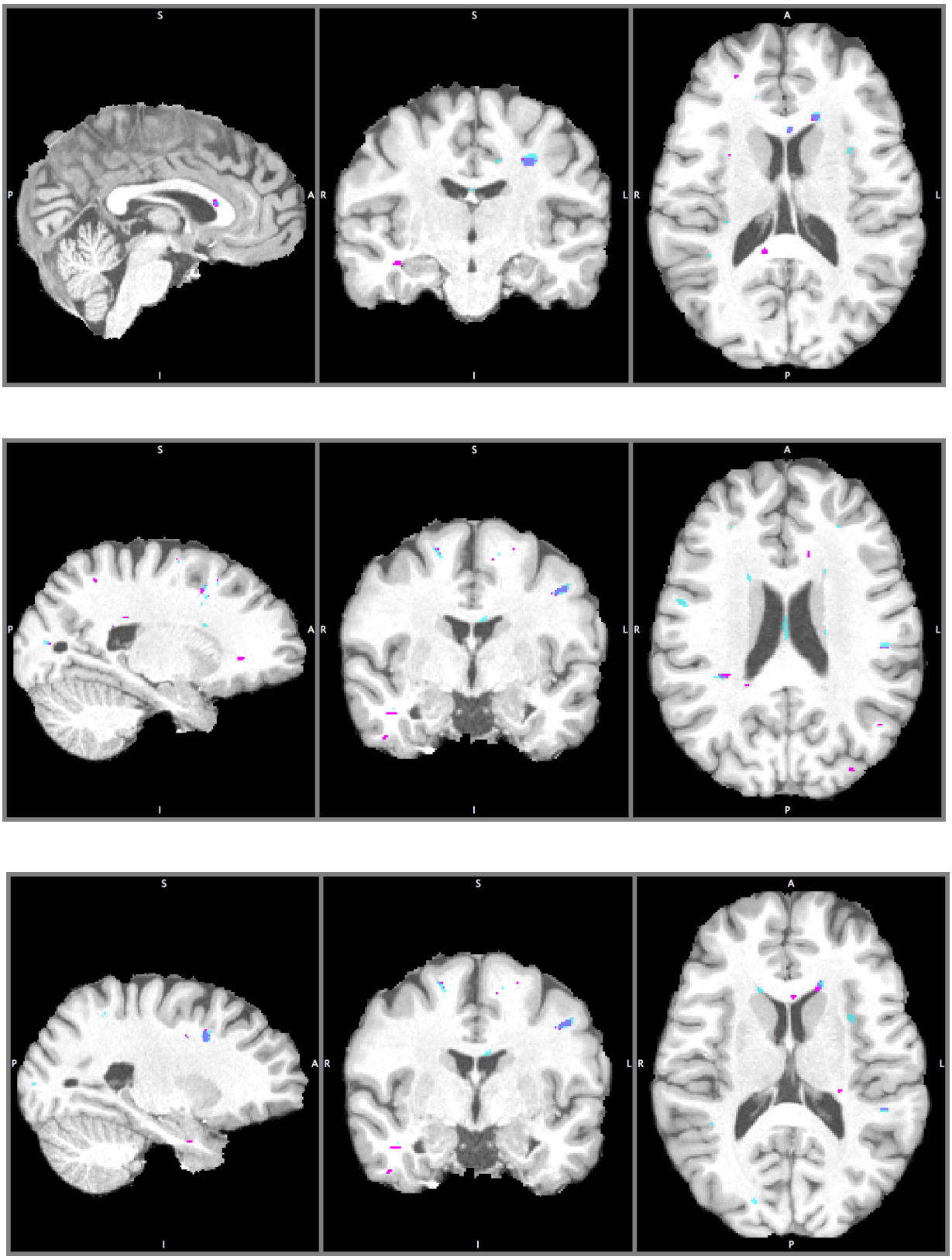

Figure 29: Qualitative Results for Subject 088MSVIS

pink:false negative
blue: false positive
purple: true positive

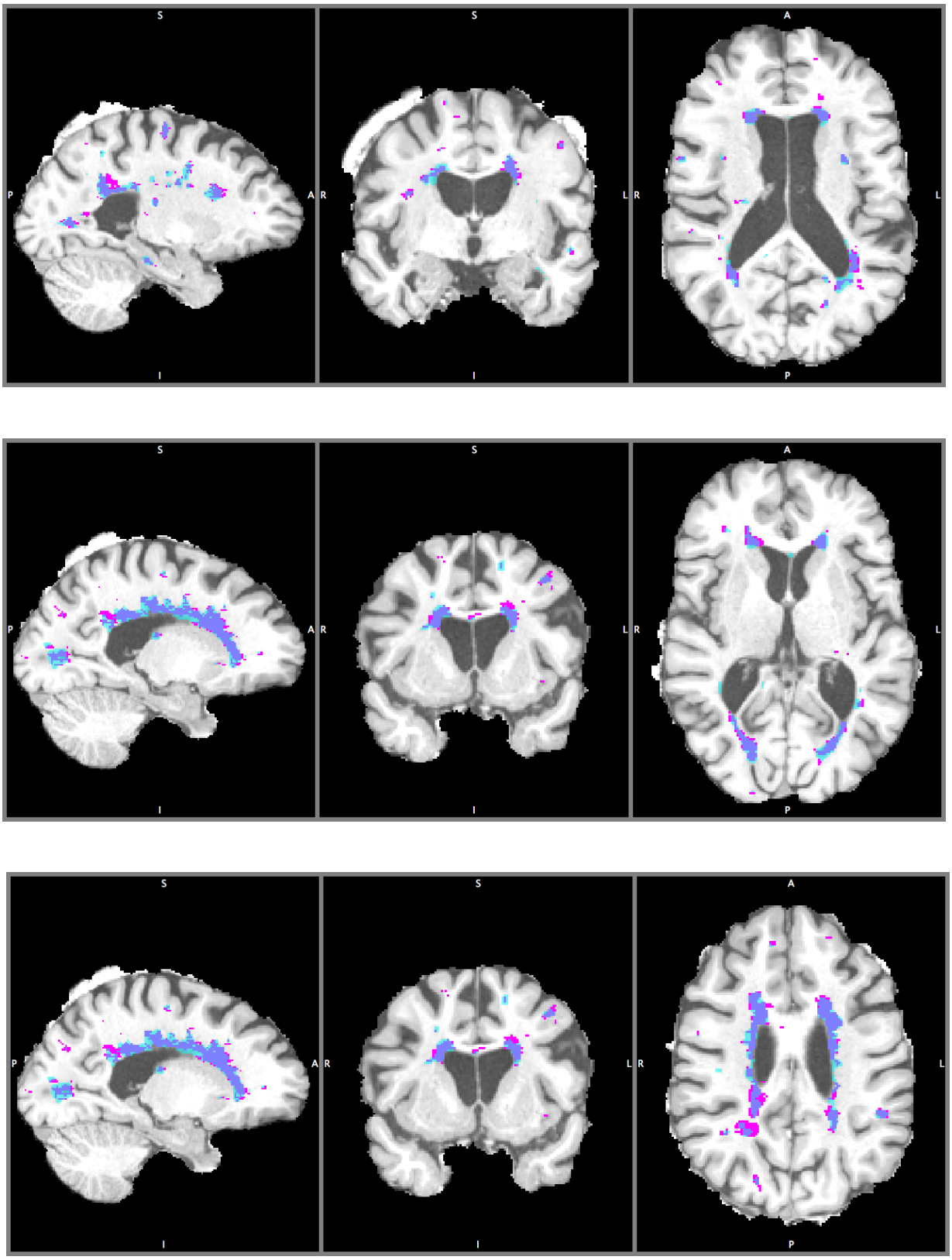

Figure 30: Qualitative Results for Subject 090MSVIS

pink:false negative
blue: false positive
purple: true positive

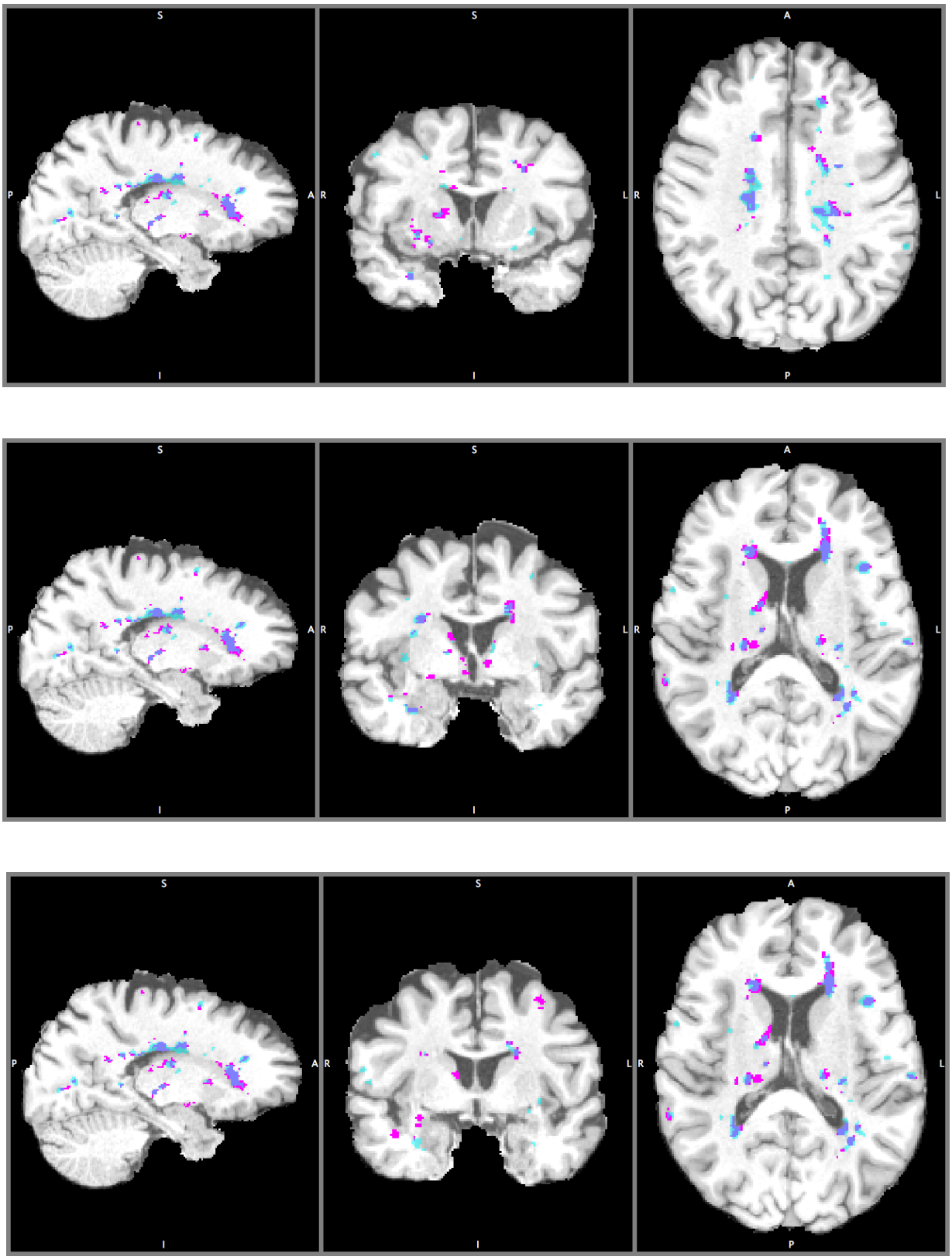

Figure 31: Qualitative Results for Subject 091MSVIS

pink:false negative
blue: false positive
purple: true positives

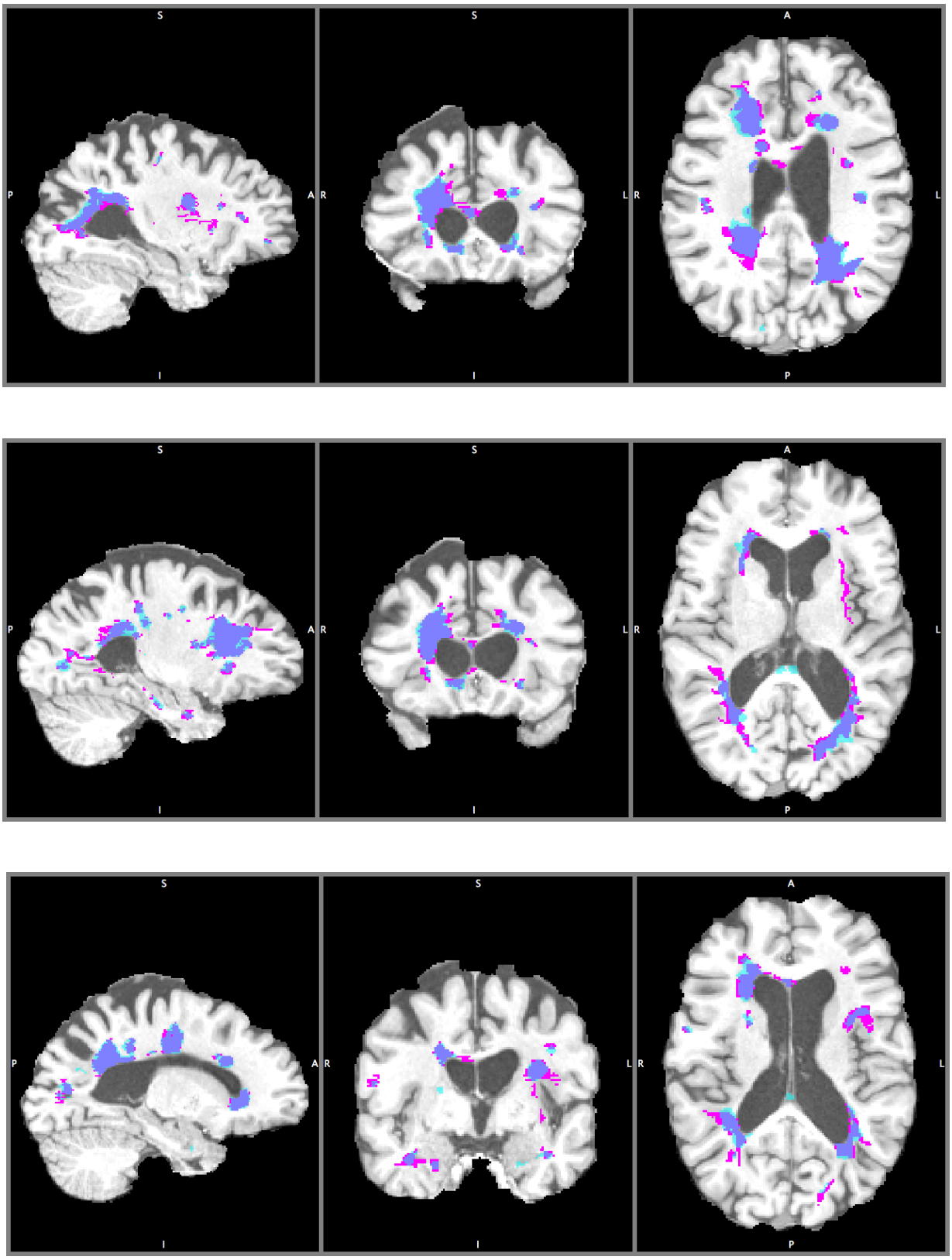

Figure 32: Qualitative Results for Subject 092MSVIS

pink:false negative
blue: false positive
purple: true positives

5 Conclusion

In this study, we implemented several deep learning models for the MS Lesion Segmentation problem. We based some of the models on similar approaches from the literature by adding our own design decisions. We evolved our models by validating with the experiments we made and, thus, we have been able to improve our results. We have seen that with a good design of the implementation and enough data, we can obtain a model using deep learning that learns the general rule for the segmentation. We found that the sub-sampling process, the patch-size, the number of classes, adding context information, adding multiple stages, etc. can have an effect on the performance of the model found. The best model we have found has been the 3-class model with a cascade approach. We obtained an average dice score of 57.5% and a true positive rate of 59.7% for a real test dataset of 9 patients from Hospital Clinic, which provided significant improvement over the commonly used automatic method of LST [22], based on the validation measures used.

All the models we found have learned the pattern to a certain degree but the best similarity with a human expert segmentation was obtained with the 3-class cascade model, which also means that this model has learned the pattern in the data the best. This model was able to detect the lesion regions, which have different intensity values in MRI Images with respect to their neighbourhood, meaning that it has captured the mathematical relationship between a 3D patch (neighbourhood of the voxel) and the class of its center voxel to a certain degree. Although it has captured the general relationship, it has failed to learn some exceptions requiring domain knowledge that are applied by human experts during MS Lesion Segmentation. The reason for this could possibly be that there was not enough cases in the training set for such exceptions or that it is necessary to feed more information to the CNN (e.g. more context) for the model to capture these patterns. As a result, we can argue that Deep Learning methods are a good solution to learn the patterns which are mathematically difficult to formulate such as MS Lesion Segmentation.

The success obtained by applying deep learning methods on a wide range of problems in neuroimaging should be more than mere chance. Although it is difficult to explain the obtained model, there are many clues as to why successful models are obtained with deep learning. There are a huge number of practical and theoretical studies concerning deep learning that we frequently hear about a new deep learning technique or approach, or an improved fine-tuning technique, a new activation function, loss function, learning algorithm etc. Moreover, the data in neuroimaging is accumulating and getting better in quality, and there are more and more efforts to centralize and standardize the data. All these improvements will work to the advantage of deep learning methods, which require a high amount of data and appropriate tuning to perform better. The results already obtained in neuroimaging problems are already very promising and considering the efforts put in improving the existing techniques and data, the results in the future are set to be even better.

6 Future Work

One shortcoming of the models we tried is that they do not have the knowledge or the intuition a human expert has, so it cannot apply any exceptions to the pattern it has learned. To give an example, some regions of the brain are more prone to having lesions. Although we added the location information to add some context, this may not have been enough since the training set may not be representative of these exceptions. Adding the lesion/non-lesion label information (detected by the model) of the neighbouring voxels in the evaluation of a voxel may help achieve better classification. In order to do this, conditional random fields can be used.

Also, in the literature restricted boltzmann machines or autoencoders are used to obtain an initial representation of the data, which will lead to better classifiers. This type of unsupervised methods can also take advantage of unlabeled data. Therefore, it might be a good idea to start with an RBM or auto-encoder and apply our models subsequently.

We observed during our experiments that smart sub-sampling affects the performance to a great degree. Although we have developed our own sub-sampling methods with which we saw an improvement, it can be further improved by carefully analyzing the MRI images or by choosing the MRI images of which the quality of manual segmentation is very high. In this way, the outliers are minimized. We can also explicitly use methods to remove the outliers from our training dataset to increase the quality of the training dataset.

Another improvement may come from increasing the number of MRI modalities trained during training. We only used T1, T2. Adding modalities such as FLAIR, fMRI, diffusion MRI would give more information as to the nature of a voxel.

In the segmentation we obtained, there were some cases in which there was a hypo-intensity in T1 but no hyper-intensity in T2. This should not be labeled as a lesion but the model labeled it as a lesion, which was a mistake on its part. This was a difficult rule to learn during training since there are not many of such cases in the training set. To integrate this information, separate networks for T1 and T2 can be trained and the separate results obtained from two networks can be combined for each voxel.

To remove false positives without sacrificing true positives, the number of cascade levels can be increased starting from a very coarse classifier (with a high sensitivity), which should include all positive cases, and later increasing the specificity of the classifier with small steps. This way removal of the non-lesion voxels can be done gradually and more carefully, removing first the easy cases and continuing with harder cases each time.

References

[1] M Symms, H R Jager, K Schmierer, T A Yousry *A review of structural magnetic resonance neuroimaging.* Neuroscience for Neurologists, (2004)

[2] *Neuroimaging Data Management* .

Retrieved from https://www.coursera.org/learn/clinical-data-management/lecture/bid77/neuroimaging-data-management

[3] *Neuroimaging Data Processing* .

Retrieved from https://en.wikibooks.org/wiki/Neuroimaging_Data_Processing.

[4] Nicholas J. Tustison, and James C. Gee *Introducing Dice, Jaccard, and Other Label Overlap Measures To ITK.* University of Pennsylvania, 2009.

[5] Mohammad Havaei, Nicolas Guizard, Hugo Larochelle, and Pierre-Marc Jodoin *Deep learning trends for focal brain pathology segmentation in MRI.* arXiv:1607.05258 2016.

[6] H. Greenspan, B. van Ginneken and R. M. Summers *Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique.* EEE Transactions on Medical Imaging, 35 (5), pp. 1153-1159, 2016.

[7] Daniel GarciaLorenzo, Simon Francis, Sridar Narayanan, Douglas L. Arnold, D. Louis Collins *Review of automatic segmentation methods of multiple sclerosis white matter lesions on conventional magnetic resonance imaging.* Medical Image Analysis, 2013.

[8] Suthirth Vaidya, Abhijith Chunduru, Ramanathan Muthu Ganapathy, Ganapathy Krishnamurthi *Longitudinal Multiple Sclerosis Lesion Segmentation Using 3D Convolutional Neural Networks.* MS Challenge miccai, 2015.

[9] Brosch, T., Yoo, Y., Tang, L.Y., Li, D.K., Traboulsee, A., Tam, R. *Deep convolutional encoder networks for multiple sclerosis lesion segmentation.* Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 3-11. Springer, 2015.

[10] Brosch, T., Tang, L., Yoo, Y., Li, D., Traboulsee, A. *Deep 3d convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation.* IEEE Transactions on Medical Imaging, 2016.

[11] M. Ghafoorian, N. Karssemeijer, T. Heskes, I. van Uden, C. Sanchez, G. Litjens, F.-E. de Leeuw, B. van Ginneken, E. Marchiori, B. Platel *Location sensitive deep convolutional neural networks for segmentation of white matter hyperintensities.* 2016.

[12] Sergi Valverde, Mariano Cabezas, Eloy Roura, Sandra Gonzalez-Villa, Deborah Pareto, Joan C. Vilanova, Lluis Ramio-Torrenta, Alex Rovira, Arnau Oliver, Xavier Llado *Improving automated multiple sclerosis lesion segmentation with a cascaded 3D convolutional neural network approach.* 2017.

[13] Guizard, N., Coupe, P., Fonov, V.S., Manjon, J.V., et al. *Rotation-invariant multicontrast non-local means for MS lesion segmentation* 2015

[14] Sergey M. Plis, Devon R. Hjelm, Ruslan Salakhutdinov, Elena A. Allen, Henry J. Bockholt, Jeffrey D. Long, Hans J. Johnson, Jane S. Paulsen, Jessica A. Turner and Vince D. Calhoun *Deep learning for neuroimaging: a validation study.*

[15] Ian GoodFellow, Yoshua Bengio, Aaron Courville *Deep learning.*

[16] Yoshua Bengio *Practical recommendations for gradient-based training of deep architectures* Version 2, Sept. 16th, 2012.

[17] *National MS Society.*

Retrieved from http://www.nationalmssociety.org/

[18] *Introduction to recurrent neural networks.*

Retrieved from http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

[19] Enrique Romero, Rene Alquezar *Advanced Computational Intelligence Slides - FIB UPC*

[20] *Deep Learning.*

Retrieved from http://deeplearning4j.org/

[21] *A Tutorial on Deep Learning: Autoencoders, Convolutional Neural Networks and Recurrent Neural Networks.*

Retrieved from https://cs.stanford.edu/ quocle/tutorial2.pdf

[22] *LST - MS Segmentation Tool .*

Retrieved from http://www.statistical-modelling.de/lst.html

[23] *Recurrent neural networks .*

Retrieved from https://en.wikipedia.org/wiki/Recurrent_neural_network

[24] *Convolutional Neural Networks .*

Retrieved from https://en.wikipedia.org/wiki/Convolutional_neural_network

[25] *Restricted Boltzmann Machines.*

Retrieved from https://en.wikipedia.org/wiki/Restricted_Boltzmann_machine

[26] *Electroencephalography .*

Retrieved from https://en.wikipedia.org/wiki/Electroencephalography

[27] *Magnetic Resonance Imaging .*

Retrieved from https://en.wikipedia.org/wiki/Magnetic_resonance_imaging

[28] *Positron Emission Tomography .*

Retrieved from https://en.wikipedia.org/wiki/Positron_emission_tomography

[29] *Magnetoencephalography .*

Retrieved from https://en.wikipedia.org/wiki/Magnetoencephalography

[30] Martin Lindquist *Principles of Functional Neuroimaging Data .*

Retrieved from https://www.samsi.info/wp-content/uploads/2016/03/Lindquist_OW_august2015.pdf