A Cascade of CNNs for Multiple Sclerosis Lesion Segmentation

Ian Jacobsen

I. Introduction

A reproduction of the algorithm [1] that won the international MICCAI 2016 challenge [2] for developing an automated method for white matter lesion segmentation in MRI scans of patients with multiple sclerosis (MS) was written and tested on a subset of patient data provided by France Life Imaging [3].

The algorithm is comprised of a cascade of two patch-wise convolutional neural networks that are trained in tandem to individually classify small volumes of the MRI on a pixel-by-pixel basis. The individual pixel classifications are then combined to reconstruct a completely segmented image, differentiating lesion areas from non-lesion areas.

II. Background

Patients with MS develop lesions in white matter tissue. MRI scans are used by physicians to both diagnose and monitor the evolution of MS over time. The evolution of the size of the lesion areas and the number of lesion areas are useful metrics for determining the patients stage of MS, and the effectiveness of the treatment [4].

An automated method for lesion segmentation is sought after because the act of manually annotating MRI scans is time consuming, expensive, and has great inter-expert variability [2]. If an accurate method for classifying lesion areas were available, physicians would be able to quantify the severity of the disease and the effectiveness of the treatment with repeatability.

The data set that was used includes professionally annotated diagnosis images for a total of 15 patients from a total of 7 experts. The experts' diagnoses were combined into a consensus image, which was used as the ground truth. The details of how the images were combined to form the consensus image can be found in [2]. The consensus image is simply a binary segmentation between lesion volumes and non-lesion volumes.

For the sake of developing a method which can generalize to a variety of MRI scanners, training data was used from a total of three different MRI scanners. They are a Philips Ingenia 3T scanner, a Siemens Aera 1.5T scanner, and a Siemens Verio 3T scanner.

III. Overview

Due to the difficulty of obtaining MRIs that are professionally annotated by radiologists, the limited sets of patient MRI scans are broken down into patches, which are individually used to train the two networks. Breaking the MRI down into patches greatly increases the size of the dataset, which is the key to training the cascade of neural networks. The patches are obtained by sampling small volumes of the complete MRI volumes. In this work, the patches are patches of the shape (x=11 pixels, y=11 pixels, z=11 pixels).

Throughout this report, patches where the center pixel is centered on a lesion will be referred to as a positive patch, and patches where the center pixel is not centered on a lesion will be referred to as a negative patch.

The system that was used is a cascade of two convolutional neural networks. The architecture of the individual networks is identical, but the training process differs between the two. The first network in the cascade, which will be referred to as network one, or N1 for short, was trained to select candidate patches which may be centered on a lesion. This approach was taken to address the extreme variation between patches in a single MRI scan; therefore the goal of N1 is to simply filter out patches which are unlikely to be centered on a lesion. The second network, which will be referred to as N2, was trained with the intention of it being fine tuned well enough to make the final decision of whether or not a candidate patch is centered on a lesion.

IV. Training Data

The MRI sequence that was used is the 3D fluid-attenuated inversion recovery, referred to as the FLAIR sequence [5]. FLAIR sequences are well suited to bring out the variant of lesions that are present in patients with MS [6].

In an effort to standardize the data between different MRI scanners, the entire FLAIR sequence was normalized by subtracting the mean and dividing by the variance. Patches are what was used to train the networks. In this project, a patch is a volume that is sampled from the FLAIR image. The center pixel of the patch is the pixel which is to be classified as positive or negative.

The size of the FLAIR sequence varied between scanners, the largest image being 512 x 512 x 224, and the smallest being 226 x 224 x 128.

V. Model Description

The architecture that was selected for N1 and N2 is a relatively shallow network; it consists of only seven layers. The input layer takes in a patch, feeds the patch into a convolutional layer, then passes through a max-pooling layer, then passes through another convolutional layer, into another max pooling layer, then a fully connected layer into a dropout layer, and the final layer is a softmax activation function which provides the classification.

VI. Training Procedure

Naturally this system requires a tandem training procedure. N1 is trained first, and then the final form of N1 is used to select training data for N2.

To select the patches for N1, a minimum distance between patches was set. The reasoning behind a minimum distance is because we want to train the network on a wide variety of patches, and two directly adjacent patches are extremely similar. By discarding directly adjacent patches, we allow the network to see a variety of equally important patches. The minimum distance used was (x=4 pixels, y=4 pixels, z=1 pixel). In other words, if the distance between the coordinates of the center pixels for two patches is less than each of the minimum distance values, then only one patch is selected for training.

After finding a set of potential training patches for N1, the list of negative patches was truncated. This is due to the fact that in patients who have advanced MS, i.e. where the total lesion volume is greater than 20mL, the lesions make up approximately 1.5% of the total brain volume [1]. Therefore the number of positive patches is much smaller than the number of negative patches. An equal proportion of positive to negative examples is desired in the training set, so the set of negative patches is subsampled down to be equal to the number of available positive examples.

Once the training set for N1 is ready, the network is ready to be trained. The ADADELTA optimizer [7] was used to train the network, as specified in [1]. The learning rate in ADADELTA is adaptive, but the initial learning rate has an effect on the training. Therefore the initial learning rate is a hyperparameter that must be selected through a validation process. The batch size used for each epoch is an additional hyperparameter that must be varied and selected using validation.

After the training of N1 was complete, every patch available in the MRI volume was fed through N1. All of the false-positive patches were used to create a set of potential training patches for N2. The entire set of positive patches that was used to train N1 was concatenated with a subset of the false-positive set, so that the new set had an equal amount of true positive patches and false positive patches. This new set was then used to train N2. Similarly to the training of N1, the ADADELTA optimizer was used, and the batch size and learning rate hyperparameters were varied as selected using the best performing validation set.

VII. Validation and Testing

Two methods of validation were used, but only one will be reported on. The method that will be reported on is the leave-one-out method. In the leave-one-out framework, for each combination of choosing 14 out of the available 15 patients, patches were fetched from 14 of the 15 patients, then the patches were placed in a pool and shuffled, then 80% of the patches were used for training, the remaining 20% of the patches were set aside for validation, and the 15th patient was set aside for testing. To be clear, none of the patches from the 15th patient are used to train or validate the networks. This procedure is repeated for a total of 15 times, so that each patient that was left out during the training has its own model.

The second method was based on training on patients from each of the different MRI machines (3 patients from the Siemens 3T Verio scanner, 3 patients from the Siemens 1.5T Aera scanner, and 3 patients from the Philips 3T Ingenia scanner) , then using one patient from each scanner for validation, and setting aside one patient from each scanner for testing.

VIII. Results and Discussion

TABLE I: Test Results for N1LR=0.3, N2LR=0.3, batch=512

	num pixels	accuracy	false positives	false negatives	dice
07043SEME	1061417	0.983526	0.0112256	0.00524864	29
07001MOEL	986032	0.937745	0.0619351	0.000319462	3
01016SACH	4774673	0.900671	0.0916961	0.00763298	47
08027SYBR	3543293	0.923772	0.0761882	3.97935e-05	2
01038PAGU	5445523	0.978271	0.0146596	0.00706966	41
07010NABO	924395	0.971475	0.0278182	0.000706408	10
01042GULE	4908547	0.950232	0.0337057	0.0160622	18
01039VITE	5228366	0.964409	0.0270974	0.00849348	51
07003SATH	876275	0.964785	0.00288152	0.0323335	31
08037ROGU	3858673	0.939312	0.0597617	0.000926225	35
08002CHJE	3994067	0.978413	0.0203577	0.00122907	33
08029IVDI	3523149	0.908348	0.0903978	0.00125371	36
07040DORE	982787	0.991983	0.00593923	0.00207776	26
01040VANE	3930109	0.987728	0.0119317	0.000340194	7
08031SEVE	3565957	0.993429	0.00615094	0.000420364	52
Average	3.17355e+06	0.958273	0.0361164	0.00561023	28.0667

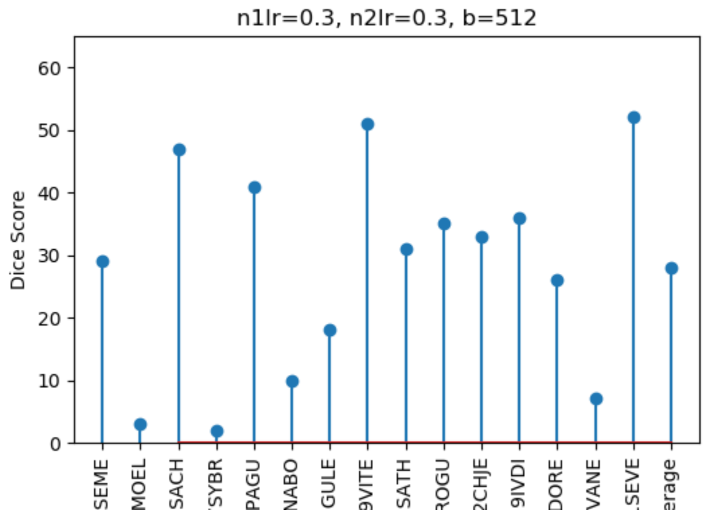

Fig. 1: Dice Score with Learning Rates of 0.3

TABLE II: Test Results for N1LR=0.03, N2LR=0.03, batch=512

	num pixels	accuracy	false positives	false negatives	dice
07043SEME	1061417	0.98532	0.0108628	0.00381754	39
07001MOEL	986032	0.925387	0.0743181	0.000295122	2
01016SACH	4774673	0.831046	0.168325	0.000629572	38
08027SYBR	3543293	0.905221	0.0947158	6.29358e-05	2
01038PAGU	5445523	0.987371	0.00466273	0.00796673	51
07010NABO	924395	0.993595	0.00531591	0.00108936	29
01042GULE	4908547	0.949261	0.0342712	0.0164676	17
01039VITE	5228366	0.963937	0.0223169	0.0137462	42
07003SATH	876275	0.964754	0.00463781	0.030608	35
08037ROGU	3858673	0.983629	0.00937758	0.0069936	56
08002CHJE	3994067	0.944291	0.0552855	0.000423128	18
08029IVDI	3523149	0.969521	0.026912	0.00356698	61
07040DORE	982787	0.98828	0.00973151	0.00198822	21
01040VANE	3930109	0.970369	0.0295905	4.07113e-05	5
08031SEVE	3565957	0.988017	0.0116451	0.000337918	37
Average	3.17355e+06	0.956667	0.0374645	0.0058689	30.2

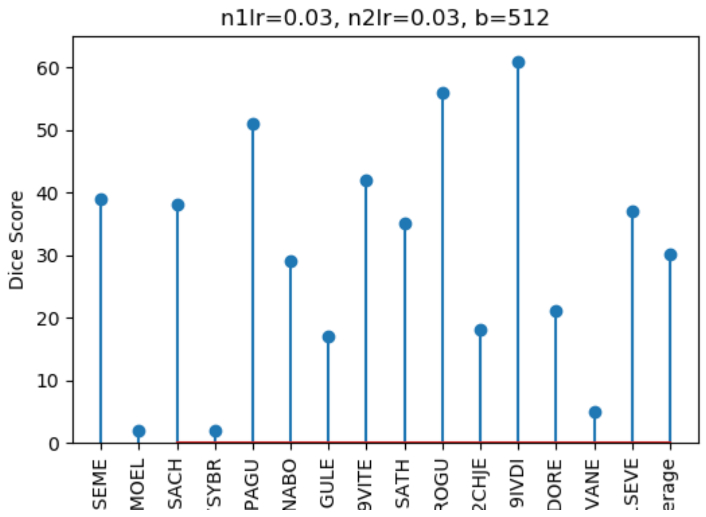

Fig. 2: Dice Score with Learning Rates of 0.03

TABLE III: Test Results for N1LR=0.003, N2LR=0.003, batch=512

	num pixels	accuracy	false positives	false negatives	dice
07043SEME	1061417	0.990295	0.00558781	0.00411714	48
07001MOEL	986032	0.971902	0.0271594	0.000939118	2
01016SACH	4774673	0.763334	0.21103	0.0256365	18
08027SYBR	3543293	0.87649	0.123206	0.000303955	1
01038PAGU	5445523	0.979083	0.0117553	0.00916184	34
07010NABO	924395	0.977142	0.0209077	0.00195046	3
01042GULE	4908547	0.932835	0.0487776	0.0183875	9
01039VITE	5228366	0.943413	0.0349878	0.0215989	16
07003SATH	876275	0.961829	0.00972298	0.0284477	38
08037ROGU	3858673	0.882558	0.115258	0.00218417	20
08002CHJE	3994067	0.895797	0.102547	0.00165546	8
08029IVDI	3523149	0.651275	0.348107	0.000617913	13
07040DORE	982787	0.991183	0.00568384	0.00313293	8
01040VANE	3930109	0.954732	0.044726	0.00054197	1
08031SEVE	3565957	0.970951	0.028087	0.000962434	17
Average	3.17355e+06	0.916188	0.0758362	0.00797587	15.7333

After adding an additional hyperparameter, the thresholding value, we can see that the best value to use is 0.7. The final layer of N1 and N2 are softmax functions, and the threshold is used to separate negative patches from positive patches. The thresholding value is used for both the training of N2, and for the final classification. To be more specific, the thresholding value is used to select the false positive patches from N1 that are used to train N2. For the final classification, any patch that N2's softmax output layer is greater than the thresholding value is classified as a positive patch. Table IV shows the average test performance when the threshold value is varied between $0.4, 0.5, 0.6, 0.7$, and 0.8.

TABLE IV: Average Test Performance with Varied Threshold

learning rate	threshold	accuracy	false positives	false negatives	dice
0.3	0.4	0.950180	0.045977	0.003843	28.600000
0.3	0.5	0.925438	0.070278	0.004283	24.466667
0.3	0.6	0.964133	0.031387	0.004479	31.133333
0.3	0.7	0.961509	0.033541	0.004950	31.200000
0.3	0.8	0.962565	0.031260	0.006175	30.933333
0.03	0.4	0.933739	0.061783	0.004478	23.866667
0.03	0.5	0.942325	0.054020	0.003655	29.600000
0.03	0.6	0.960502	0.034922	0.004577	32.666667
0.03	0.7	0.974749	0.019138	0.006113	37.333333
0.03	0.8	0.983256	0.008443	0.008301	34.533333

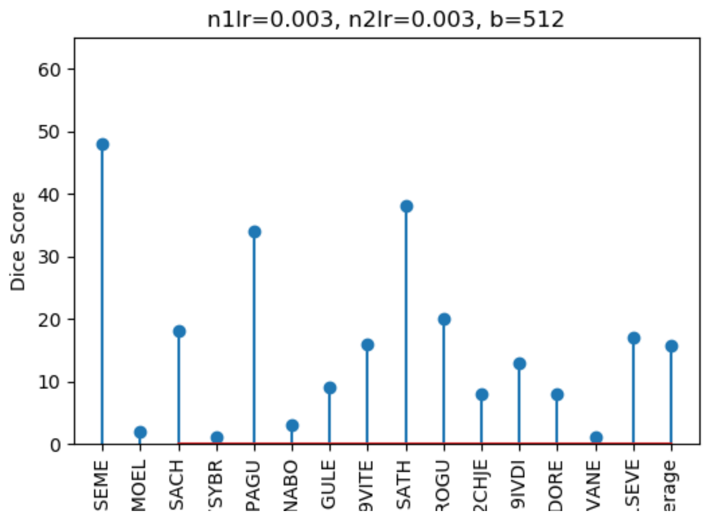

Fig. 3: Dice Score with Learning Rates of 0.003

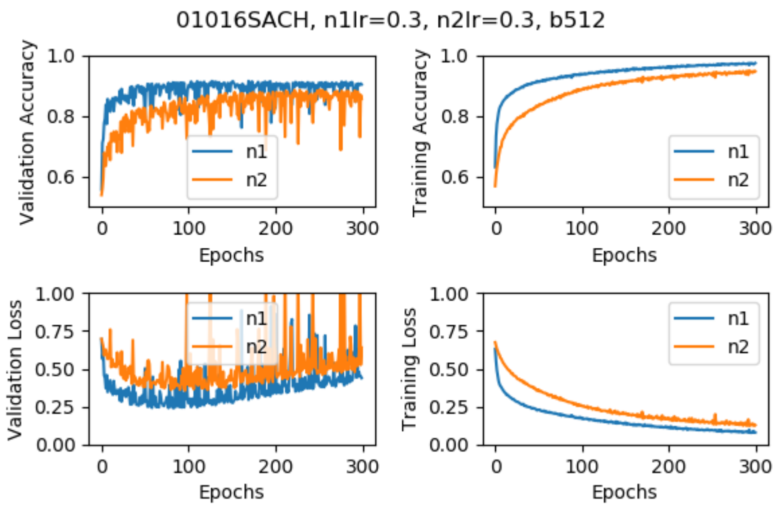

Fig. 4: Training and Validation with Learning Rates of 0.3

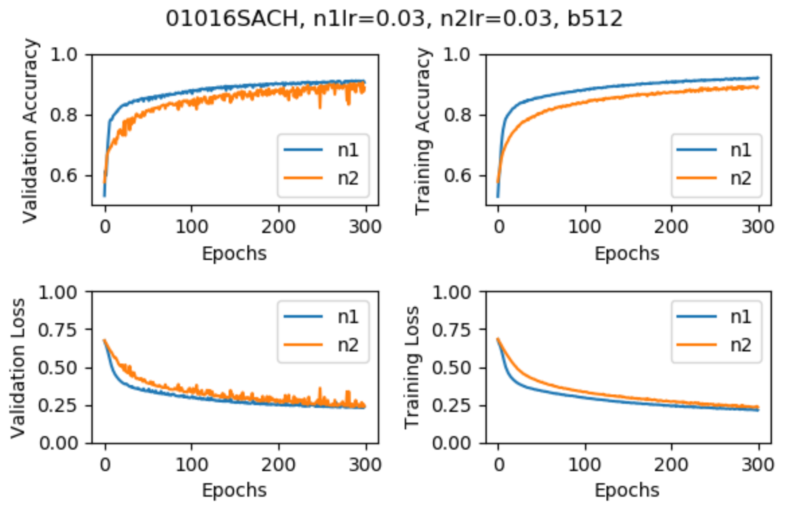

Fig. 5: Training and Validation with Learning Rates of 0.03

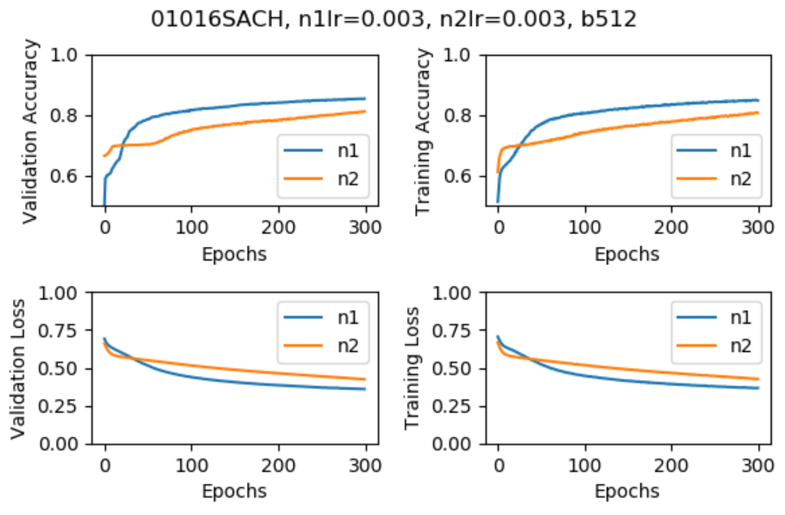

Fig. 6: Training and Validation with Learning Rates of 0.003

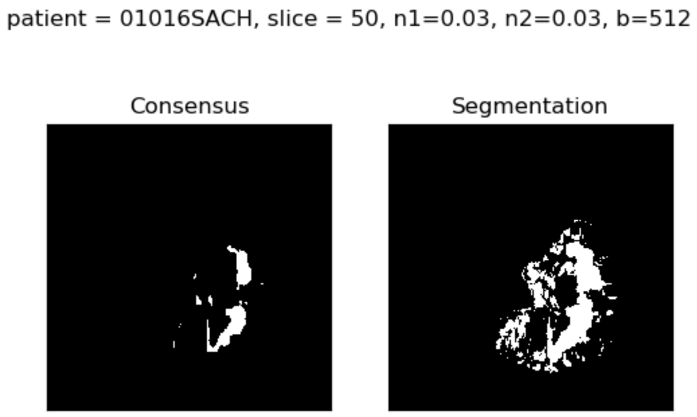

Fig. 7: Slice of Patient 01016SACH with Learning Rate of 0.03

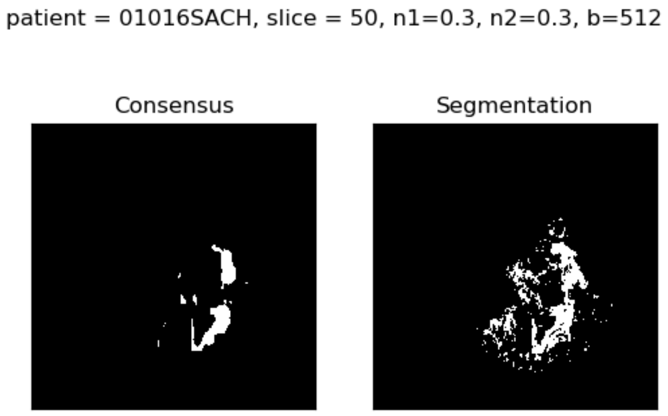

Fig. 8: Slice of Patient 01016SACH with Learning Rate of 0.3

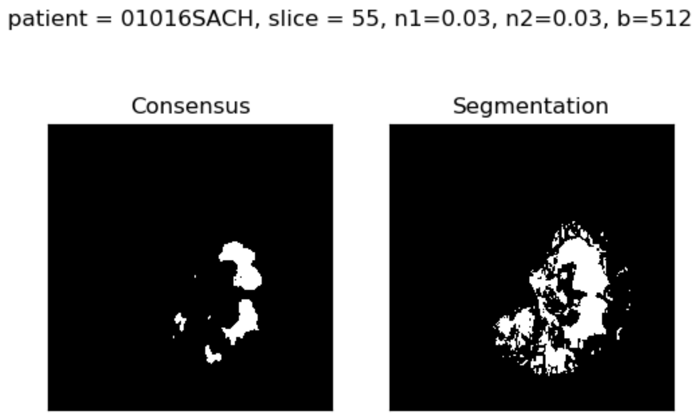

Fig. 9: Slice of Patient 01016SACH with Learning Rate of 0.03

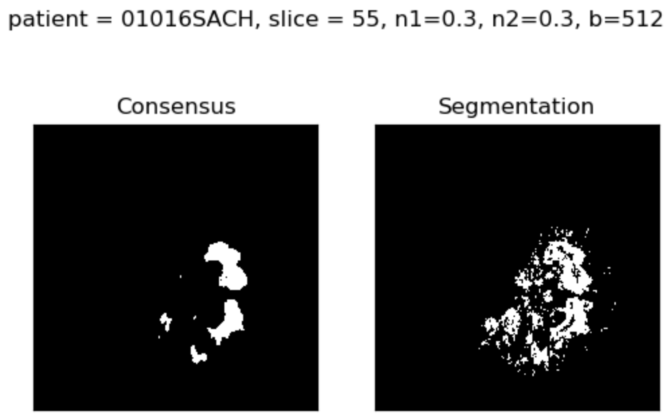

Fig. 10: Slice of Patient 01016SACH with Learning Rate of 0.3

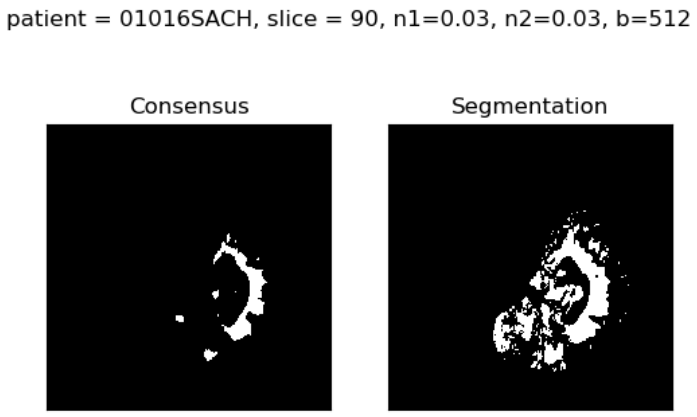

Fig. 11: Slice of Patient 01016SACH with Learning Rate of 0.03

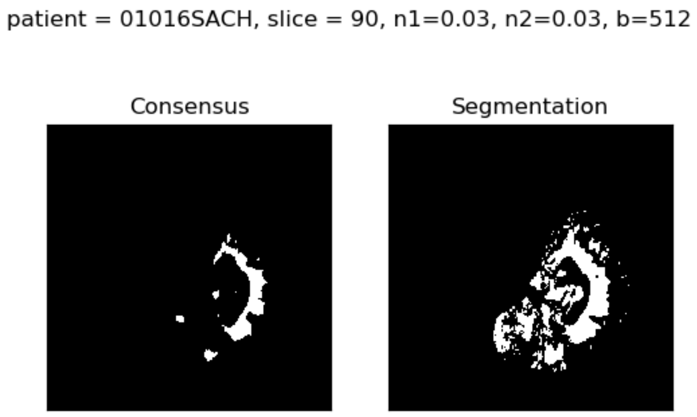

Fig. 12: Slice of Patient 01016SACH with Learning Rate of 0.3

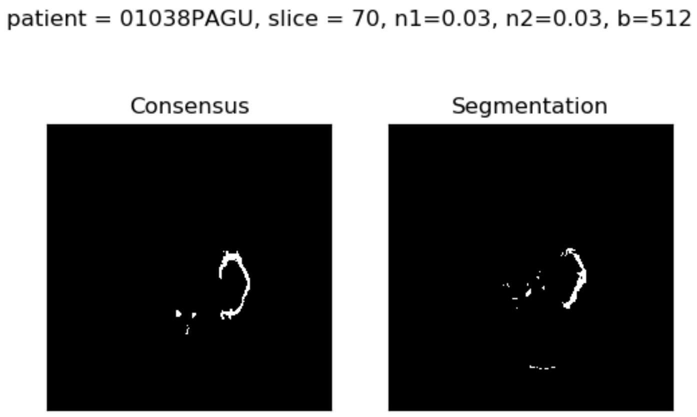

Fig. 13: Slice of Patient 010138PAGU with Learning Rate of 0.03

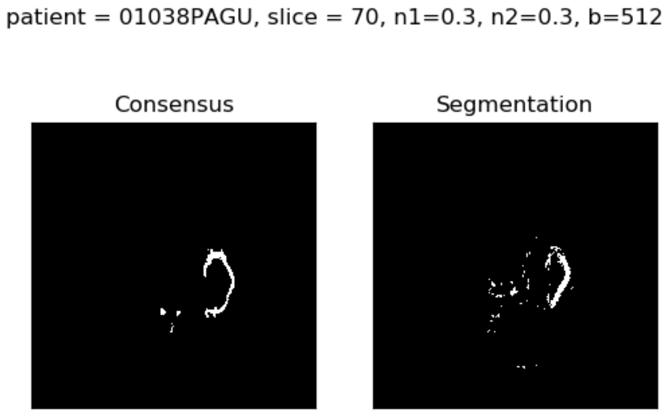

Fig. 14: Slice of Patient 010138PAGU with Learning Rate of 0.3

The tables show the number of pixels that were classified in each patient, the percent of correctly classified pixels, the percent of all classified pixels that were wrongly classified as positive examples, the percent of all classified pixels that were wrongly classified as negative pixels, and the Dice Similarity Coefficient [8]. Following [1], the Dice Similarity Coefficient (DSC) is defined by:

$$DSC = \frac{2 \times TP}{FN + FP + 2 \times TP} \times 100 \tag{1}$$

where TP is the number of true positive pixels, FN is the number of false negative pixels, and FP is the number of false positive pixels.

The model with the highest DSC on the test data was acheived using a learning rate of 0.3 for N1, a learning rate of 0.3 for N2, and a batch size of 512.

It is difficult to make a comparison with the results that were reported in [1] because the data set that was used in [1] is different than what was used in this reproduction. The sizes of the data sets that were used in [1] are $n = 25$, and $n = 35$. With that said, the mea

Figreported in [1] is 53.5 for the $n = 35$ set, and 56.0 for the $n = 25$ set. These values are significantly higher than what was achieved in this reproduction, where the highest average DSC score between hyperparameters is 30.2.
ure 4, Figun DSC re 5, and Figure 6 shows the impact on the training and validation when the hyperparameters are varied. We can see that with a larger learning rate the validation becomes more unstable, but a smaller learning rate inhibits the learning of the models. It is especially important that N1 trains well, because the quality of the samples used to train N2 depend on the training of N1.

Figures 7, 8, 9, 10, 11, 12, 13, and 14 display a comparison between the consensus image and the output segmentation of N2. We can see that N2 can correctly find the general shape of most lesion areas, however the classification of the surrounding areas is not ideal.

IX. Conclusion

For future work, post-processing techniques will likely improve the final segmentation. From the figures that compared the consensus image with the N2 segmentation, we can see that there are small outlying batches of area that were wrongly classified as lesions by N2. One way to remedy this is to pass the output of N2 through a post-processing algorithm that filters out outlying areas that were likely misclassified, based on the surrounding pixels.

References

[1] S. Valverde, M. Cabezas, E. Roura, S. González-Villà, D. Pareto, J. C. Vilanova, L. Ramió-Torrentà, A. Rovira, A. Oliver, and X. Lladó, "Improving automated multiple sclerosis lesion segmentation with a cascaded 3d convolutional neural network approach," CoRR, vol. abs/1702.04869, 2017.

[2] O. C. et al, Objective Evaluation of Multiple Sclerosis Lesion Segmentation using a Data Management and Processing Infrastructures, 2018.

[3] "Mmseg portal," 2018.

[4] R. McKinley, L. Grunder, R. Wepfer, F. Aschwanden, T. Fischer, C. Friedli, R. Muri, C. Rummel, R. Verma, C. Weisstanner, M. Reyes, A. Salmen, A. Chan, R. Wiest, and F. Wagner, "Automatic detection of lesion load change in multiple sclerosis using convolutional neural networks with segmentation confidence," CoRR, vol. abs/1904.03041, 2019.

[5] B. R. et all, "Fluid-attenuated inversion recovery magnetic resonance imaging detects cortical and juxtacortical multiple sclerosis lesions.," 2001.

[6] P. G. de Lima Freire and R. J. Ferrari, "Multiple sclerosis lesion enhancement and white matter region estimation using hyperintensities in FLAIR images," CoRR, vol. abs/1807.09619, 2018.

[7] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," CoRR, vol. abs/1212.5701, 2012.

[8] L. R. Dice, "Measures of the amount of ecologic association between species," 1945.