Universidad Nacional de Educación a Distancia

Departamento de Inteligencia Artificial

Enhanced preprocessing and adaptive weighted loss function for improved for white matter hyperintensity segmentation with convolutional neural networks.

Master thesis for Master in Advanced Artificial Intelligence

Author: Pablo Duque Asens
Tutors: Mariano Rincón and Jose Manuel Cuadra

September 2020

Contents

1 Introduction **5**

2 Related work **6**
2.1 MRI preprocessing 6
2.2 Convolutional neural networks 7
2.3 Attention gates 8

3 Methodology **9**
3.1 Dataset 9
3.2 Preprocessing stages 10
3.3 Data augmentation 12
3.4 Network architecture 13
3.5 Training 15
3.6 Postprocessing 16

4 Results **17**

5 Discussion and future work **19**

6 Conclusion **21**

Appendices **27**

A Magnetic resonance imaging **27**

B Convolutional neural networks **28**
B.1 Image representation 28
B.2 Convolution layer 29
B.2.1 Padding 30
B.2.2 Strides 31
B.2.3 Activation layer 31
B.2.4 Kernel initializer 31
B.2.5 Transposed convolution 32
B.3 Pooling layer 33

C Attention gates **33**

D WMH evaluation **35**
D.1 Dice Similarity Coefficient 35
D.2 Hausdorff distance (95th percentile) 35
D.3 Average volume difference 35
D.4 Recall for individual lesions 36
D.5 F1 for individual lesions 36

E Code **36**

F Previous work **37**

Preface

This master thesis is presented in an article format with appendices to complete it with the background information that it requires. It is also suplemented by our previous work which was published in the 8th International Work-Conference on the Interplay Between Natural and Artificial Computation in 2019.

It is recommended for readers to start on appendices A, B, C, D and optionally F before reading the article itself to learn or review concepts that are assumed to be known in the article.

Enhanced preprocessing and adaptive weighted loss function for improved for white matter hyperintensity segmentation with convolutional neural networks.

Duque, P., Cuadra, J. M., & Rincón-Zamorano, M.

September 20, 2020

Abstract

There is a great interest in automating White Matter Hyperintensities (WMH) segmentation due to their importance in the medical field as well as the great amount of inter- and intra-observer variability that appears when it is manually segmented in magnetic resonance imaging.

In this work we present a multistep tailored preprocessing consisting mainly of brain extraction, intensity contrast enhancement, subject based slice cropping and intensity standardization. The segmentation task is then performed by a fully convolutional neural network with attention gates which employs a customized loss function based on the dice similarity coefficient and the F1 score.

Experimental results on the white matter hyperintensities segmentation challenge [Kuijf et al., 2019] show that our proposed preprocessing improves segmentation, that attention gated U-Net further improves segmentation tasks compared to the original U-Net and our proposed loss function has the potential to improve lesion-wise F1 on DSC based segmentations.

1 Introduction

In the world around 15 million people have a stroke every year, from which 6 million die and 5 million are left permanently disabled. Approximately 35.6 million people worldwide suffer from dementia and is expected to keep increasing [Wardlaw et al., 2015].

Cognitive impairment in older people alone or in combination with Alzheimer's disease contributes to substantial worsening of cognitive functions. This impairment generally results in visible lesions on brain scanning [Wardlaw et al., 2015].

White matter hyperintensity (WMH) is a pathology in the brain of elderly people which is related to cognitive impairment, dementia and increased risk of stroke. The most sensitive modality for detecting WMH is through magnetic resonance imaging (MRI). Since manual segmentation of WMH on MRI has proven to be a user-biased and time consuming process as well as presenting intra-observer variability [Rincón et al., 2017], automating this process is consequently of great interest in the field and a great amount of research has been dedicated during the last years.

Data is preprocessed based on a previous study of different preprocessing methods applied to the particular problem of white matter hyperintensity segmentation to boost the performance we are able to obtain using FConvNN [Duque et al., 2019]. This is because MRI scanner might capture a great amount of noise and artifacts during the scanning process. The ability to correctly produce an image that maximizes the success of the segmentation is key. This process is based on classical techniques used among other machine learning use cases as well as some specific ones that are commonly applied to MRI images and WMH segmentation.

We also present a modified version of the Attention Gated U-Net [Schlemper et al., 2019] to perform white matter hyperintensity segmentation. The idea behind this method is to combine the proven segmentation capabilities of fully convolutional neural networks (FConvNN) [Christ et al., 2016] [Guerrero et al., 2018], specifically those which follow a U-Net or similar architecture [Ronneberger et al., 2015] [Milletari et al., 2016] with the ability of attention gates (AG) to leverage salient regions in medical images [Schlemper et al., 2019].

In summary the main contributions of this paper are the following:

- Propose a tailored and improved preprocessing workflow for boosting the performance that a CNN is able to achieve segmenting WMH. Consisting in brain extraction, corregistration, contrast enhancement, sub-

ject based slice cropping, intensity standardization and data augmentation.

- Demonstrate how a loss function that combines lesion wise F1 and dice similarity coefficient reach better all around results in later stages of the training. This metric combines the segmentation power of the DSC but boosts its overall performance around individual lesion metrics.

- Modify the attention gated U-Net to improve its efficiency for WMH. Combined with all of the above to achieve a 0.079 rank (being 0 the highest rank and 1 the lowest) in the WMH challenge specifically getting great results in AVD, H95 and DSC metrics.

2 Related work

2.1 MRI preprocessing

Magnetic resonance imaging (MRI) differs from any other regular camera or similar tools that produce the images generally used in computer vision and therefore the way we preprocess this image will have to be tailored for MRIs and specifically for WMH segmentation.

Automating segmentation tasks in medical images is a topic of research that has seen a great increase over the last few years [Fourcade & Khonsari, 2019]. This is in most cases done by leveraging the power of fully convolutional neural networks [Ronneberger et al., 2015] [Duque et al., 2019]. That is why many efforts have gone into the preprocessing and enhancing of MRI images to allow CNNs to segment pathologies better [Duque et al., 2019]. This is done mostly by removing noise elements that images may contain as well as trying to enhance the features of the pathologies being studied. For brain MRI the skull is in this case a source of noise since there is no WMH present in it and its voxels have very high intensities, thus there have been a few methods presented for creating brain masks to remove the skull [Isensee et al., 2019] [Smith, 2002].

Bias field inhomogeneities correction is also something that is important to do before feeding the images to any segmentation tool since the presence of artifacts might create a great amount of false positives around the edges of white matter and gray matter.

Misalignment between different input images would generate unreliable results consequently image coregistration between the types of MRI images available is important to achieve good results.

Other pathologies may be present in different intensities ranges but in the case of WMH it appears in hyperintense areas or at least in intense areas where their surroundings are hypointense. Increasing the contrast between these hyperintense and hypointense areas will greatly improve the segmentation ability of CNNs. We use the method proposed in [Khademi et al., 2009] for enhancing contrast in FLAIR images.

Our previous study focuses on how combinations of these mentioned techniques work with each other [Duque et al., 2019]. Main conclusions from that study is that removing the skull, applying contrast enhancement and intensity standardization greatly benefit segmentation performance. In this work we further improve the cropping of slices.

2.2 Convolutional neural networks

Deep convolutional neural networks had a great success in a variety of problems like image classification [Krizhevsky et al., 2012] [Hershey et al., 2017] [Yu et al., 2017], text classification [Wang et al., 2018] [Johnson & Zhang, 2015], object instance segmentation [He et al., 2020], [Chen et al., 2019], malware classification [Kalash et al., 2018] [Gibert, 2016], sequence modelling [Bai et al., 2018] and medical image segmentation [Ronneberger et al., 2015] [Duque et al., 2019] [Enokiya et al., 2018] [Schlemper et al., 2019] [Li et al., 2018b] [Guerrero et al., 2018] [Christ et al., 2016].

Traditional computer vision techniques based in extracting features such as borders, corners, SIFT/SURF or any other custom feature to then train a machine learning model on those have been replaced by the effectiveness of convolutional neural networks. One of the first cases when CNN clearly achieved lower error than other solutions was in the classification of ImageNet [Krizhevsky et al., 2012] when they were able to lower the test set error rates from 45.7% from previous methods (SIFT + FVs) to 37.5% in the top-1 and from 25.7% to 17.0

Initial solutions to attempt to solve the problem of WMH segmentation were based on manual and tailored feature extraction which was then fed into some machine learnirng classifier such as a support vector machine, like the one shown in [Rincón et al., 2017] but most recent publications in the field have been focused on the used of CNN [Li et al., 2018a] [Li et al., 2018b] or

the preprocessing of the data to then feed it to a CNN [Duque et al., 2019] [Isensee et al., 2019].

The problem of image segmentation and specifically medical image segmentation has found great results by using U-shaped or V-shaped fully convolutional neural network architectures. These networks were named after such shapes, U-Net [Ronneberger et al., 2015] and V-Net [Milletari et al., 2016], which have seen multiple successors and modifications that improve their performance for certain use cases [Li et al., 2018a] [Oktay et al., 2018]. These architectures consist in a contracting path on the left which extracts most features on different levels of sizes and a reconstructing path that upsamples the image to be able to produce a full size segmentation output image as shown in the original diagram of the U-Net in Figure 1. The U-Net was originally presented with a weighted crossentropy as loss function, but later uses for segmentation generally include a Dice similarity coefficient as loss function since it has proven great results for this problem [Li et al., 2018a] [Duque et al., 2019].

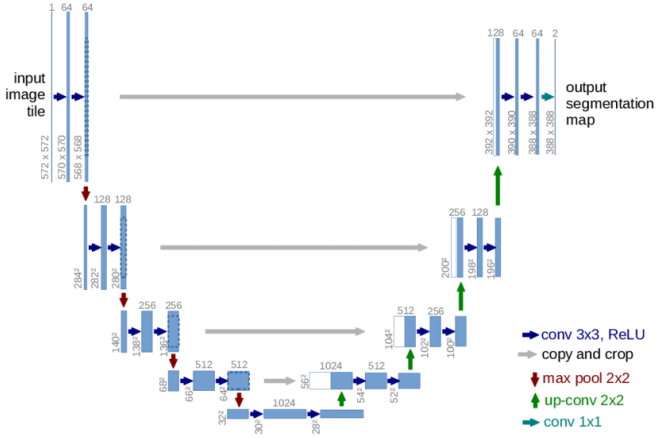


Figure 1: Original U-Net [Ronneberger et al., 2015]

Multiple of the top solutions in the WMH segmentation challenge base their solution on the U-Net architecture [Li et al., 2018a] [Kuijf et al., 2019].

2.3 Attention gates

Attention mechanisms have been around for some years but their usage has increased over time obtaining great results in different fields such as machine

translation [Vaswani et al., 2017], sentence classification [Liu et al., 2019], image classification [Wang et al., 2017], for generative adversarial networks [Zhang et al., 2019] and model interpretation [Serrano & Smith, 2020] [Wiegreffe & Pinter, 2020].

Attention has shown great ability to focus on specific features or parts of images to highlight the most important ones based on a context vector and therefore not focusing on the irrelevant features that may be present in the image. This is done by building a mask by combining lower level tensor which has a smaller size and better feature representation with the current level tensor. A full walkthrough of the attention gates structure is explained in apendix C.

One of the other reasons attention mechanisms have had so much success is their ability to be added as modules to existing network architectures improving their already great results. A good example of this is how attention gates were added to the U-Net architecture resulting in the Attention Gated U-Net [Oktay et al., 2018] improving the already field dominating results of U-Net [Ronneberger et al., 2015] architectures in medical image segmentation.

Attention gates build a mask to filter features in a image and then apply such mask to both highlight most usefull features while reducing the impact of non important ones.

3 Methodology

3.1 Dataset

We use the publicly available dataset from the MICCAI White Matter Hyperintensities challenge [Kuijf et al., 2019] to train and test our methods. It consists in 60 cases. For each subject, a 3D T1-weighted volume, and a 2D multi-slice FLAIR volume were provided. FLAIR images had the following acquisition characteristics: Utrecht (3T Philips Achieva, voxel size: 0.96 0.95 3.00 mm^3, image resolution: 240 240 48, TR/TE/TI: 11000/125/2800 ms), Singapore (3T Siemens TrioTim, voxel size: 1.00 1.00 3.00 mm^3, image resolution: 252 232 48, TR/TE/TI: 9000/82/2500 ms) and Amsterdam (3T GE Signa HDxt, voxel size: 0.98 0.98 1.20 mm^3, image resolution: 132 256 83, TR/TE/TI: 8000/126/2340ms). T1 and FLAIR images were aligned using elastix [Klein et al., 2010] [Shamonin et al., 2014] by the challenge organizers and bias correction was applied by using the SPM12 software

[Friston et al., 1994]. WMH were manually segmented by experts and this ground truth were used for training and testing. All subjects MRIs were applied a mask to remove the face and therefore the identity from the image.

Ground truth images contain a third label different from WMH and non WMH tagged as "other pathology" which was removed during the preprocessing stage for training.

Working solutions are submitted in a docker image [Merkel, 2014] and blind tested against a dataset containing unseen data from the three scanners that were available in the training set as well as data from two other unseen scanners.

Metrics evaluated on the blind test data are the following: dice, hausdorff distance (modified, 95th percentile), average volume difference (in percentage), sensitivity for individual lesions (in percentage), F1-score for individual lesions. A final metric is obtained by averaging the previous five in order to be able to rank all solutions submitted to the WMH challenge [Kuijf et al., 2019]. In appendix D all metrics are explained in detail.

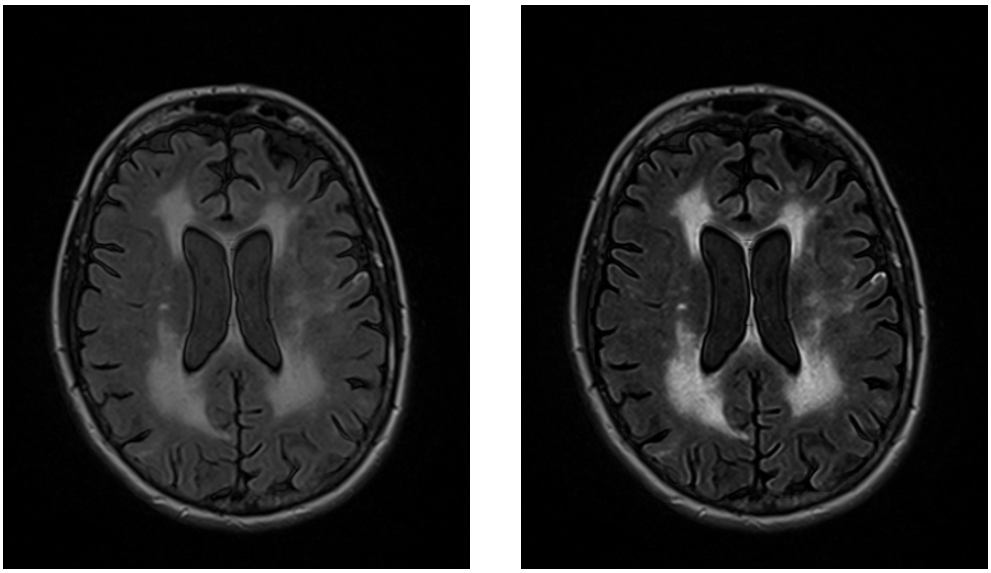
3.2 Preprocessing stages

The first step in the preprocessing is to generate the brain mask using HD-BET [Isensee et al., 2019] from the T1 image. In a previous work [Duque et al., 2019] we used the method explained in [Smith, 2002] but more recent work proposed in [Isensee et al., 2019] has proven better results basing their method en neural networks as well.

Then T1 images are co-registered to the FLAIR space to be able to generate a brain mask for FLAIR images as well. This was done using BRAINFit and BRAINResample modules from 3DSlicer [Fedorov et al., 2012].

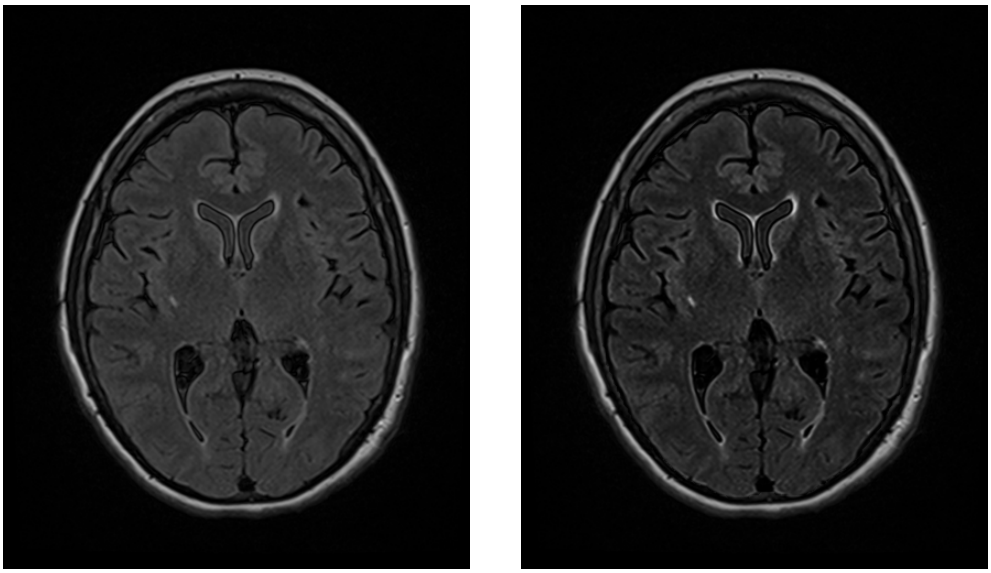
A non linear contrast enhancement transformation is run on the FLAIR image to enhance areas where WMH is present, this has shown an improvement in DICE compared to not using this technique [Duque et al., 2019]. Figure 2 and 3 show the result of applying contrast enhancement to both big and small lesions respectively. Brain masks are then applied to both T1 and FLAIR images by simply multiplying both matrices to the mask.

Some kind of normalization of voxel intensities is a must for gradient descent methods such as the ones applied in the optimization of CNNs in order to work properly, however standardization has proven to be a much better scaling technique before feeding images to CNNs [Duque et al., 2019].



(a) Original image (b) Contrast enhanced image

Figure 2: Contrast enhancement results for large lesions



(a) Original image (b) Contrast enhanced image

Figure 3: Contrast enhancement results for small lesions

All voxel values are standardized to a distribution with zero mean and unit variance.

For each subject, slices from the top and bottom part of the head were removed since the slices out of the brain cannot contain WMH and the top and bottom areas of the brain rarely contain them. We first remove all non brain slices by using the brain mask. Of the remaining slices we removed $\lceil n * 0.05 \rceil$ slices from the top and $\lceil n * 0.12 \rceil$ from the bottom where n is the number of axial view slices of each subject in the original full image. Images from all scanners were padded and cropped to achieve a slice shape of (200, 200).

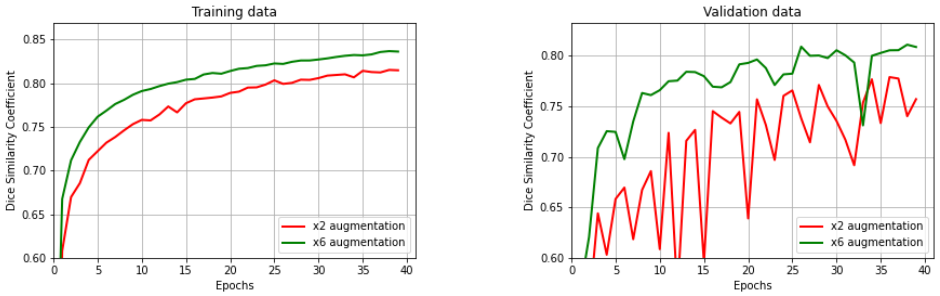
In previous solutions of our method we simply cropped the top and bottom slices based on a percentage of the total amount of axial view slices which worked well in the Utrecht, Singapore and Amsterdam datasets (3T Philips Achieva, 3T Siemens TrioTim and 3T GE Signa HDxt scanners respectively) but performed badly on other scanners. Our current solution is much more robust with a wider range of scanners that add more neck slices. Also it allows for a tailored crop for each patient, since our method works in two dimension each patient can have a different number of slices processed both for training and predictions.

3.3 Data augmentation

Medical datasets are generally small given how expensive it is to label images, therefore data augmentation is a great tool to enlarge the amount of training data available.

All slices that contain any level of WMH in the training set were augmented 6 times. The first five were done by applying random affine data transformations where values were picked from a normal distribution within the following ranges; for rotations with $[-30°, 30°]$ angles, shifts applied to both the x and y axis [-30%, 30%] of the total width and height, respectively, zoom on both axes with values in the ranges [0.9, 1.2] and shears in the range [-0.2, 0.2]. The sixth augmentation was done by applying all 5 transformations to each slice to further improve generalization.

A comparison of applying the same data transformations to the dataset but generating just 2 new slices per original slice against generating 6 as stated above can be shown in figure 4. Data augmentation does not only improves training metrics but it greatly improves validation metrics.



(a) DSC in the training set per epoch (b) DSC in validation set per epoch

Figure 4: Comparison between x2 augmentation and x6 augmentation. DSC values shown do not represent the performance of the final model but it is simply a comparison between the mentioned techniques.

3.4 Network architecture

The proposed solution uses a fully convolutional neural network based on the attention gated U-Net architecture [Schlemper et al., 2019], figure 5 shows the overall structure of the network. An overview of the main layers of CNNs is explained in appendix B.

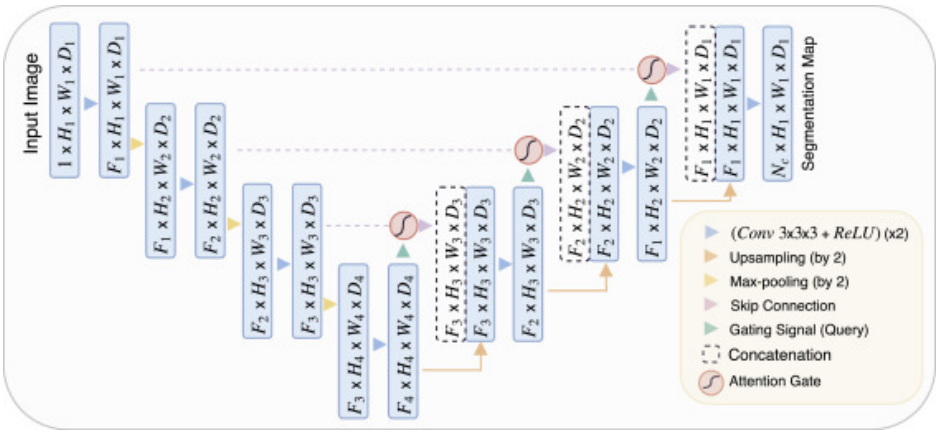
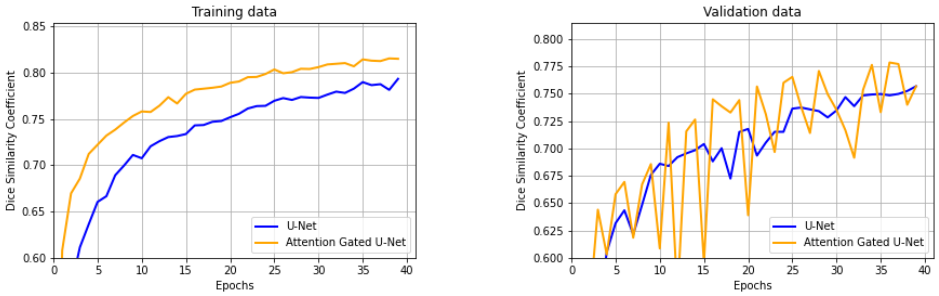


Figure 5: Original Attention U-Net [Oktay et al., 2018]

Our proposed attention gated U-Net has three levels instead of four (see Figure 1 for original U-Net), which means it has only three pooling layers in the contracting path and three transposed strided convolutions (instead



(a) DSC in the training set per epoch (b) DSC in the validation set per epoch

Figure 6: Comparison between our attention gated U-Net (blue) and our modified U-Net (orange/yellow). DSC values shown do not represent the performance of the final model but it is simply a comparison between the mentioned techniques.

of upsampling layers) in the expanding path. This allowed us to reduce the number of parameters while keeping the same performance since the focus on small features in obtained thanks to the increased convolution kernel size and an initial stack of convolutions. All convolutional kernels are increased to size 11 to capture richer local data besides convolutions within the attention gates. RELU activations are used in all convolution layers. Within the attention gates convolution kernels are (5, 5) besides the obvious (1, 1) convolutions. Pooling layers are kept of (2, 2). There is an initial stack of 5 convolutions before the U-Net pattern of two convolutions and pooling starts then that pattern is applied three times in the contracting path. Initialization of all convolutional kernels is done by using the He normal [He et al., 2015] besides in the attention gates in which we use the Glorot or Xavier uniform [Glorot & Bengio, 2010].

In the reconstruction path of the U-Net is where attention gates are located. Reconstruction upsampling are always done by using (2, 2) strided transposed convolutions (sometimes wrongly referred as deconvolutions) to match the (2, 2) pooling on the downsampling path. Attention gates use batch normalization at the end of their process.

We compared our previous solution which followed a U-Net architecture but without attention mechanisms and showed that attention gates boost performance in terms of DSC for white matter hyperintensities segmentation as it is shown in Figure 6.

3.5 Training

For most medical segmentation tasks the negative value of the Dice Similarity Coefficient (DSC) has become the standard loss function [Li et al., 2018a] [Duque et al., 2019] due to the very high imbalance that the classes within problems present [Li et al., 2018a] [Duque et al., 2019]. However this loss function has shown to struggle for certain metrics such as lesion-wise recall and F1 as it gives importance to big lesions and makes smaller ones irrelevant. When it comes to medical image segmentation lesion-wise recall and F1 are important, regardless of the size of the blobs. In late stages of training capturing small blobs makes no difference in terms of DSC and therefore blob wise recall and precision could be not well optimized if their size is not relevant.

In this work we propose a combination of a DSC and lesion blob F1 loss function. The idea behind this is to first optimize based on DSC and as this metric reaches better results start giving more weight to the F1 score. It is important to know that a metric based on F1 alone would not be possible as it is not differentiable and optimizing only for precision or recall could result in all white or black images as prediction. A comparison of DSC and F1 metrics for both loss functions can be found in figure 7

$$f1\_weighted\_DSC\_loss = -(DSC + F1 \times DSC \times weight)$$

Where weight is a fixed parameter which was set to 1.7 experimentally. Lower values of this weight made it just perform similarly to a regular DSC loss function. Lesion-wise recall and lesion wise precision are calculated using 2D connected components as units since our method is two dimensional as shown below:

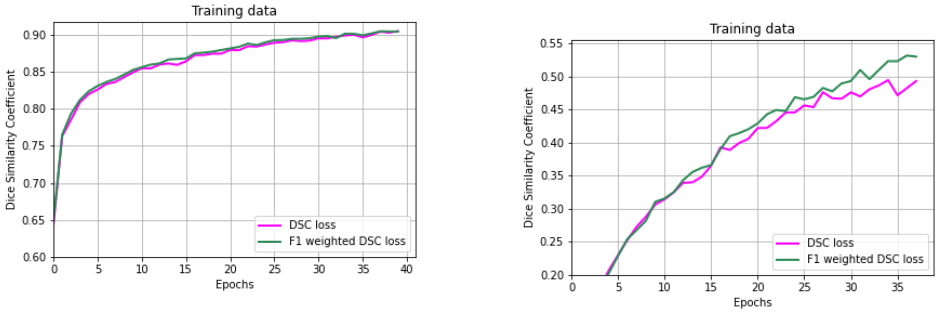
$$recall = \frac{|conn_G \times Pred|}{N_G}$$

$$precission = \frac{|conn_G \times Pred|}{N_{Pred}}$$

Where $conn_G$ represents the 2D connected components of the ground truth, then it is multiplied to the prediction and the cardinality of unique tags is obtained which leaves us with the number of correctly predicted individual lesions. N_G is the total amount of unique lesions in the ground truth and N_{Pred} is the total amount of unique lesions predicted. F1 score can be then calculated using precision and recall.

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

Network is trained for 40 epochs with a learning rate of 0.0001 with the Adam optimizer [Kingma & Ba, 2015]. All models were trained on a Nvidia GTX 1080. Learning rate is kept low as previous work has proven that higher learning rates makes training to stuck during the first few epochs [Duque et al., 2019] [Li et al., 2018a].



(a) DICE similarity coefficient per epoch in the training set.

(b) F1 score per epoch in the training set.

Figure 7: adaptive weighted loss function (seagreen) compared to DSC loss function (fuchsia).

For comparing own different methods a train, validation and test split patient wise was done on each dataset with a common seed across trainings. For each dataset containing 20 patients, 16 were used for training, 3 for validation and 1 for testing. Best performing model was then trained on the totality of the dataset.

3.6 Postprocessing

In order to produce an image which size matches the input all the cropping and padding has to be reversed. All slices are cropped or padded back using 0 value to their original shape. Removed slices are replaced with slices of value 0 by calculating again the gap between the original image and the brain mask. Finally adding as many slices that were removed from both the

top and the bottom of the image.

4 Results

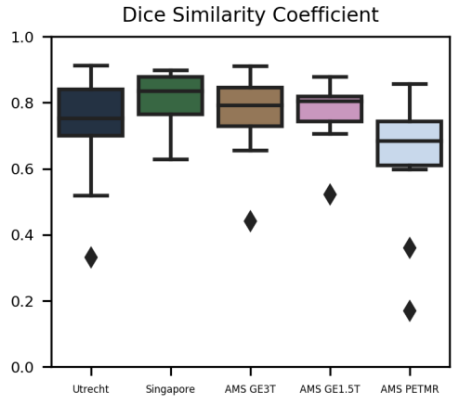
The results provided in this section are the ones produced by the blind test run by WMH challenge organizers [Kuijf et al., 2019]. Our method ranks 20th out of the 43 total submissions at the moment of writing this article. Specifically it excells in AVD, our best ranking metric, followed by Hausdorff distance and DSC.

	DSC	H95 (mm)	AVD (%)	recall	f1
Utrecht (n=30)	0.75	9.42	27.55	0.76	0.6
Singapore (n=30)	0.81	6.15	15.60	0.71	0.68
AMS GE3T (n=30)	0.78	6.12	18.93	0.71	0.69
AMS GE1.5T (n=10)	0.77	10.58	13.98	0.73	0.74
AMS PETMR (n=10)	0.63	22.29	116.95	0.79	0.33
weighted average	0.76	8.90	28.83	0.73	0.63
rank [0..1]	0.079	0.074	0.016	0.220	0.230

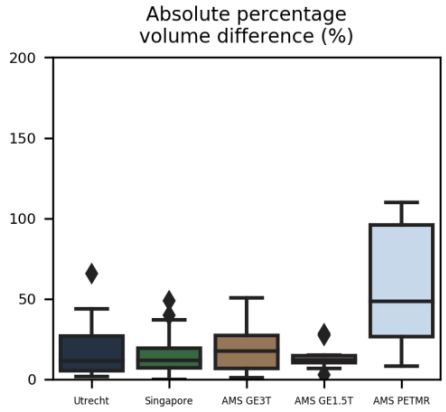
Table 1: Results from the blind test.

For DSC our solution achieves a weighted average of 0.76, performing best for the Utrecht dataset and worst for the AMS PETMR. For H95 we reach an average of 8.90, being GE3T our best dataset for this metric with 6.12 and 22.29 for AMS PETMR as our worst case. In the case of AVD our solutions does best at AMS GE1.5T with 13.98 and worst for AMS PETMR with 116.95, being in this case a extreme outlier compared to the rest of datasets. For recall we perform best for AMS PETMR with 0.79 but in this case is a sign of oversegmentation and that is why the rest of metrics are worse for this dataset. Our best F1 metric is for the AMS GE1.5T dataset with 0.74 and our worst for AMS PETMR with 0.33 as part of the oversegmentation problem stated before.

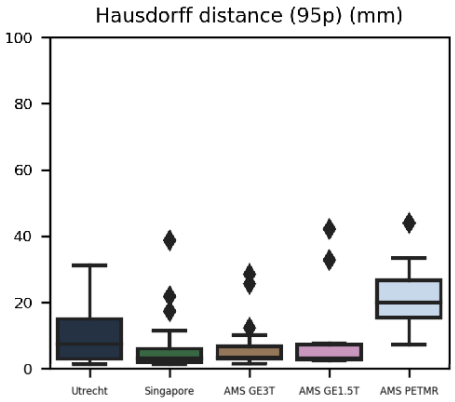
As it can be seen on both Table 1 and Figure 8 our method performs great across all scanners on DSC. But when it comes to the rest of the metrics, except for recall, it underperforms for the AMS PETMR dataset due to the oversegmentation it produces.



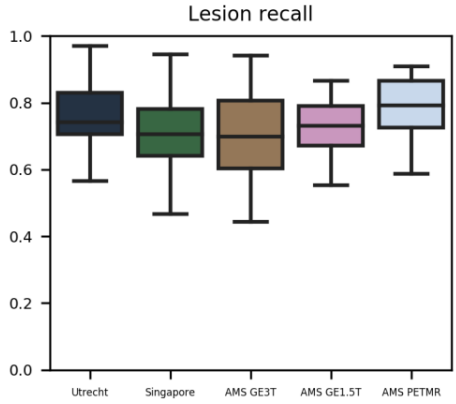
(a) Dice Similarity Coefficient



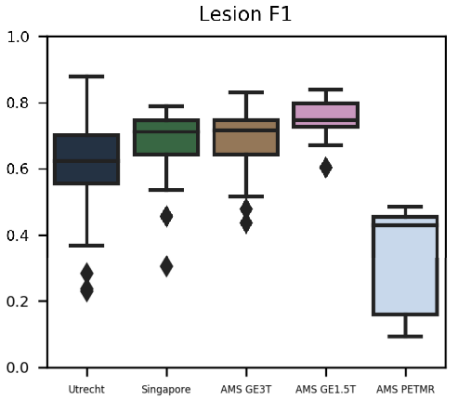
(b) Absolute percentage volume difference



(c) Hausdorff distance



(d) Lesion recall



(e) Lesion F1

Figure 8: Boxplots for all five metrics for each of the datasets tested against.

Our solutions performs best in absolute percentage volume difference and hausdorff distance (95p) excelling in the Utrech, Singapore, AMS GE3T and AMS GET 1.5T datasets in terms of the ranking provided by the challenge.

In Figure 10 shows segmentation examples for three subjects in the three available datasets. Segmentation is good in many areas but it still strugles with some of the small blobs as well as certain edges of bigger ones.

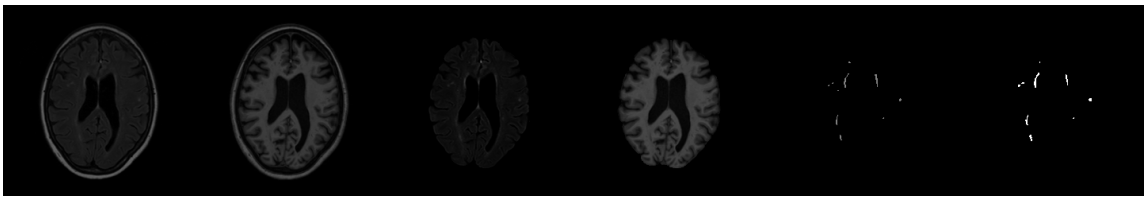
5 Discussion and future work

There is still room for improvements in a few areas. First of all our solution does not work great on all scanners and it underperforms in the AMS PETMR where it oversegments getting too many false positives. This could be caused by multiple factors like the preprocessing either not removing correctly neck or skull areas, the contrast enhancement could be over-highlighting parts that are not WMH or the model is creating the false positives itself by being overfitted to the other tree scanner types. Artifacts could also be a cause of oversegmentation in this specific scanner.

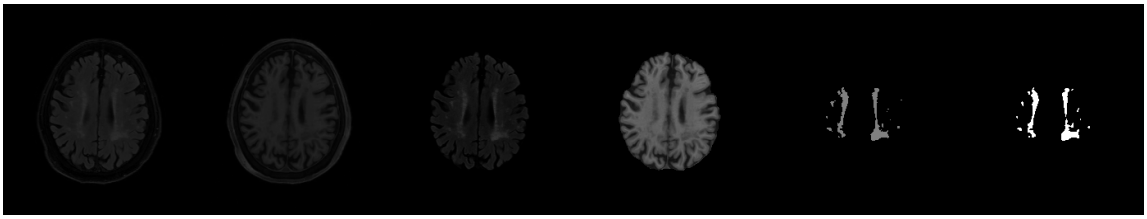
F1 and recall lesion metrics are the other two in which we could find improvements for our current solution. In order to improve recall a greater amount of small lesions need to be targeted, as bigger lesions are generally reached. However, this has to be balanced with not over-segmenting since we are also trying to improve F1 by reducing the false positives that we currently have. Giving more weight in our loss function to F1 part of it during late stages of the training could be an option. Adaptive weighted loss function has a great potential to deliver great results in segmentation tasks but more research will be conducted along this line to explore how to better optimize it.

Reviewing that the preprocessing never fails to highlight small lesions could be another path into improving recall. Since failing to properly preprocess lesions of all size would result in only segmenting big lesions and therefore bad recall and F1 results.

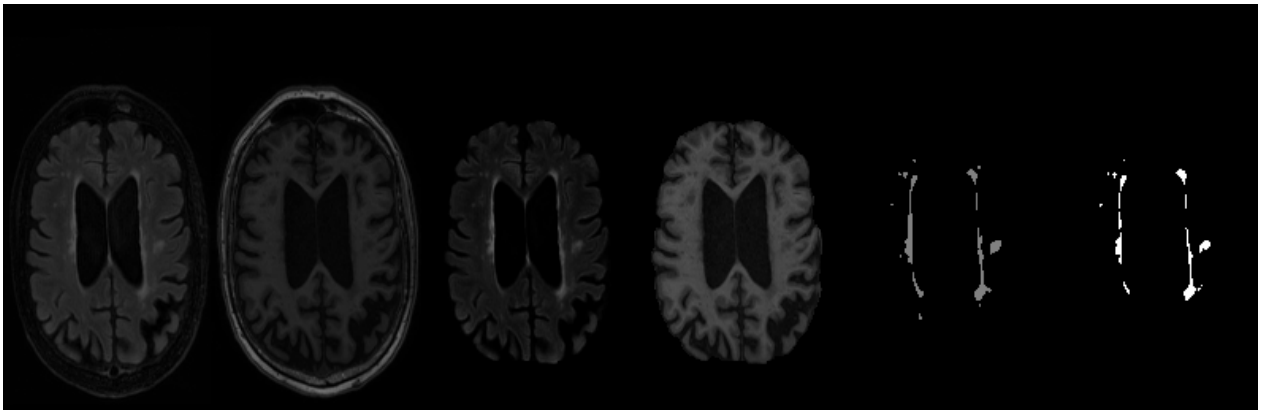
It is a non trivial problem to find the best architecture and set of hyperparameters for very deep neural networks as the one we are using in this problem. Newer techniques such as Network Architecture Search (NAS) could help us find an even more optimal architecture for this specific problem and datasets. This would allow us to further explore better optimization for all metrics including the ones that our method already performs well.



(a) Patient from Utrecht dataset

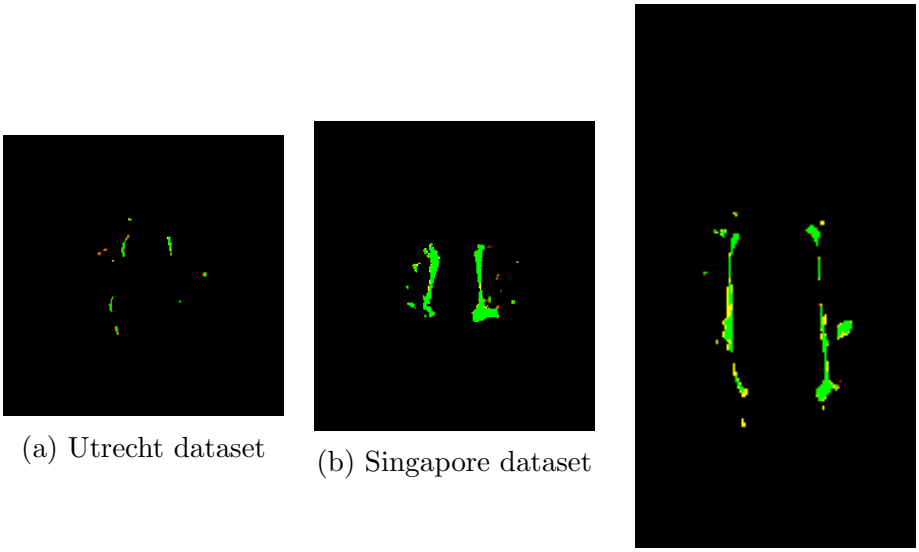


(b) Patient from Singapore dataset



(c) Patient from GE3T dataset

Figure 9: Preprocessing and segmentation results for holdout patients in each of the datasets. From left to right, original FLAIR, original T1, processed FLAIR, processed T1, ground truth and model segmentation.



(a) Utrecht dataset (b) Singapore dataset (c) Amsterdam dataset

Figure 10: Examples of segmentation results for each of same three hold out patients shown in figure 9. Green shows true positives, red means false positives and yellow false negatives.

Generally solutions perform better on average when an ensemble of networks is used instead of a single one, this helps reduce both false negatives and false positives. Using the same network in an ensemble of 3 or 5 units and then using voting mechanisms to get final predictions could improve overall results across different metrics.

6 Conclusion

In conclusion we found that preprocessing is key for achieving good results in WMH. Data augmentation does also benefit the improvement of the results. Our tailored slices cropping improves the performance of our method across scanners.

U-Net architectures clearly can achieve great results for WMH segmentation. On top of that Attention Gated U-Net can even improve such results even further. Fully optimizing an architecture to a specific problem is very challenging and our research will continue also in this direction.

DICE similarity coefficient is a great choice for loss function in segmentation tasks but it lacks the ability to take in consideration small lesions which might be irrelevant in terms of amount of voxels. Our proposed adaptive weighted loss function has all the benefits from a regular DSC loss function but also takes into account smaller lesions once it reaches a good optimization in terms of DSC. However, further research has to be conducted along this line to fully optimize and explore this kind of loss function potential.

References

[Abadi et al., 2015] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[Bai et al., 2018] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.

[Chen et al., 2019] Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., Loy, C. C., & Lin, D. (2019). Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Christ et al., 2016] Christ, P. F., Elshaer, M. E. A., Ettlinger, F., Tatavarty, S., Bickel, M., Bilic, P., Rempfler, M., Armbruster, M., Hofmann, F., D'Anastasi, M., Sommer, W. H., Ahmadi, S. A., & Menze, B. H. (2016). Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random fields. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

[Dumoulin & Visin, 2016] Dumoulin, V. & Visin, F. (2016). A guide to convolution arithmetic for deep learning. *ArXiv e-prints*.

[Duque et al., 2019] Duque, P., Cuadra, J. M., Jiménez, E., & Rincón-Zamorano, M. (2019). Data Preprocessing for Automatic WMH Segmentation with FCNNs. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11487 LNCS (pp. 452–460).: Springer Verlag.

[Enokiya et al., 2018] Enokiya, Y., Iwamoto, Y., & Chen, Y.-w. (2018). Automatic Liver Segmentation Using U-Net with Wasserstein GANs. 6(2), 152–159.

[Fedorov et al., 2012] Fedorov, A., Beichel, R., Kalpathy-Cramer, J., Finet, J., Fillion-Robin, J. C., Pujol, S., Bauer, C., Jennings, D., Fennessy, F., Sonka, M., Buatti, J., Aylward, S., Miller, J. V., Pieper, S., & Kikinis, R. (2012). 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magnetic Resonance Imaging*.

[Fourcade & Khonsari, 2019] Fourcade, A. & Khonsari, R. (2019). Deep learning in medical image analysis: A third eye for doctors. *Journal of Stomatology, Oral and Maxillofacial Surgery*, 120(4), 279–288.

[Friston et al., 1994] Friston, K. J., Holmes, A. P., Worsley, K. J., Poline, J. ., Frith, C. D., & Frackowiak, R. S. (1994). Statistical parametric maps in functional imaging: A general linear approach. *Human Brain Mapping*.

[Gibert, 2016] Gibert, D. (2016). Convolutional Neural Networks for Malware Classification. *University Rovira i Virgili, Tarragona, Spain*.

[Glorot & Bengio, 2010] Glorot, X. & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Journal of Machine Learning Research*.

[Goyal et al., 2020] Goyal, M., Goyal, R., Venkatappa Reddy, P., & Lall, B. (2020). Activation functions. In *Studies in Computational Intelligence*, volume 865 (pp. 1–30). Springer Verlag.

[Guerrero et al., 2018] Guerrero, R., Qin, C., Oktay, O., Bowles, C., Chen, L., Joules, R., Wolz, R., Valdés-Hernández, M. C., Dickie, D. A., Wardlaw, J., & Rueckert, D. (2018). White matter hyperintensity and stroke le-

sion segmentation and differentiation using convolutional neural networks. *NeuroImage: Clinical*, 17, 918–934.

[He et al., 2020] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2020). Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2).

[He et al., 2015] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*.

[Hershey et al., 2017] Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R. J., & Wilson, K. (2017). CNN architectures for large-scale audio classification. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*.

[Isensee et al., 2019] Isensee, F., Schell, M., Pflueger, I., Brugnara, G., Bonekamp, D., Neuberger, U., Wick, A., Schlemmer, H. P., Heiland, S., Wick, W., Bendszus, M., Maier-Hein, K. H., & Kickingereder, P. (2019). Automated brain extraction of multisequence MRI using artificial neural networks. *Human Brain Mapping*.

[Johnson & Zhang, 2015] Johnson, R. & Zhang, T. (2015). Effective use of word order for text categorization with convolutional neural networks. In *NAACL HLT 2015 - 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*.

[Kalash et al., 2018] Kalash, M., Rochan, M., Mohammed, N., Bruce, N. D., Wang, Y., & Iqbal, F. (2018). Malware Classification with Deep Convolutional Neural Networks. In *2018 9th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2018 - Proceedings*.

[Khademi et al., 2009] Khademi, A., Venetsanopoulos, A., & Moody, A. (2009). Automatic contrast enhancement of white matter lesions in flair MRI. In *Proceedings - 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, ISBI 2009*.

[Kingma & Ba, 2015] Kingma, D. P. & Ba, J. L. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

[Klein et al., 2010] Klein, S., Staring, M., Murphy, K., Viergever, M. A., & Pluim, J. P. (2010). Elastix: A toolbox for intensity-based medical image registration. *IEEE Transactions on Medical Imaging*, 29(1).

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*.

[Kuijf et al., 2019] Kuijf, H. J., Casamitjana, A., Collins, D. L., Dadar, M., Georgiou, A., Ghafoorian, M., Jin, D., Khademi, A., Knight, J., Li, H., Lladó, X., Biesbroek, J. M., Luna, M., Mahmood, Q., Mckinley, R., Mehrtash, A., Ourselin, S., Park, B. Y., Park, H., Park, S. H., Pezold, S., Puybareau, E., De Bresser, J., Rittner, L., Sudre, C. H., Valverde, S., Vilaplana, V., Wiest, R., Xu, Y., Xu, Z., Zeng, G., Zhang, J., Zheng, G., Heinen, R., Chen, C., Van Der Flier, W., Barkhof, F., Viergever, M. A., Biessels, G. J., Andermatt, S., Bento, M., Berseth, M., Belyaev, M., & Cardoso, M. J. (2019). Standardized Assessment of Automatic Segmentation of White Matter Hyperintensities and Results of the WMH Segmentation Challenge. *IEEE Transactions on Medical Imaging*, 38(11).

[Li et al., 2018a] Li, H., Jiang, G., Zhang, J., Wang, R., Wang, Z., Zheng, W. S., & Menze, B. (2018a). Fully convolutional network ensembles for white matter hyperintensities segmentation in MR images. *NeuroImage*, 183, 650–665.

[Li et al., 2018b] Li, H., Zhang, J., Muehlau, M., Kirschke, J., & Menze, B. (2018b). Multi-Scale Convolutional-Stack Aggregation for Robust White Matter Hyperintensities Segmentation.

[Liu et al., 2019] Liu, Y., Ji, L., Huang, R., Ming, T., Gao, C., & Zhang, J. (2019). An attention-gated convolutional neural network for sentence classification. *Intelligent Data Analysis*, 23(5).

[Merkel, 2014] Merkel, D. (2014). Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239), 2.

[Milletari et al., 2016] Milletari, F., Navab, N., & Ahmadi, S. A. (2016). V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*.

[Oktay et al., 2018] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., Glocker, B., & Rueckert, D. (2018). Attention U-Net: Learning Where to Look for the Pancreas.

[Rincón et al., 2017] Rincón, M., Díaz-López, E., Selnes, P., Vegge, K., Altmann, M., Fladby, T., & Bjørnerud, A. (2017). Improved Automatic Segmentation of White Matter Hyperintensities in MRI Based on Multilevel Lesion Features. *Neuroinformatics*, 15(3).

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., & Brox, T. (2015). Unet. *MICCAI2015*, 9351, 234–241.

[Schlemper et al., 2019] Schlemper, J., Oktay, O., Schaap, M., Heinrich, M., Kainz, B., Glocker, B., & Rueckert, D. (2019). Attention gated networks : Learning to leverage salient regions in. *Medical Image Analysis*, 53, 197–207.

[Serrano & Smith, 2020] Serrano, S. & Smith, N. A. (2020). Is attention interpretable? In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*.

[Shamonin et al., 2014] Shamonin, D. P., Bron, E. E., Lelieveldt, B. P., Smits, M., Klein, S., & Staring, M. (2014). Fast parallel image registration on CPU and GPU for diagnostic classification of Alzheimer's disease. *Frontiers in Neuroinformatics*, 7(JAN).

[Smith, 2002] Smith, S. M. (2002). Fast robust automated brain extraction. *Human Brain Mapping*.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-December.

[Wang et al., 2017] Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., & Tang, X. (2017). Residual attention network for image classification. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-January.

[Wang et al., 2018] Wang, S., Huang, M., & Deng, Z. (2018). Densely connected CNN with multi-scale feature attention for text classification. In *IJCAI International Joint Conference on Artificial Intelligence*, volume 2018-July.

[Wardlaw et al., 2015] Wardlaw, J. M., Valdés Hernández, M. C., & Muñoz-Maniega, S. (2015). What are white matter hyperintensities made of? Relevance to vascular cognitive impairment.

[Wiegreffe & Pinter, 2020] Wiegreffe, S. & Pinter, Y. (2020). Attention is not not explanation. In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*.

[Yu et al., 2017] Yu, S., Jia, S., & Xu, C. (2017). Convolutional neural networks for hyperspectral image classification. *Neurocomputing*.

[Zeiler et al., 2010] Zeiler, M. D., Krishnan, D., Taylor, G. W., & Fergus, R. (2010). Deconvolutional networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 2528–2535).

[Zhang et al., 2019] Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). Self-attention generative adversarial networks. In *36th International Conference on Machine Learning, ICML 2019*, volume 2019-June.

Appendices

A Magnetic resonance imaging

MRI scanners use powerful magnets to produce a very strong magnetic field that makes protons in the body to align with that field, the protons are

stimulated and spin out of equilibrium straining against the pull of magnetic field. Then the MRI sensors are able to detect the energy released as the protons realign with the magnetic field [**?**]. Depending on the environment and chemical aspects of the molecules the time it takes the protons to realign and the amount of energy released may vary.

MRI machines can produce a set of 2D images which if the distance between them is short, can be seen as 3D images.

Although there is a wide variety of MRI types, in this work we use fluid attenuated inversion recovery (FLAIR) and T1-weighted images. FLAIR is a special kind of inversion recovery sequence with a long inversion time which removes the cerebrospinal fluid (CSF) in the resulting images. In FLAIR images, grey matter could appear brighter than white matter and CSF is dark/black. This is the most useful kind of MRI for detecting WMH as it appears hyper intense. The timing of radiofrequency pulse sequences used to make T1 images produces them to highlight fat tissue in the body and for this specific purpose it is great for differentiating tissues in the brain.

B Convolutional neural networks

Convolutional neural networks are a main component of the Deep Learning field. They rely on a set of different layers which mainly are convolutional layers and pooling layers. The learning and optimization process occurs in the convolutional layer.

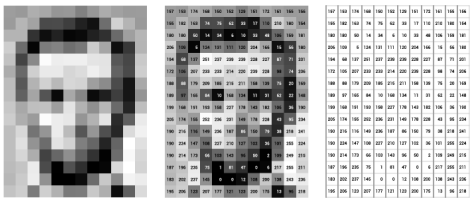
This kind of neural networks work very well with images because they rely on the matrix shape of images to apply discrete convolution as their main operation, however they can be applied to any kind of input data as long as its first reshaped to fit into the network.

B.1 Image representation

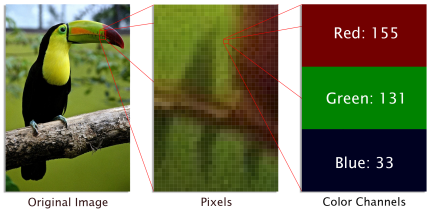
Images are stored in memory as N-channels matrices, where a gray image would simply have one channel being each pixel value the intensity of that pixel and color images would have generally 3 channels each one representing a different color, and the pixels of each channel representing the intensity of that color. Most common color image representation is RGB (red, green, blue) however there are other representations such as HSL (hue, saturation, lightness) and HSV (hue, saturation, value). Actual values can have a variety

of ranges depending of the data type of the image, but a very common one is for it to have [0, 255] range.

In 11a a representation of a gray image is shown and 11b explains an RGB image.



(a) Gray image representation



(b) RGB image representation

Figure 11: General cases of image representation and storage.

MRI are single channel images and we use FLAIR and T1 images types which are stack as channels for the network input, therefore our network input is composed of two single channel images that creat a 2-channel image.

B.2 Convolution layer

Convolution layers can have an input of a image of shape $H \times W$ with any number of channels, which leaves us with an image of $H \times W \times C$. They then apply a convolution on a kernel of $K_h \times K_w$ to generate the output of the layer, even though generally kernels are squared (for 2 dimensional convolutions) it is not technically enforced.

Two dimensional convolution is defnied as:

$$y[i,j] = \sum_{h=-\infty}^{\infty} \sum_{w=-\infty}^{\infty} K[h,w]\dot{I}[i-h, j-w]$$

Where y is the output image, i and j are the coordinates of the image, K refers to convolution kernel and I is input image.

To illustrate this with a example, if the kernel has size 3×3 then the indices would range from -1 to 1 as shown in the formula below:

$$y[i,j] = K[-1,-1]I[i+1,j+1] + K[-1,0]I[i+1,j] + K[-1,1]I[i+1,j-1] + K[0,-1]I[i,j+1] + K[0,0]I[i,j] + K[0,1]I[i,j-1] + K[1,-1]I[i-1,j+1] + K[1,0]I[i-1,j] + K[1,1]I[i-1,j-1]$$

Kernels is moved throughout the image as a sliding window convolving at each of the pixels. This way it generates an new output image. Figure 12 shows an illustration of the convolution formulas with a graphical example. The values of this kernels are parameters of the network and therefore what is optimized during training.

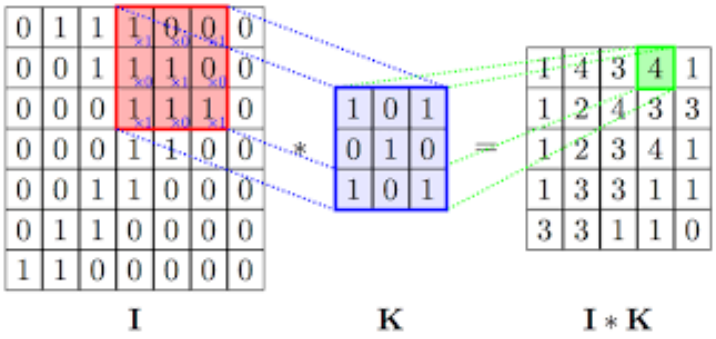


Figure 12: Convolution operation

Convolutional layers are highly customizable and the main parameters in which one can modify it are number of filters, kernel size, strides, padding, activation and kernel initializers. The number of filters/kernels is just the number of different filters that will be involved in the layer and its size the size of all of the kernels in such layer. The other parameters are explained below.

B.2.1 Padding

If no padding is applied to the convolution, the result image will be smaller than the input image as shown in 12, more specifically it will be reduced by $2\lfloor \frac{K}{2} \rfloor$ on both dimensions (assuming the kernel is squared).

Generally deep learning frameworks such as Tensorflow [Abadi et al., 2015] will have two options for this parameter, "valid" will apply no padding at all and "same" will make the necessary padding (top, bottom, left and right) to mantain the original image input size.

B.2.2 Strides

Stride is the number of pixels shifts over the input matrix. Non strided convolution works as explained before, what is also referred to as a stride of value 1. N value strides make to shift the kernel N pixels each time a convolution is applied, the greater N is the smaller the output matrix will be. A graphical representation of strided convolution with strides value 2 can be seen in Figure 13

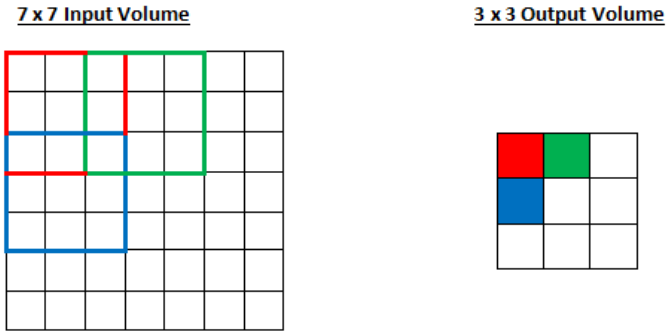


Figure 13: Example of convolutions with strides value 2.

B.2.3 Activation layer

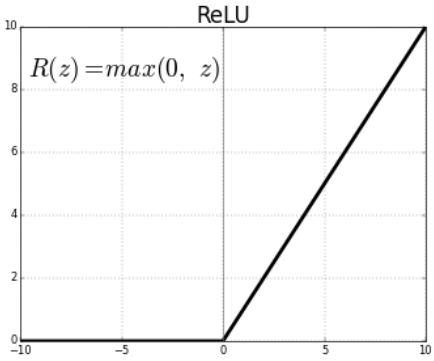
After the convolution is applied on the data a non linear activation function is applied to modify the output of the layer. There are a great amount of activation functions [Goyal et al., 2020] that have been used in different neural networks for many use cases, but one of the most common ones for convolutional layers is the rectified linear.

$$ReLu(z) = max(0, z)$$

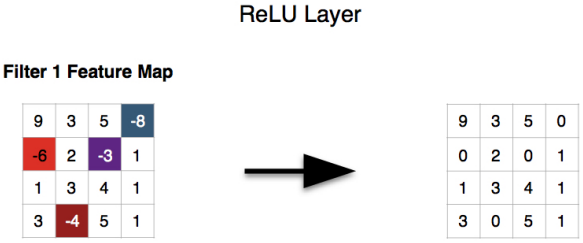
A graphical view of the function as well as an example of an output and its corresponding input can be found in Figure 14.

B.2.4 Kernel initializer

Kernels are optimized during the training process but they need to be initialized before training starts. This initialization is made randomly but there are an infite ways of initializing them. The one that was used for



(a) ReLu function



(b) Example of input and output of a ReLu activation function.

Figure 14: Rectified Linear Unit (ReLU)

the original U-Net [Ronneberger et al., 2015] and that has shown great results [Li et al., 2018a] [Duque et al., 2019] in very deep convolutional neural networks is he_normal [He et al., 2015].

B.2.5 Transposed convolution

Transposed convolution is the transposed operation of the convolution, it is intended to undo or reverse the effect of the convolution. It was initially proposed in [Zeiler et al., 2010] and has been wrongly referred to as deconvolution in its initial work and other ocasions. The literature and also the deep learning frameworks that implement this layer have been correcting this naming since.

Many of the parameters that can be configured in the convolution layer can be also modified for the transposed convolution layer but they have a different effect. For example strided transposed convolutions will make output image greater than the input image, as opposed to strides in regular convolution which would make the output image compared to the input. Strided transposed convolution layers are used in this work in replacement of upsampling layers since they add a learning component to the upsampling in exchange for computational cost (it adds more parameters to the network).

For a great visual resource for visualizing convolution and transposed convolution with its many configurations of parameters please refer to [Dumoulin & Visin, 2016]

B.3 Pooling layer

Pooling layers are used to reduce the size of the image and therefore the amount of data/features in them, in a selective way. Pooling is applied to each subsection of the image in which an operation is made to generate the output pixel for this section. The size of this subsection is given by the value specified for this parameter, all pooling layers in this work were 2×2.

The most common operations for pooling layers are max, min, mean, median and sum but there is no technical restriction and other operations could be applied. Figure 15 shows an example of an average and max pooling, the latter is the one used in this work for all pooling layers since our objective is to keep the pixels with the highest intensities in order to segment white matter hyperintensities.

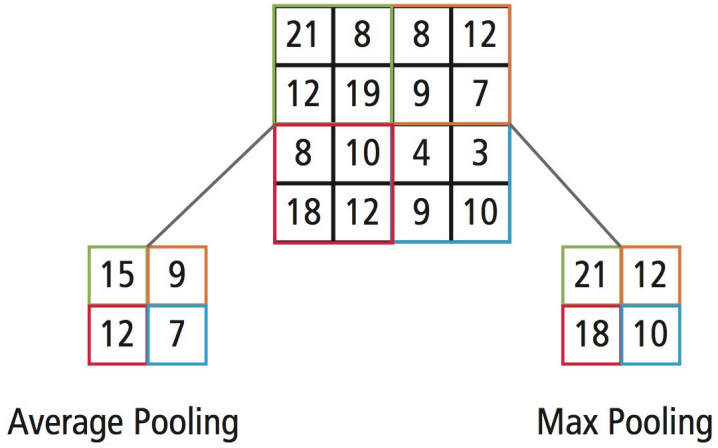


Figure 15: Average and max pooling operation

The other operations would be applied in the same way shown in 15 but with the new operation of choice.

C Attention gates

Attention is a way to highlight certain activations of a network and discard the ones that are less relevant for the problem at hand. There are two types of attention, hard and soft, in this work we only make use of the latter. Hard attention can only focus on one part of the image and is non

differentiable and therefore needs to be applied along with other learning component such as reinforcement learning. Soft attention on the other hand is in fact differentiable and can highlight multiple parts of the image, which means in can be back propagated in a neural network.

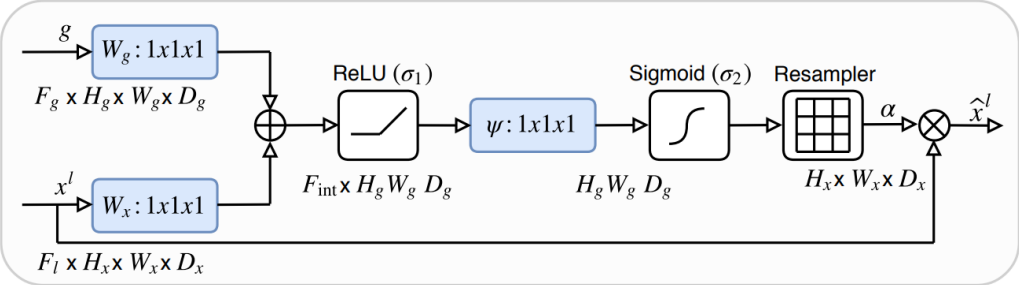


Figure 16: Attention gate diagram [Oktay et al., 2018].

Following the naming in Figure 16, attention gates take two inputs, the original tensor x and the tensor g from the next lowest layer of the network. Since g comes from a lower layer has a smaller dimensions and better feature representation.

First x goes through a strided convolution which reduces its size to the same as g, the latter goes through a 1x1 convolution to adapt de number of filters in a way that it fits the ones x has, once both steps are done they are added element wise. This produces aligned weights to be larger while unaligned become smaller which is then passed through a ReLu activation to remove negative values. Since we are looking to have a single channel for this mask, we will do make a 1x1 convolution to colapse all channels into one. The last step to produce the attention coefficients is to scales the range of the values to [0, 1] where values close to 1 will indicate relevant features and values close to 0 the opposite. This is done with the help of a sigmoid layer.

Once attention coefficients are ready they are upsampled to have the same size as the original input x and then they are both multiplied elementwise which produces the filtering based in the relevance found in the attention.

D WMH evaluation

In this section G refers to the ground truth and $Pred$ to the prediction mask.

D.1 Dice Similarity Coefficient

The DSC is one of the main metrics in this type of segmentation problems and the negative value of the DSC (DSC loss function) has become one of the most used loss functions for this type of tasks [Li et al., 2018a] [Duque et al., 2019] [Oktay et al., 2018].

$$DSC = \frac{2 \times |G \cap Pred|}{|G| + |Pred|}$$

$$DSCloss = -\frac{2 \times |G \cap Pred|}{|G| + |Pred|}$$

Where the intersection between the ground truth and the prediction is done by multiplying both prediction and ground truth mask, the number of voxels in this intersection is counted, multiplied by two to then be divided by the amount of voxels in G plus the ones in the prediction.

D.2 Hausdorff distance (95th percentile)

$$H(G,P) = max\{\underset{x \in G^{y \in Pred}}{supinfd(x,y)}, \underset{y \in G^{x \in G}}{supinfd(x,y)}\}$$

In the challenge it is used a robust version that calculates the 95th percentile instead of the maximum which would be the 100th percentile.

D.3 Average volume difference

Average volume differences is a comparison between the amount of voxels that are marked as WMH in the prediction mask and the ones present in the ground truth regardless if they are a true or false positive.

$$AVD = \frac{|V_G - V_{Pred}|}{V_G}$$

Where $|x|$ represents in this case absolute value and not cardinality. It is a great sanity check for comparing the number of voxels predicted as WMH

compared to the gold standard. Systems that overpredict would be equally penalized that the ones that underpredict assuming the difference is the same.

D.4 Recall for individual lesions

Recall is calculated based on individual lesions and not on individual voxels, therefore first the individual lesions have to be calculated, this is done with 3D connected components. The number of individual lesions that are correctly detected is divided by the number of lesions present in the ground truth image. Note that a single voxel of intersection would consider the lesion as correctly detected.

$$recall = \frac{|conn_G \times Pred|}{N_G}$$

Where $conn_G$ is the connected components and $|x|$ represents cardinality in this case. The connected components matrix is then projected into the binary prediction matrix. The number of distinct correctly detected connected components is then divided by the total amount of connected components present in the ground truth.

D.5 F1 for individual lesions

F1 for individual lesions is calculated in a similar fashion as recall but for the F1 score.

$$F1 = \frac{|conn_G \times Pred|}{N_{Pred}}$$

Where N_{Pred} is the total number of predicted individual lesions. Note that the cardinality is calculated only taking into account the amount of unique lesions.

E Code

The code that is submitted to the WMH challenge is provided in the following repository and will be added to the SIMDA research group website.

`https://github.com/pabloduque0/WMH_AttGatedUnet_CustomLoss`

Commands to run and execute predictions for new images are provided in the README of the repository.

F Previous work

Data Preprocessing for Automatic WMH Segmentation with FCNNs

P. Duque( ), J. M. Cuadra, E. Jiménez, and Mariano Rincón-Zamorano

Departamento Inteligencia Artificial, UNED, Madrid, Spain
pablo.duque55@gmail.com, {jmcuadra,esterjimenez,mrincon}@dia.uned.es, http://simda.uned.es/

Abstract. Automatic segmentation of brain white matter hyperintensities (WMH) is a challenging problem. Recently, the proposals based on Fully Convolutional Neural Networks (FCNN) are giving very good results, as it is demostrated by the top WMH challenge architectures. However, the problem is non completely solved yet. In this paper we analyze the influence of preprocessing stages of the input data on a fully convolutional network (FCNN) based on the U-NET architecture. Results demostrate that standarization, skull stripping and contrast enhancement significantly influence the results of segmentation.

Keywords: White matter hyperintensities · Fully Convolutional Neural Networks · U-NET · Contrast enhancement · Normalization · Standardization

1 Introduction

The presence of leukoaraiosis or white matter hyperintensities (WMH) in the brain of elderly individuals is linked to increased risk of stroke, cognitive impairment, dementia and ultimately, death. Magnetic resonance imaging (MRI) is by far the most sensitive modality for detecting WMH and MRI is consequently a very central diagnostic procedure in the elderly population. Manual WMH segmentation is very time-consuming and prone to user-bias, which has resulted in several attempts to generating automated analysis tools for WMH segmentation [1–3].

Recently, solutions based on Fully Convolutional Networks (FCNN) are giving very good results as shown by the first positions in the WMH challenge [3]. Nevertheless, there are still problems to solve such as the great inter- and intra-observer variability, so a systematic study of the phases of the problem solution is necessary. In this context, this paper focuses on the analysis of input data preprocessing and its influence on a FCNN based on the U-Net [4].

© Springer Nature Switzerland AG 2019
J. M. Ferrández Vicente et al. (Eds.): IWINAC 2019, LNCS 11487, pp. 452–460, 2019.
https://doi.org/10.1007/978-3-030-19651-6_44

2 Materials and Methods

2.1 Dataset

In all reported experiments, we relied on the publicly available dataset from the MICCAI WMH Challenge [3], organized as a joint effort of the UMC Utrecht, VU Amsterdam and NUHS Singapore for benchmarking methods for automatic WMH segmentation. It consists in 60 cases, 20 from each one of the three centres. For each subject, a 3D T1-weighted volume, and a 2D multi-slice FLAIR volume were provided. FLAIR images had the following acquisition characteristics: Utrecht (3T Philips Achieva, $0.96 \times 0.95 \times 3.00$, $240 \times 240 \times 48$), Singapore (3T Siemens TrioTim, $1.00 \times 1.00 \times 3.00$, $252 \times 232 \times 48$) and Amsterdam (3T GE Signa HDxt, $0.98 \times 0.98 \times 1.20$, $132 \times 256 \times 83$). T1 and FLAIR images were aligned using elastix [5,6] and bias correction was applied by using the SPM12 software [7]. WMH were manually segmented by experts and this masks were used for training and testing.

All slices were set to 240×240. Slices were conveniently cropped or padded to keep the center of the image. Top and bottom slices are removed to reduce noise since there is no white matter in such slices of the brain. We opted to remove the bottom 6 slices and the top 4 slices.

2.2 Preprocessing

Apart from the initial basic preprocessing performed by the WMH challenge organizers, we aimed to analyze how different transformations of the input data impact and facilitate machine learning with FCNNs.

Skull Stripping. The MRI modalities currently used to segment WMH are (1) FLAIR, where WMH appear as hyperintensities, and (2) T1, where tissues are distinguishable. In this sense, T1 gives complementary and necessary information for the segmentation task as hyperintesities in the FLAIR image may correspond also to artifacts in the GM-CSF interface, the skull or infarcted tissue. A first preprocessing step that can remove a lot of noise by focusing the analysis on the area of interest is to remove the skull using a mask. FSL-BET [8] was used to obtain the brain mask in T1, and it was applied to both input volumes afterwards.

Normalization. FLAIR and T1 images have different scales and intensity levels and are subject and machine dependent, so a normalization process is very necessary for the correct functioning of deep learning algorithms. Due to the high intersubject intensity variance, three different types of normalization were applied on a per-case basis instead of applying it directly to the entire dataset.

Generally, not normalizing images at all leads to the network not being able to converge properly, or to very unstable results. A very simple way to normalize data is to linearly move all intensities to range [0, 1] with the min-max normalization.

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

As both volume inputs, FLAIR and T1, present a leptokurtic distribution with extreme outliers, a min-max normalization can squeeze the data in very low ranges as shown in Fig. 1. In order to spread the range of intensities as much as possible to allow the network to differentiate hypointense pixels from hyperintense ones we can use quantile normalization.

$$x' = \begin{cases} 0 & if\ x < P_{0.5} \\ 1 & if\ x > P_{99.7} \\ else & \frac{x - P_{0.5}}{P_{99.7} - P_{0.5}} \end{cases} \tag{2}$$

where P_n is the percentile n. It is a non-linear transformation that cuts the ends of the distribution while preserving linearity in the central region (an example is shown in Fig. 1).

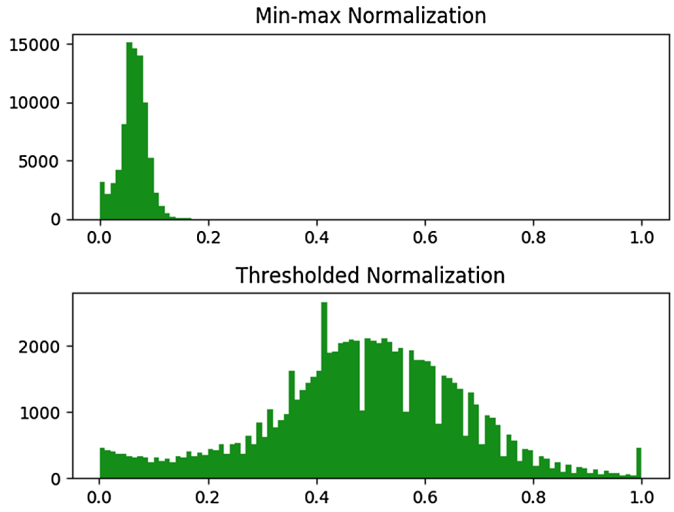


Fig. 1. FLAIR intensity distribution before and after applying quantile normalization.

Standardization. Standardization is one of the most common feature scaling techniques in machine learning. It is also widely used in fully convolutional networks as well as in similar segmentation problems. In this case, the linear transformation rescales the distribution to have zero mean and a standard deviation of one.

$$x' = \frac{x - mean(x)}{\sigma}$$

Contrast Enhancement. There exists a large inter- and intra- observer variability in the manual delineation of WMH. This makes the gold standard used for training not very precise and therefore, to obtain precise results by automatic segmentation becomes difficult [1].

In MRI brain images, several problems can lead to erroneous segmentation or pixel classification errors [9]. These problems can be: average partial volume, noise overlap and intensity of adjacent tissue classes and for WMH in FLAIR images a lower contrast may appear at the edge while in the center of the region a higher contrast [10]. To avoid these problems, a better separation between the pathology and the background of the image can be achieved using contrast enhancement techniques. A review on contrast enhancement techniques, not only subscribed to MRI, can be found in [11], for a review in MRI field see [10].

In our work we use the technique developed in [9] to improve the contrast of WMH in FLAIR images. This technique uses an estimate of the WMH edge magnitude and the intensity values combined through several transformations to highlight WMH, see Fig. 2. All the transformations are only calculated from the characteristics of each slice of the image, so they are adaptive. The technique achieves an average contrast improvement of 41.1% in the experiments performed in the original work.

Fig. 2. Contrast enhancement: original FLAIR (left), edge map (center) and enhanced FLAIR (right).

2.3 Fully Convolutional Neural Network (adapted U-Net)

In this work we propose a deep learning approach using a FCNN that follows the U-Net architecture [4]. Due to the non-isotropic voxel size of the FLAIR volumes, we applied a two dimensional approach, analising the volume slice by slice from the axial view. The FCNN was feeded with two input channels, one corresponding to FLAIR information and the other to the T1.

In order to obtain good results, deep learning techniques usually apply data augmentation when the dataset in small batches, which allows the model to converge and generalize better. In our case, to triple the dataset, we applied affine data transformations, such as rotations on $[-30°, 30°]$ angles, shifts on both the x and y axis $[-30\%, 30\%]$ of the total width and height, respectively, zoom on both axes in the ranges $[0.9, 1.2]$ and shears in the range $[-0.2, 0.2]$. Actual values where picked randomly from a normal distribution.

The U-Net is a fully convolutional neural network. It has a contracting path on the left side and a expanding path on the right, giving it a U shape. The contracting follows a more conventional structure of two convolutions followed by a pooling layer, this process repeats four times. As the expanding path upsamples the feature maps it is concatenated with the respective level of the contracting path, then two convolutions are applied. This process is also repeated four times.

The U-Net was originally designed for multi class classification and a softmax activation layer was used in the last layer. However, here we are using a sigmoid as the final activation function due to the binary nature of the segmentation problem. On the other hand, the weighted cross entropy loss function proposed for the U-Net was not the proper choice for this problem due to how highly unbalanced our data is (only 0.17175% were WMH voxels). Instead the Dice Similarity Coefficient was used as the loss function for training the network. This metric is widely used as loss function for similar binary segmentations [12].

The loss function used was the negative Dice Similarity Coefficient:

$$DSC_{loss} = -2 \times \frac{\sum_{n=1}^{N} |p_n \circ g_n| + s}{\sum_{n=1}^{N} |p_n + g_n| + s}$$

where $\circ$ is the element-wise product of two matrices (also represented as intersection since we are using binary matrices), $|x|$ is the sum of values of matrix x, p_n and g_n stand for predicted segmentation and ground truth, respectively, and s stands for smoothing and assures that there will be no division by 0. Generally s is set to 1.0, however it can have a big influence in the average when the number of non-zero pixels is low (which happens often in these datasets). So we lowered s to 0.01.

The learning rate was set to 0.000001 to guarantee convergence. Related work [13] set a higher learning rate but they also conclude that with high learning rates the U-Net gets stuck at a low level of DSC loss and is not able to converge in multiple trainings. For a learning rate of 0.01, the DSC value will not surpass 0.01.

All convolution kernel sizes were changed to 5×5 instead of the original 3×3, to capture richer local data. All maxpoolings are kept as 2×2 with a stride of 2×2.

The original study for the U-Net proposes initializing weights from a truncated Gaussian distribution centered on 0 with a standard deviation of:

$$stdv = \sqrt{\frac{2}{N}} \tag{3}$$

where N is the number of inputs from the previous layer. We used this initialization, which granted more stability across trainings than the Glorot uniform, also named Xavier uniform [14], which is set as default in the Keras framework.

Since the batch size and learning rate affect the gradient, we set batch size to 30 to guarantee convergence given the selected learning rate.

The number of epochs was cut to 35 to prevent the model from over-fitting. We used 81.6% of the combined datasets for training. Another 15% of the data for testing during training and evaluating our method. The 3.4% left is used for validating predictions visually and generating output images.

The network was trained on Amazon Web Services (AWS). Out of their GPU portfolio we chose the p2.xlarge EC2 instance, which suited our needs for this task. The average training time was 10 h, however this time was lower on the trainings with only one channel. We used the Keras implementation of the Adam optimizer [15] for stochastic gradient based optimization.

3 Experimental results

Table 1 shows the results, ordered by DICE, after training for the different preprocessing configurations. It can be observed that the best data normalization is given by standardization. Only one of the configurations with standardization is not found in the first ranking positions, probably due to an error during conduction of the experiment.

Secondly, the use of a brain mask and the contrast enhancement technique have a significant influence in the results.

Finally, since normalization serves to put all variables on the same scale and thus facilitate that all entries have the same influence on the solution, and that, in our case, roughly, we could say that the FLAIR image provides Information about the WMH and the T1 image on the type of tissue, it might be interesting to weight the FLAIR image more than the T1 image. To evaluate this hypothesis, three experiments with different influence of T1 were performed: (1) T1 with the same weight as FLAIR (T1 = yes), (2) T1 weighted to 0.1 w.r.t. FLAIR (weighted 10%) and (3) eliminating T1 of the input data (T1 = No). The results obtained (first three configurations of the Table 1 do not support this hypothesis, since the difference between them is not significant. It is necessary to carry out more experiments to analyze if the T1 image has any significant influence in the segmentation, since the brain mask could also be obtained from the FLAIR image.

Figure 3 shows the results obtained with the FCNN in three slices with different load of WMH. It can be observed that detection of WMH is quite consistent except for the smaller ones.

4 Discussion

Input data normalization by standardization is a determining factor in the improvement of results. This may be motivated because the distribution of intensities in each case (volume, subject) is dominated by the number of tissue voxels of WM and GM, independently of the WMH load, which approximates a normal distribution. By standardizing the intensities of voxels within each case, we are normalizing with respect to the mean and the variance and, therefore, improving the correlation of the different tissues between cases.

Table 1. Segmentation performance of the FCNN trained on the WMH challenge input data with different preprocessing. tables.

Mask	Flair	T1	Normalization	DSC
Brain	Enhanced	Yes	Standardization	81.30
Brain	Enhanced	Weighted 10%	Standardization	81.05
Brain	Enhanced	No	Standardization	81.02
No	Enhanced	Yes	Standardization	79.49
Brain	Original	Yes	Standardization	78.29
No	Original	Yes	Standardization	76.81
No	Enhanced	Yes	Min max	76.72
Brain	Enhanced	Yes	Min max	76.34
Brain	Enhanced	Yes	Quantile	74.89
No	Enhanced	Yes	Quantile	73.04
Brain	Original	Yes	Quantile	72.70
Brain	Original	Yes	Min max	69.72
No	Original	Yes	Quantile	69.12
No	Original	Yes	Min max	66.89
Brain	Original	Weighted 10%	Standardization	64.04
Brain	Original	Yes	Min max	51.80

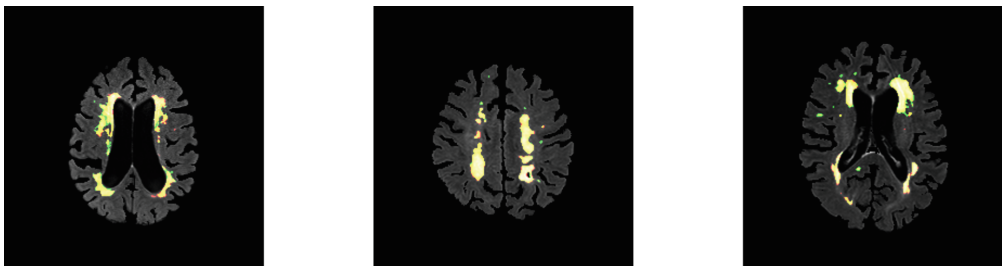


Fig. 3. Segmentation results on three slices from different subjects. True positive voxeles are shown in white or yellow colors, false positive voxels in red color and false negative voxels in green color. (Color figure online)

With the results obtained from the first three configurations of Table 1 we could conclude that the information introduced by T1 for the segmentation of the WMH is scarce, but it will be necessary to carry out more experiments to assess if the improvement is statistically significant.

5 Conclusions

In this work we analyze the use of different preprocessing techniques to improve automatic WMH segmentation based on multicontrast MRI analysis with

FCNNs. The biggest improvement is obtained by (1) using per-case standardization to normalize the data because it improves tissue intensity correlation between cases and (2) focusing the analysis in the brain removing the skull, and (3) applying a non-linear transformation that enhances WMH contrast.

The tests with FCNNs are very costly temporally and computationally (10 h on average per training), but it is demonstrated that the results obtained are competitive with the methods found in the state of the art.

In this paper we use a non-linear transformation that increases the contrast in WMH, which improves the performance of convolutional networks and at the same time poses a way of reducing the inter- and intra-observer variability. The use of more precise references would allow the system to also increase its precision.

Acknowledgements. This work is partially supported by the Autonomous Community of Madrid (PEJD-2018-PRE/TIC-8977).

References

1. Rincón, M., et al.: Improved automatic segmentation of white matter hyperintensities in MRI based on multilevel lesion features. UNED, Madrid, Spain (2017)
2. Caligiuri, M.E., Perrotta, P., Augimeri, A.: Automatic detection of white matter hyperintensities in healthy aging and pathology using magnetic resonance imaging: a review. Neuroinformatics **13**, 261 (2015). https://doi.org/10.1007/s12021-015-9260-y
3. WMH Segmentation Challenge. https://wmh.isi.uu.nl/
4. Ronneberg, O., Fischer, P., Brox, T.: U-Net: convolutional netowrks for biomedical image segmentation. University of Freiburg, Germany (2015)
5. Klein, S., Staring, M., Murphy, K., Viergever, M.A., Pluim, J.P.W.: elastix: a toolbox for intensity based medical image registration. IEEE Trans. Med. Imaging **29**(1), 196–205 (2010)
6. Shamonin, D.P., Bron, E.E., Lelieveldt, B.P.F., Smits, M., Klein, S., Staring, M.: Fast parallel image registration on CPU and GPU for diagnostic classification of Alzheimer's disease. Front. Neuroinformatics **7**(50), 1–15 (2014)
7. Ashburner, J., Barnes, G., Chen, C., Daunizeau, J., Flandin, G.: SPM12. Wellcome Trust Centre for Neuroimaging (2017)
8. Smith, S.M., et al.: Advances in functional and structural MR image analysis and implementation as FSL. NeuroImage **23**(S1), 208–19 (2004)
9. Khademi, A., Venetsanopoulos, A., Moody, A.: Automatic contrast enhancement of white matter lesions in FLAIR MRI. In: 2009 IEEE International Symposium on Biomedical Imaging, From Nano to Macro, pp. 322–325 (2009)
10. Isa, I., Sulaiman, S.N., Abdullah, M.F., Tahir, N.M., Mustapha, M., Karim, N.K.A.: New image enhancement technique for WMH segmentation of MRI FLAIR image. In: 2016 IEEE Symposium on Computer Applications Industrial Electronics (ISCAIE), pp. 30–34, May 2016
11. Huang, S.-C., Yeh, C.-H.: Image contrast enhancement for pre-serving mean brightness without losing image features. Eng. Appl. Artif. Intell. **26**(5), 1487–1492 (2013)

12. Milletari, F., Navab, N., Ahmadi, S.-A.: V-Net: fully convolutional neural networks for volumetric medical image segmentation (2016)
13. Li, H., et al.: Fully convolutional network ensembles for white matter hyperintensities segmentation in MR images (2018)
14. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. Universite de Montréal, Québec, Canada, DIRO (2010)
15. Kingma, D.P., Lei Ba, J.: ADAM, A method for stochastic optimization (2015)