Medical Image Segmentation
3D MRI Brain Image Segmentation using 2D U-Net

Cylia Ouadah[1], Jana Vujadinovic[1]
[1]University of Girona, Spain

***Index Terms*—MRI images, Brain Segmentation, 2D U-Net**

I. Introduction

IMAGE segmentation presents essential part of computer vision and as a result of a segmentation an image is partitioned in distinctive parts which enable localization of specific objects. Furthermore, each part is labeled with a specific value which is unambiguous among parts.

Segmentation of the MRI data into specific tissue types and identification of specific anatomical structures is a useful diagnostic tool. It enables visualizing anatomical features, assessing brain changes, defining diseased regions etc. [1]

At its most fundamental level, the anatomy of the brain is divided into three separate classes: white matter (WM), gray matter (GM), and cerebrospinal fluid (CSF). The IBSR18 dataset was used to segment the three classes indicated above. The criteria for evaluation were Dice score (DSC), Hausdorff distance (HD) and average volumetric difference (AVD).

II. Materials and Methods

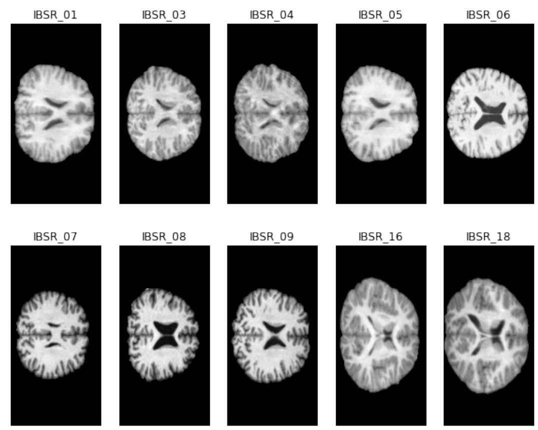
A. Data

The project consisted in using the IBSR18 dataset. The dataset was set up so that 10 images were used for training, 5 images for validation, and 3 images were used for testing. The images used for training and validation included their respective ground truth data for labels, but no ground truth data for test data was provided. Another factor to consider was that the voxel spacing varied between cases. Size of the volume equaled between cases and is 256 x 128 x 256.

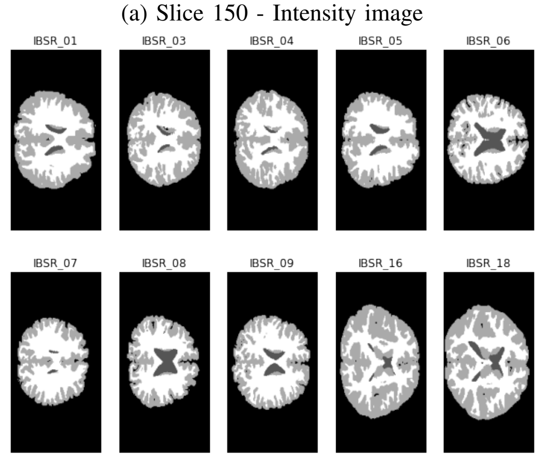
Name	X	Y	Z
IBSR_01	0.937500	1.5	0.937500
IBSR_03	0.937500	1.5	0.937500
IBSR_04	0.937500	1.5	0.937500
IBSR_05	0.937500	1.5	0.937500
IBSR_06	0.937500	1.5	0.937500
IBSR_07	1.000000	1.5	1.000000
IBSR_08	1.000000	1.5	1.000000
IBSR_09	1.000000	1.5	1.000000
IBSR_16	0.837054	1.5	0.837054
IBSR_18	0.837054	1.5	0.837054

TABLE I: Voxel spacing for each volume case

As seen in Table I the given dataset contains three unique voxel spacings. Five cases had voxel spacing of 0.9375, 1.5, 0.9375 with respect to the x, y, and z axes, three cases had voxel spacing of 1.0, 1.5, and 1.0, and two cases had voxel spacing of 0.8375, 1.5, 0.8375.



(a) Slice 150 - Intensity image



(b) Slice 150 - Label Image

Fig. 1: Visual representation of axial view of slice 150 with respect to the intensity image (a) and label image (b). In the label image the darkest region belongs to the CSF, slightly lighter to the GM and the white region belongs to the WM.

In terms of both intensity and spatial information, a heterogeneity can be seen in Fig 1a and 1b. In axial view, a separate anatomical portion of the brain is visible in each example for the exact same numerical slice. The intensity difference between volumes with respect to specific classes can be seen in Fig A1. Additionally, the number of pixels corresponding to each class varies.

B. Data Pre-processing

1) CLEHE

As previously said, there is a distinction between images, and images also appear washed out, which is why pre-processing is required prior to using any image segmentation technique. For image pre-processing, CLAHE (Contrast Limited Adaptive Histogram Equalization) was applied.

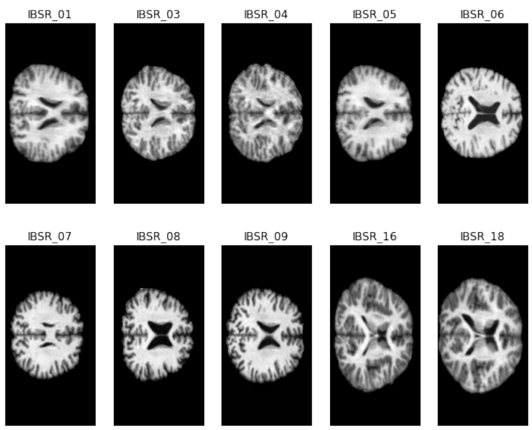


Fig. 2: Visual representation of axial view of slice 150 with respect to the intensity image after applying CLEHE as pre-processing technique

2) Patches

Using any type of deep learning model, in general, necessitates the availability of a large dataset, which is not always the case. Only 10 volumes were available in this situation, and that number was to be lowered due to the training/validation split during the training procedure of the specific DL method. Even when looking at the problem in 2D, the number of possible images in axial view was 2560, despite the fact that the number of slices contains no information. When this is taken into consideration, the number of images accessible for training is reduced to 1301.

Given this, the technique for expanding the quantity of data available for training the DL algorithm includes partitioning the data set into overlapping and non-overlapping patches of a specific size. The exact size of the patch was evaluated by quantitative analysis of the results when comparing the results with respect to the validation set provided.

Patch extraction was done with tensorflow's built-in function, which requires pre-defining patch size and the stride, which determines whether or not we can get overlapping patches. The stride determines when the next patch begins in relation to when the previous one started. As a result, overlapping patches can occur when the stride is smaller than the size of the patch's specified axis. Furthermore, if the number of components in the patch exceeds a pre-determined threshold in terms of non-background elements, the patch is deemed useful; otherwise, it is skipped. The threshold used in the final model was 0.5 (0.3 was also tried), which means that at least half of each patch is not background.

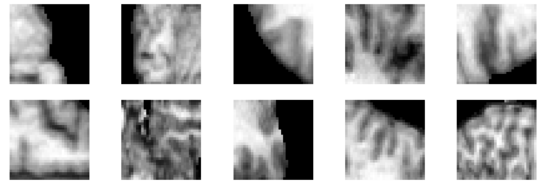


Fig. 3: Sample of patches of size 32x32 used for training the DL network.

C. Data Imbalance

The fact that this dataset had a class imbalance was the third important point to consider. The following frequency of brain classes was present when constructing non-overlapping patches of a specified size. 17.4 % of the all the images is taken up by background, 1.5 % t by CSF, 48.6 % by GM, and 32.5 % by WM. The percentage was calculated by summing the number of elements in the whole dataset belonging to a specific class and dividing by the total number of elements.

With relation to very low amount of CSF present, data repetition was included to address this issue. Every patch with CSF greater than a pre-determined threshold in terms of the amount of pixel elements belonging to the aforementioned class was repeated at a pre-determined rate. Repeated images were stacked on the training dataset and the dataset was shuffled before training.

D. Image Segmentation - U-Net

The architecture used for tissue segmentation is a 2D U-NET optimized to our task. We chose the model to be 2D instead of 3D to tackle the lack of data, since we only have 10 volumes. Using 3D patches with a threshold regarding background didn't result in a big amount of input data. Moreover, according to the literature results of 2D and 3D models in tissue segmentation are very close.

U-NET :

Downsampling (encoder part) and upsampling (decoder part) are two symmetric aspects of the U-Net architecture. Encoder section extracts high-order abstract characteristics from images while shrinking their size. The decoder, on the other hand, gradually restores the input image's original matrix size.

The network's downsampling or contracting path consists of repeated double 3x3 convolution and ReLU activation functions followed by max pooling operation. Additionally, when max-pooling is used, the number of features is doubled.

When it comes to upsampling, the process is similar just reversed. The feature map at the given level is upsampled, and the number of features is reduced by two. The corresponding feature map from the downsampling path is concatenated before convolution and ReLU is applied. The final level of the architecture includes using 1x1 convolution.

One more thing that is introduced to the network is dropout. Dropout in down-sampling path is applied before maxpooling, whereas in the up-sampling path it is applied before upsampling.

U-Net optimization will be further discussed in Optimization Process section.

E. Evaluation Metric

For evaluation of results Dice Similarity Metrics (DSC), Hausdorff Distance (HD) and Average Volumetric Difference were used. When doing parameter optimization only DSC was used and for the final model other metrics were calculated as well.

1) Dice Similarity Coefficient

Denoting GT and S as ground truth and segmentation of each slice in the volume of MRI image the following formula was used

$$DSC = \frac{2|GT| \cap |S|}{|GT| + |S|}$$

2) Hausdorff Distance

Hausdorff Distance measures the maximum distance of a set to the nearest point in the other set. [3]

$$d_H(A, B) = \max(d(A, B), d(B, A))$$

3) Volumetric Difference

It measures the difference between the volumes of the segmented image and the ground truth.

$$VD = 100 * \frac{GT.sum() - SEG.sum()}{GT.sum()}$$

F. Implementation

For the data-preprocessing Python 3.7 was used in combination with nibabel 3.2.1 for loading the images and saving, numpy 1.21.2 and skimage 0.18.1. Skimage was used for application of CLEHE and nibabel was used for saving considering the header information with respect to voxel dimensions from the previously loaded nifti files. For the implementation of U-Net keras 2.7.0 was used. Data visualization was done using SnapITK and matplotlib.

III. Optimization Process and Results

Using any deep learning network necessitates the optimization of a number of parameters, as selecting the ideal network hyperparameters can substantially alter the segmentation's final output. The number of convolutional blocks in the encoder and decoder, the number of initial parameters that are downsampled and then upsampled, and the dropout rate were among the parameters that were fine-tuned out of the many that might be optimized. One parameter was altered while the rest were kept constant for determining the optimum parameters. The best value for a parameter was chosen based on the results obtained using the Dice coefficient as an evaluation metric. Furthermore, data-related parameters were optimized in light of the fact that the network results differed depending on the type of patches extracted, as this has an impact on the data network's learning.

A. U-Net Hyperparameter optimization

For each training during the optimization procedure, the model was trained on maximum of 20 epochs using Adam optimizer, early stopping of 10 epochs in case the validation error does not decrease. The Dice was calculated on the validation dataset passed into original axial slices without patches division, in order to optimize the model.

Furthermore, the training data was partitioned so that 30% of it was utilized as validation during training, and the validation data-set provided by the challenge was used to evaluate the network results and determine the optimum parameters. First, CLEHE was applied to 3D volumes, and then 2D patches were extracted for training. Following that, interesting patches were extracted from the 2D slices and used to increase the amount of data for training. Table II shows the initial values for the parameters that were fine-tuned, including network and data-related hyperparameters.

Network Related Hyperparameters	
Number of Encoder Blocks	3
Number of Initial Parametrs	32
DropOut Rate	0.2
Loss Function	Categorical Cross-Entropy
Data Related Parameters	
Patch Size	32x32
Patch Stride	32x32
Replication Rate of CSF	0

TABLE II: Initial parameters used for training the network.

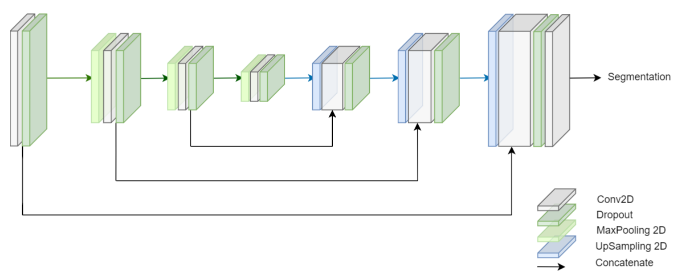


Fig. 4: Architecture of the modified U-Net

Following the aforementioned, the optimization procedure for U-Net hyperparameters can be seen in table III. For each sub-block in Table III the parameter which leads to best results is showcased in bold. The hyperparameter optimization based on best Dice Score for the provided validation data set lead to fine-tuning the U-Net model with following parameters:

- Number of Encoder blocks is set to 3
- Number of Initial Filters is set to 64
- Dropout is set to 0.4

Furthermore, the increase in segmentation of all three classes is apparent and the biggest increase can be seen with respect to the CSF when looking at results in Table III. The optimized U-Net model used in the end can be seen in Fig. 4

B. Data Parameter Optimization

Once the model hyperparameters have been adjusted the next step was to choose best parameters for patch extraction as well as the amount of CSF present when training the model.

# of Encoder Blocks	# of Initial Filters	DropOut	DSC Background	DSC CSF	DSC GM	DSC WM
2	32	0,2	0.996	0.747	0.909	0.889
3			**0.996**	**0.786**	**0.906**	**0.880**
4			0.996	0.770	0.904	0.880
	8		0.985	0.556	0.811	0.856
	16		0.996	0.780	0.902	0.884
	32		0.996	0.786	0.906	0.880
3	**64**		**0.996**	**0.790**	**0.913**	**0.892**
		0	0.996	0.790	0.908	0.890
		0,2	0.996	0.788	0.913	0.892
3	**64**	**0,4**	**0.996**	**0.837**	**0.913**	**0.896**
		0,6	0.996	0.777	0.908	0.893

TABLE III: Hyperparameter optimization of the U-Net model. One parameter was varied and the other was set to be constant until the best performing metric was achieved.

The parameters that were modified with respect to the patches is the patch size and patch stride. From the Fig A3 (see appendix) it can be seen that the patch size was varied and took two values 32x32 and 64x64. Patch overlap was avoided in one example of 32x32 patch size by using a 32x32 patch stride, whereas patch overlap was introduced in the other by using a 16x16 patch stride. Furthermore, the patch stride utilized for the 64x64 patch was 64x64 and 32x32 in order to achieve the same effect. Another thing that can be observed in the table is that when the patch size is increased to 64x64 and overlapping patches are considered, the percentage of CSF increases. However, when using overlapping patches of size 32x32 better results were obtained.
The next step was to introduce oversampling to increase the amount of CSF. This was accomplished by repeatedly applying patches with a thresholded amount of CSF at a set rate to increase the amount of CSF. Fig A3 (see Appendix) only shows the findings for x30 and x50 in our situation.

Although the optimal dropout was initially set to 0.4, a brief experiment was carried out with a dropout of 0.2, and the results obtained were best with the dropout of 0.2 and the oversampling rate of x50 for CSF in combination with the aforementioned parameters as can be seen from Table IV. Furthermore, it can be deduced from Table IV that the data utilized for network learning, as well as the level of presence of a specific class, have a significant impact on network learning.

The final segmentation model hyperparameters and data related parameters can be found in Table IV.

Network Related Hyperparameters	
Number of Encoder Blocks	3
Number of Initial Parametrs	64
DropOut Rate	0.2
Loss Function	Categorical Cross-Entropy
Data Related Parameters	
Patch Size	32x32
Patch Stride	16x16
Replication Rate of CSF	50

TABLE IV: Optimized Parameters for the U-Net model used.

IV. Final model Results and Discussion

The final segmentation model used was the one following TABLE IV parameters. The model was trained for 20 epochs, taking into account the validation loss to monitor the weights saving, the validation loss didn't improve after the 14th epoch, that is why augmenting the number of epochs was not needed. The results of the final segmentation model on the validation dataset were as follow :

1) Dice coefficient : The Dice score obtained : 0.887 for CSF, 0.923 for GM and 0.912 for WM.
2) Volumetric difference: 6.09 for CSF,0.657 for GM and 0.648 for WM.

From the previous results we can see that the model performed good on the 3 classes overall, the dice coefficient is high, the volumetric difference is low which means that the predicted volume and the ground truth volume are similar. However, the results for the CSF are less good compared to GM and WM and this can be explained by the lack of csf samples in the data, since the presence of csf in the labeled volumes is small.

Qualitative results obtained from the final model are visualized by Snap-Itk. Figures 5-9 showcase the segmentation results for different views (axial, sagital and coronal) in the upper line, and the given ground truth (axial, sagital and coronal) in the second line.
All results are for position (x = 135, y=60 , z=150).

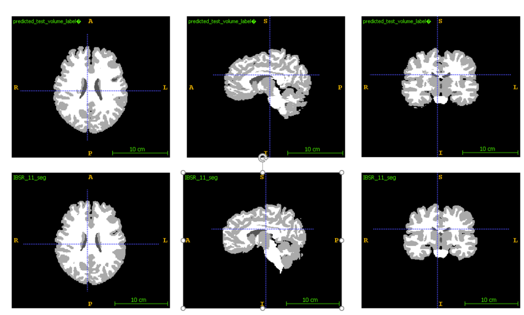


Fig. 5: Results of segmentation and Ground truth for IBSR_11

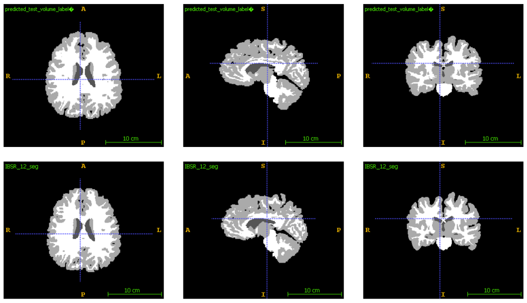


Fig. 6: Results of segmentation and Ground truth for IBSR_12

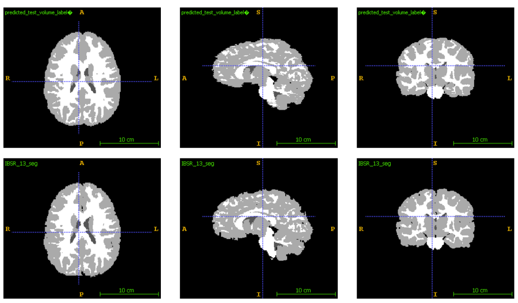


Fig. 7: Results of segmentation and Ground truth for IBSR_13

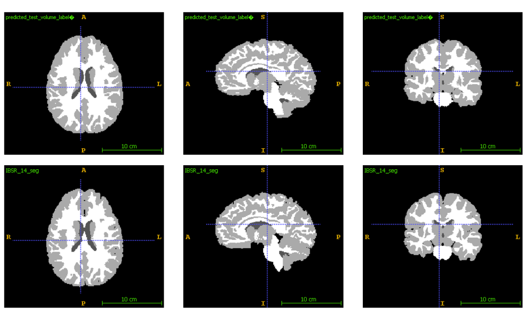


Fig. 8: Results of segmentation and Ground truth for IBSR_14

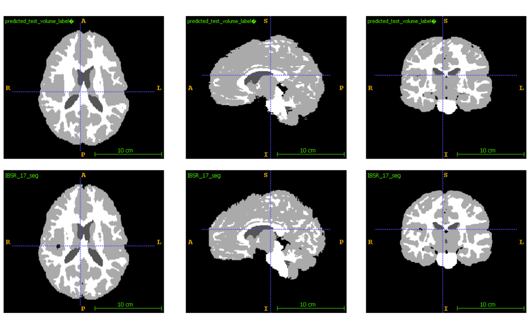


Fig. 9: Results of segmentation and Ground truth for IBSR_17

V. Conclusions

- 2D U-Net is a suitable architecture to use for 3D brain image segmentation and it is less computationally expensive.
- Dividing the data slices into overlapping patches is a good strategy to tackle the problem of small datasets in the task of brain tissue segmentation.
- Unlike harder tasks (tumor localization, lesions segmentation etc), brain tissue segmentation can be performed without the need of spatial information. However, incorporating spatial information (Using probabilistic atlas multiplied by U-Net probabilities outputs for example) would improve the segmentation results and reduce underestimation and/or overestimation of classes.
- Oversampling the minority class helps improve the accuracy of its detection.

VI. Organization and Development of the Coursework

The project required firstly to search about the brain tissue segmentation part and possible architectures.We decided to use a U-Net since it is the most used network for segmentation and according to some recent research papers (including the paper of the nn-Unet), the U architecture is the most suitable for any type of biomedical imaging and it can produce the best results as long as proper data preparation and optimization is done.

The next main step was to study different strategies of MRI image prepossessing. We started by doing image registration of the volumes taking as a reference the volume with the highest resolution. This could have been helpful when using a spatial information based approach, we didn't included in the final trials since our final strategy was to use 2D patches instead of the whole slices. Other types of prepossessing techniques were evaluated such as Histogram equalization, Adaptative Histogram equalization, Histogram matching and intensity normalization. This part took us several days to choose the prepossessing used in the final pipeline.

The optimization part of the architecture, hyperparameters and data parameters was the longest part of the project, several trials were made for each combination. The final model selection was based on the segmentation results of the 5 validation volumes.

References

[1] Despotović I, Goossens B, Philips W. MRI segmentation of the human brain: challenges, methods, and applications. Comput Math Methods Med. 2015;2015:450341

[2] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

[3] http://cgm.cs.mcgill.ca/g̃odfried/teaching/cg-projects/98/normand/main.html.

[4] Fabian Isensee, Jens Petersen, Andre Klein et al. nnU-Net: Self-adapting Framework for U-Net-Based Medical Image Segmentation

Appendix

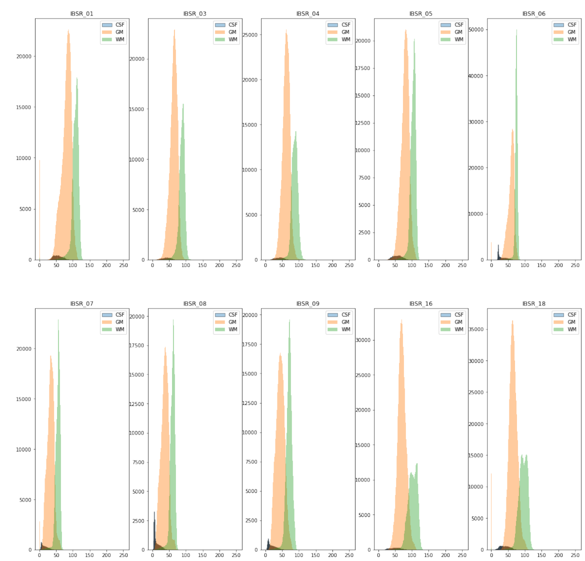


Fig. A1: Histogram of provided volumes in IBSR18 Dataset before pre-processing

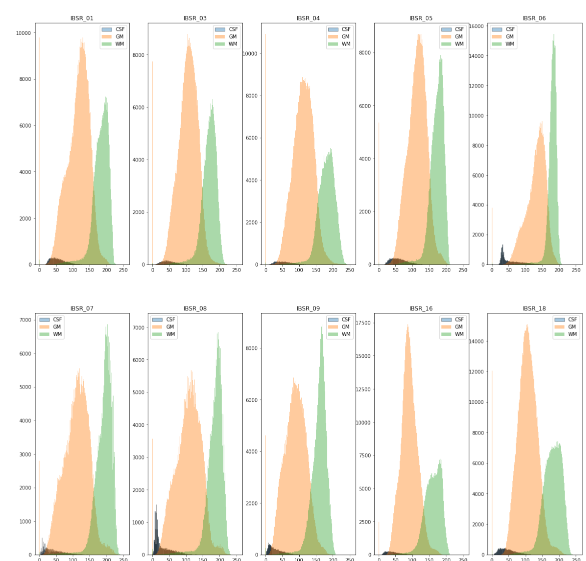


Fig. A2: Histogram of provided volumes in IBSR18 Dataset after applying CLEHE. It is important to note when plotting the histograms all the values were multiplied by 255 to be in the range between 0-255.

DropOut	Patch Size	Patch Stride	Augmented CSF	BG [%]	CSF [%]	GM [%]	WM [%]	Accracy Background	Accracy CSF	Accuracy GM	Accuracy WM
0.4	**32x32**	**16x16**	**0**	**16.1**	**1.4**	**49.1**	**33.3**	**0.996**	**0.874**	**0.919**	**0.908**
0.4	32x32	32x32	0	17.6	1.2	48	33	0.996	0.837	0.913	0.896
0.4	64x64	64x64	0	32.2	1.4	39.9	26.7	0.996	0.722	0.906	0.883
0.4	64x64	32x32	0	26.7	9.2	41.8	29.9	0.996	0.869	0.917	0.902
0.2	32x32	16x16	x30	11.7	9.2	43	36.2	0.996	0.882	0.924	0.911
0.2			**x50**	**9.9**	**12.3**	**40.5**	**37.3**	**0.996**	**0.887**	**0.923**	**0.912**
0.4			x30	11.7	9.2	43	36.2	0.996	0.880	0.924	0.911
0.4			x50	9.9	12.3	40.5	37.3	0.996	0.878	0.922	0.909

Fig. A3: Optimization of parameters with respect to data size and CSF repetition.