A Deep Learning Approach to Parameter Mapping from Perfusion MRI in Ischemic Stroke

Aaron Sossin, Mackenzie Carlson, Matthew Radzihovsky
Stanford University
(Dated: March 19, 2021)

We investigate the generation of 3D parametric scans from time-series 3D magnetic resonance perfusion (MRP) images for ischemic stroke patients. Parametric scans are parameter maps for metrics such as cerebral blood flow and time-to-max (Tmax), which are essential for possible life-saving treatment of stroke patients. In this paper, we specifically generate Tmax mappings which typically rely on expensive RAPID software using a variety of Convolutional Neural Network (CNN) architectures. Constrained by a small dataset, we garnered successful results particularly by a 2-Layered CNN. This work provides a valuable foundation with which to eventually create an open-sourced and free platform for parametric mapping.

I. INTRODUCTION

We have been pursuing a deep-learning and computer vision approach to use magnetic resonance perfusion (MRP) imaging, also know as bolus-tracking MRI or perfusion weighted imaging to predict parametric perfusion maps (including time of maximum value of the residue function, time-to-max or tmax [5]). MRP is dynamically acquired using MRI by injecting a contrast bolus into the patient and repeatedly scanning the same image volume $30-40$ times over 3 minutes, generating a time series of 3D scans, or a 4D set of images. Typically, these scans are fed into complex mathematical models to generate parametric maps to assess the severity of stroke and select the patients to undergo therapy.

However, this mathematical modelling software, RAPID, is very expensive and many hospitals cannot afford it. Therefore, we hope to generate useful parametric maps directly from the MRP without utilizing RAPID, which has the potential to save lives by enabling more hospitals to quickly predict key parametric maps in order to better treat and stratify ischemic stroke patients [1, 2]. An additional benefit of using deep learning for generating parametric maps (in this case, we choose to predict Tmax maps) is to avoid the need for arterial sampling in patients during scanning (which is necessary for the RAPID approach). Tmax is chosen since it has been demonstrated to have the most clinical utility out of the parameters computed by RAPID and is especially useful in determining which stroke patients are good candidates for thrombolysis, a potentially life-saving treatment with many absolute contraindications, where treatment must commence within hours of the stroke.

II. RELATED WORK

To our knowledge, no other groups have attempted to use machine learning to directly generate parametric maps for this application. However, there are related studies in using machine learning for MRP imaging such as using singular value decomposition deconvolution [4] or machine learning based preprocessing [6]. Singular value decomposition deconvolution is a good proof-of-concept work, pointing towards the utility of such methods in a clinical setting, however more complex models will certainly be able to improve results; in our project, for instance, we plan to implement a U-net in order to build off of some of these techniques. ML-based preprocessing could be useful in the future in order to eliminate having to separately preprocess data before input into a model; instead, the raw data could be input directly. However, it doesn't actually accomplish the goal of what we are trying to do, that is generate parametric maps from the processed data.

RAPID is the state-of-the-art software method for automatically analyzing perfusion MR images, specifically in the context of stroke imaging [7]. The resulting parametric maps can be produced within 5 minutes, giving a, "...fast and reliable estimate of salvageable brain tissue to help select patients for endovascular treatment." The major limitation to RAPID is the high cost of the software itself. RapidAI, the platform offering RAPID for perfusion MRI, also offers parametric mapping for stroke using other imaging modalities, including CT, which traditionally has poorer soft-tissue contrast than MRI. The 'AI' part of the platform indicates that deep learning is involved in the software parametric mapping, however groups have only used deep learning to extract other, new information from the parametric maps generated by RAPID [8]

The basis of our work is predicting a 3D image from a 3D image in time series (4D image), an application of dimensionality reductions. One other group at Stanford has tackled a similar type of project, where they used 11 MR image series per subject to predict a PET image illustrating cerebrovascular reserve [9]. In our case, the PET image is analogous to a parametric map and the 11 separate MR image series are analogous to our 4D time-series MR images. They used a two-dimensional encoder-decoder with a U-Net architecture and 6-fold cross validation. While there are differences due to cross-modality image prediction, the approach is still good and highly appropriate to the project. Generally speaking, CNNs

were fundamentally designed to handle imaging data and thus are a great approach to begin. There is growing application of deep learning approaches in medical and neuro imaging [10] and our work builds off of it for MRP imaging which has not been thoroughly explored.

III. DATASET AND FEATURES

Dr. Elizabeth Tong has provided a 45 patient dataset of unlabeled healthy and ischemic stroke subjects with 4D MRI perfusion scans and the associated stack of ground-truth 3D parametric maps generated by RAPID. The perfusion dataset for each subject includes a (128x128x24) image volume acquired at 60 pre-defined time points. In all, the input of one patient is a vector of shape (128x128x24x60). The parametric Tmax map comprises a (128x128x24) image volume per subject. The 24 channels represent the 24 different slice locations. So, each patient's input 4D MRI gets collapsed along the time axis in order to generate 24 (128x128) images as the output. Pixel intensity values range from 0 to 17199. Higher pixel intensities indicate a higher likelihood that this area of the brain is part of either the core or penumbra of the stroke region.

Each individual image is in the DICOM file format. Its metadata contains information about that image such as the patient ID, timestamp, slice location etc... which was used to orient our data.

IV. METHODS

A. Preprocessing

We use Horos software to view and examine the images. We then load the DICOM images into two folders, one for perfusion images (input) and one for Tmax maps (target). Perfusion images were preprocessed by skull stripping, using a threshold-based method set to 10% in MATLAB, then normalized between 0 and 1. This threshold value was chosen by trial and error after visual inspection. A sample of the pre-processing pipeline result is shown in **Figure 1** and the MATLAB scripts used can be found in our github, linked below.

In the event of NaN values in the pixel data, a value of 0 (same as background) was assigned. In the event of missing DICOM files amongst the perfusion images, as long as no one patient had more than 5% of its data missing, substitute images full with zeros were inserted.

B. Model Description

Our strategy is to start simple, and slowly add complexity to our models. Overall, we implement 3 Convolutional Neural Networks using *Keras*. Each one contains

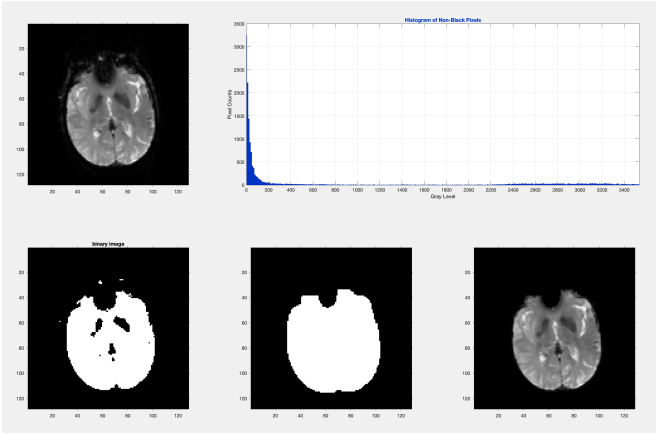

Figure 1. Pre-processing pipeline example. Top left shows original image, top right shows pixel intensity histogram that is thresholded to 10% for skull stripping. Bottom left shows initial thresholded image, middle is image adjusted to fill whole brain volume, and bottom right shows skull-stripped image normalized to 0 to 1 pixel intensity.

Conv3D layers as we are convolving 4D patient data into 3D parametric map predictions.

The simplest model (**Figure 2**) consists of a single filter within a Conv3D Layer of size (1x1x1). The aim of this implementation was to assess what the most basic architecture could achieve. Importantly, filters of size (1x1x1) were used so as to only collapse the channel (in our case time) dimension, and not the size of the resulting images or number of slices. Next, we implement a 2-Layered model (**Figure 3**), also containing solely (1x1x1) filters that has additional *keras* defined BatchNormalization layers. Batch normalization is highly recommended in CNN architectures and helps regularize (to a small degree) and speed up convergence. This model still has a low complexity compared to the literature, and contains 1033 parameters.

Finally, a modified U-net architecture (**Figure 4**) is implemented as an upper bound for complexity. This model consists of 13,123,329 parameters and was inspired by a model designed for brain segmentation located in an open-source github repo [18]. The large advantage to U-net architectures is that it permits the use of MaxPooling layers to extract abstract features that collapse the image dimensions. The U-net architecture collapses each dimension before then building it back up in order to output a 3D shape. The original U-net was configured for 4D input to 4D output, and so we removed a max pooling layer between the 4th and 5th convolutional layers, and changed the filter sizes of the 5th layer to (1x1x1) in order to fit our 4D to 3D specifications.

C. Training Strategy

Due to time constraints and surpassing our allotted Amazon Web Services (AWS) credit allowance, our

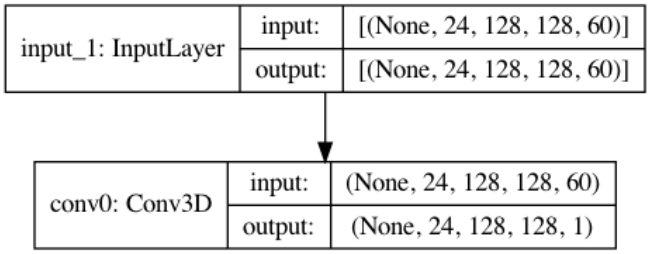

Figure 2. 1-Layered 3D Model Architecture

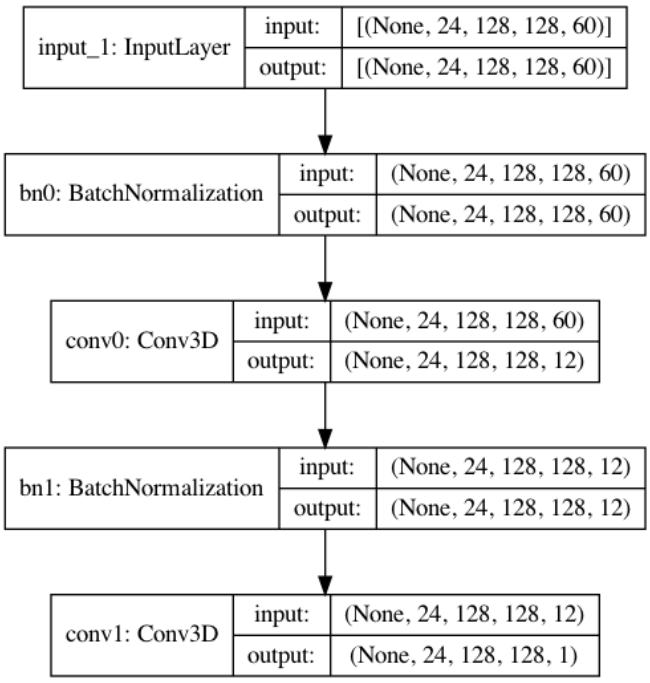

Figure 3. 2-Layered 3D Model Architecture

hyper-parameter tuning remained limited. We chose the Adam optimizer as it is popularly used, and a batch size of 1 in order to increase generalization ability and reduce the computational load of our algorithm. All models were run within an EC2 Instance of type p2.xlarge attached with a Volume of 16000(GiB) in AWS. Our models were leveraged using *CUDA* in conjunction with a *tensorflow* backend. Preprocessed images are passed into our pipeline which automatically performs the hyper-parameter tests and outputs a Results folder with the scores of each run and prediction MRIs on the holdout patient.

Models are all trained with a mean squared error (MSE) loss function, the same as our evaluation metric. Put simply, each predicted pixel value is compared to its expected value and this difference squared is averaged across all 24 brain slices in the output to evaluate performance.

Of our 45 patients, 44 are chosen for the train/dev set and a single handpicked patient (with a nice Tmax map)

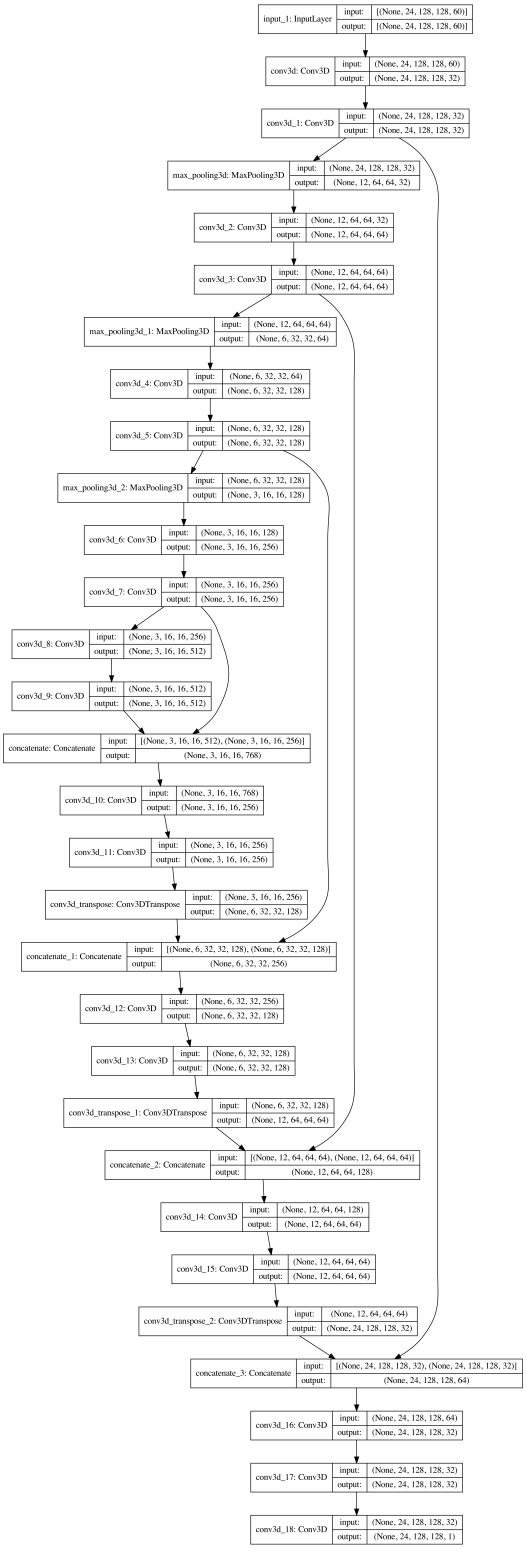

Figure 4. Modified U-net Architecture

is treated as the holdout test patient. The 44 train/dev samples undergo 6-fold cross validation with the above hyper-parameters to assess model performance. Then, each variation is trained on the entire 44 train/dev set and tested on the holdout patient to visually inspect results of each model against a consistent input.

V. RESULTS AND DISCUSSION

Table 1 shows the results of each model architecture after 6-fold cross validation and with varying learning rates at 25 epochs. The 2-Layered Network with a learning rate of 1e-3 performed the best (MSE = 203112), although its MSE was comparable to the U-net architecture (MSE = 214303). **Figures 5, 6, and 7** show the RAPID-generated ground truth map, the most successful U-net result and the most successful 2-Layered Model result respectively. Each visualization is performed after training on the entire dataset except the holdout patient, and then predicting the maps for said patient.

The results suggest that we are far from testing the optimal model architecture. It seems as though both the 1-Layered and 2-Layered models did not yet converge, and are underfitting the data. This is shown in part by the fact that bigger learning rates outperformed smaller ones and the 2-Layer model outperformed the 1-Layer model. Thus, we predict that for each of these models, a higher number of epochs would increase results, as well as increased complexity generated from more layers. The 1-Layered model performed drastically worse than the others.

MSE and visual inspection both confirm that the 2-Layered model performed the best. Although visual inspection reveals that it has a low precision (since many voxels were erroneously attributed a high Tmax value), it is clear that the most important regions isolated by RAPID similarly stand out in our prediction. This result is highly promising as not only did our model recreate an appropriate brain shape, it has shown the ability to extract features pertinent to Tmax selection.

The U-net has drastically more complexity but with middling results. It succeeded at recreating brain images but failed at distinguishing areas of specific interest. We predict that adding dropout and batch normalization layers to our modified U-net would drastically improve its outcome. As a model of higher complexity, it makes sense that more tuning is necessary.

Both the U-net and 2-Layered model overly predict Tmax importance of most voxels (the U-net being at the extreme). This suggests that incorporating a loss function that penalizes guesses that are "too high" more severely than "too low" might be beneficiary. Moreover, although the U-net and 2-Layered Model have similar MSEs, it is clear that the 2-Layered Model outperformed the U-net. This also points to needing a more sophisticated loss function and evaluation metric.

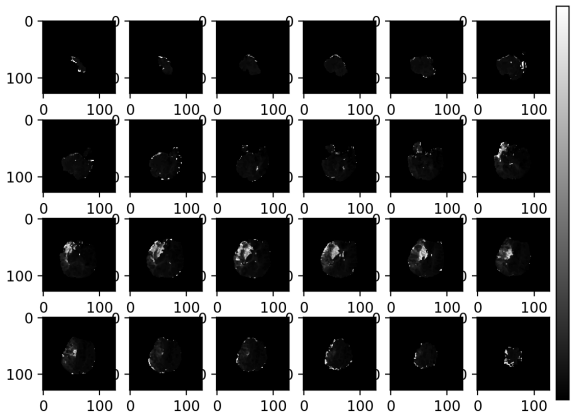

Figure 5. RAPID-generated Tmax Map of Holdout Patient

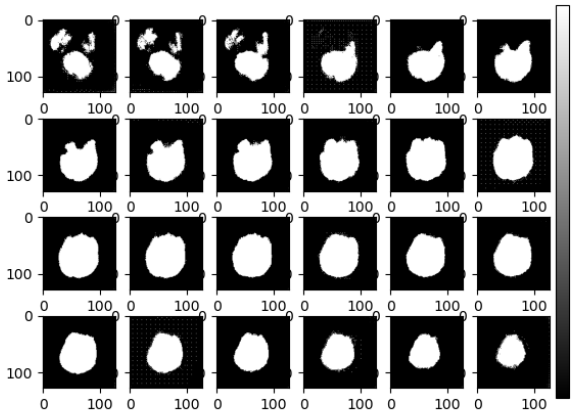

Figure 6. U-net Generated Tmax Map of Holdout Patient

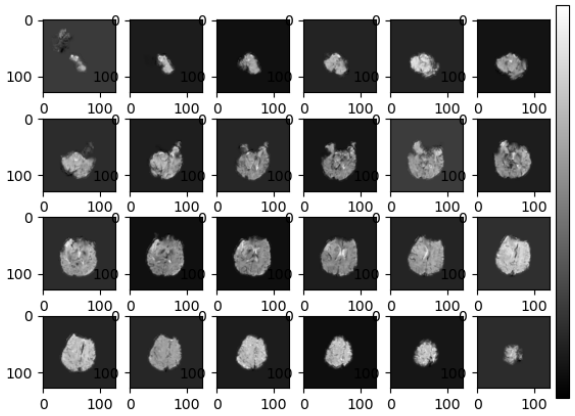

Figure 7. 2-Layer Model Generated Tmax Map of Holdout Patient

Table I. Model Results after 25-epoch 6-Fold Cross Validation Runs

Model	Learning Rate	MSE
1 Layer	1e-5	162699887
1 Layer	1e-4	7046720
1 Layer	1e-3	3664168
2 Layer	1e-5	214097
2 Layer	1e-4	212887
2 Layer	1e-3	203112
U-Net	1e-5	214303
U-Net	1e-4	214270
U-Net	1e-3	214270

VI. CONCLUSION AND FUTURE WORK

CNN architectures have been shown to be an effective strategy towards parametric mapping that warrants further research. Although the results in this paper do not exceed alternative software, they represent a limited data set (45 patients) and relatively minimal testing. A 2-Layered Convolutional Network with Batch Normalization layers and 1033 parameters performed the best with a MSE of 203122. Althouth the U-net has a similar MSE, it is clear from visual inspection that the U-net is simply 'white-ing out' the brain, instead of distinguishing Tmax brain areas. This result highlights the limitations of using MSE as a metric. This work is still preliminary but lays a solid foundation with which to eventually create an open-sourced software that rivals RAPID in efficacy.

The most obvious next step is to increase the data size. There are currently $\tilde{1}00$ patients who are excluded due to incomplete data-sets. We have already begun creating a pipeline for including these patients which will increase our data-set 3-fold.

Next, as mentioned, the hyper-parameter and architectural tuning was quite limited due to AWS credit constraints and time restrictions. Seeing as the 2-Layered model performed the best, we suggest testing a host of different architectures (between 2 and 12 layers) that include further regularization techniques like dropout layers on top of the batch normalization layers. Considering the results we have already garnered, it is likely that with more tuning, the success of our models will increase dramatically. It is also clear that the U-net architecture could benefit from further regularization.

Tmax is but one parametric map that helps clinicians treat ischemic stroke patients. We suggest that the same pipeline be applied to other maps such as CBV (Cerebral Blood Volume), and TTP (Time to Peak) to provide a more complete analysis of the brain and stroke assessment.

The CNNs implemented in this report are a naive approach, that don't include domain knowledge about strokes, and perfusion analyses. Although we are confident that further tuning will improve the CNN models, we also believe that combining CNNs with standard mathematical modelling approaches has the best chance at exceeding industry standards. A prime example is to use mathematical modelling to isolate predictive brain regions such as primary arteries and veins in the brain - and then input these features into our models.

Last, although MSE has been a successful first pass, the similarity between the U-net and 2-Layered models highlight its limitations. We propose the use of clinicians to test new metrics such as the dice coefficient or structural similarity index measure (SSIM) to improve performance. Namely, our present results suggest that precision needs to be emphasized in our loss functions.

VII. GITHUB

https://github.com/aaronsossin/CS230_Project

VIII. CONTRIBUTIONS

Thank you to Dr. Elizabeth Tong for providing the MRP dataset and expanations on parametric scans and RAPID software works.

Aaron Sossin: Writing Codebase, Build/train model, literature searching, writing and editing.

Mackenzie Carlson: Data preprocessing, literature searching, writing and editing.

Matthew Radzihovsky: Build/train model, literature searching, writing and editing.

REFERENCES

[1] Essig, M., Shiroishi, M.S., Nguyen, T.B., Saake, M., Provenzale, J.M., Enterline, D., et al. P*Perfusion MRI: The five most frequently asked technical questions.* 2013;200:24–34. doi: 10.2214/AJR.12.9543.

[2] Copen, W.A., Schaefer, P.W, Wu, O. *MR Perfusion Imaging in Acute Ischemic Stroke.* 2012;21(2):259–283. doi:10.1016/j.nic.2011.02.007.

[3] Ronneberger, O., Fischer, P., Brox, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation.* 2015; doi:1505.04597.

[4] McKinley, R., Hung, F., Wiest, R. Liebeskind, D. and Scalzo, F. *A Machine Learning Approach to Perfusion Imaging With Dynamic Susceptibility Contrast MR.* 2018. doi:10.3389/fneur.2018.00717

[5] Calamante, F., Christensen, S., Desmond, P., Østergaard, L., Davis, S., Connelly, A. *The Physiological Significance of the Time-to-Maximum (Tmax) Parameter in Perfusion MRI.* Stroke. 2010;41:1169–1174. doi: 10.1161/STROKEAHA.110.580670.

[6] Scannell C., et. *Deep-Learning-Based Preprocessing for Quantitative Myocardial Perfusion MRI.* 2020 Jun;51(6):1689-1696. doi: 10.1002/jmri.26983.

[7] Vyas, D., Bohra, V., Karan, V., and Huded V. *Rapid Processing of Perfusion and Diffusion for Ischemic Strokes in*

the Extended Time Window: An Indian Experience. 2019. doi: 10.4103/aian.AIAN_142_18.

[8] *RapidAI Awarded for AI-Powered Stroke Imaging, Diagnosis Technologies*. 2021

[9] Chen, D., Ishii, Y., Fan, A., Guo, J., Zhao, M., Steinberg, G., and Zaharchuk, G. *Predicting PET Cerebrovascular Reserve with Deep Learning by Using Baseline MRI: A Pilot Investigation of a Drug-Free Brain Stress Test*. Radiology. 2020, https://doi.org/10.1148/radiol.2020192793.

[10] Zaharchuk, G. *Next generation research applications for hybrid PET/MR and PET/CT imaging using deep learning*. National Library of Medicine. 2019 Dec;46(13):2700-2707. doi: 10.1007/s00259-019-04374-9.

[11] Goodfellow, I.J., Abadie, J.P., Mirza, M., Xu, B., Farley, D.W., Ozair, S., Courville, A., Bengio, Y. *Generative Adversarial Networks*. 2014:arXiv:1406.2661.

[12] Radford, A., Metz, L., Chintala, S. *Unsupervised Representation Learning with Deep Convolutionals Generative Adversarial Networks*. 2016: arXiv:1511.06434.

[13] Wu, J., Zhang, C., Xue, T., Freeman, W.T., Tenenbaum, J.B. *Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling*. 2017: arXiv:1610.07584.

[14] Karras, T., Aila, T., Laine, S., Lehtinen, J. *Progressive Growing of GANs for Improved Quality, Stability, and Variation*. 2017: arXiv:1710.10196.

[15] Brock, A., Donahue, J., Simonyan, K. *Large Scale GAN Training for High Fidelity Natural Image Synthesis*. 2018: arXiv:1809.11096.

[16] Isola, P. Zhu, J.Y., Zhou, T., Efros, A.A. *Image-to-Image Translation with Conditional Adversarial Networks*. 2016: arXiv:1611.07004:

[17] Zhu, J.Y., Park, T., Isola, P., Efros, A.A. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. 2017: arXiv:1703.10593:

[18] https://github.com/mrkolarik/3D-brain-segmentation/blob/master/3D-unet.py