Computational Imaging Lab

MoDL vs VarNet vs SSDU

Dhruv Sharma dhruv.sharma@fau.de
MSc. Data Science
Friedrich-Alexander-Universität
Erlangen-Nürnberg, Germany

Supervisor: Dr. Zhengguo Tan
zhengguo.tan@fau.de

Abstract

This paper describes the use of different deep learning architectures (MoDL, VarNet and SSDU) for the reconstruction of accelerated Magnetic Resonance Imaging (MRI) data. Data acquisition in MRI is slow and requires different accelerated techniques to be practically feasible. Model-Based Deep Learning (MoDL) combines model-based reconstruction schemes with Deep Learning (DL) and likewise, the Variational Network (VarNet) embeds a generalized Compressed Sensing (CS) concept within a DL framework. These two techniques require fully sampled data for training whereas Self-Supervision via Data Undersampling (SSDU) is an approach that can be trained without fully sampled reference data. We train the three models on the fastMRI brain dataset, perform experiments and discuss which architecture performs well in different settings as well as the pros and cons.

Keywords: MRI, Deep Learning, Compressed Sensing, MoDL, VarNet, SSDU, fastMRI

1 Introduction

MRI data acquisitions tend to be slow since it collects multi-dimensional k-space data thus, limiting patient throughput. Techniques like parallel imaging and compressed sensing have been the standard to make MRIs faster and increase efficiency in the healthcare system. This is done by collecting the data at sub-Nyquist rates and then reconstructing the image from the undersampled k-space. However, at high acceleration rates, the conventional approaches to image reconstruction suffers from noise amplification and residual artefacts. Several ideas inspired by deep learning techniques for computer vision and image processing have been successfully applied to accelerated MRI. MoDL, VarNet and SSDU are such neural network-based reconstruction strategies.

The MoDL scheme involves a data-consistency (DC) term and a Convolutional Neural Network (CNN) based denoiser. The CNN blocks captures information about the image set while the data-consistency blocks encourage consistency with the measurements. This alternating algorithm between the CNN block and the DC block yields a deep network when unrolled. Aggarwal et al. (2019)

Analogous to the MoDL architecture, Varnet also consists of the data consistency and regularization terms that are learned via a sequence of T gradient decent steps.
The original paper uses the Inertial Incremental Proximity Gradient Algorithm (IIPG) whereas in our experimental setup, the ADAM optimizer is used. Moreover, we utilize a list for UNETs without weight sharing to implement learning kernel filters for the regularization term. This is in contrast with MoDL where weights for the CNN are shared in all unroll steps. Hammernik et al. (2017)

SSDU is a learning scheme that does not require fully sampled data for network training. It splits the acquired undersampled k-space indices into two disjoint sets. The DC unit of the network uses one of the set while the other set is used for evaluating the loss function. The input images of the network are obtained from the undersampled k-space data and Coil Sensitivity Maps (CSM). The output image is then converted to k-space to calculate the loss. Yaman et al. (2020)

A key point to note in the three architectures is how the regularization block is implemented. SSDU utilizes a ResNet, MoDL uses a CNN and VarNet operates on the UNET.

2 Method

2.1 Setting up MoDL, VarNet and SSDU

For the task of MRI reconstruction using fully sampled k-space data, we will utilize the following implementation of MoDL and VarNet in Pytorch.

ZhengguoTan (2023) VarNet_MoDL_PyTorch [Source Code]
https://github.com/ZhengguoTan/VarNet_MoDL_PyTorch

For SSDU, the implementation is in the Tensorflow framework. Hence, it is necessary to manage it in a different environment. Alternatively, it is also possible to migrate the Tensorflow code to Pytorch.

byaman14 (2021) SSDU [Source Code]
https://github.com/byaman14/SSDU

After setting up the environment and satisfying the requirements, we can proceed to create the dataset and train the three architectures.

2.2 fastMRI dataset preparation for MoDL and Varnet

These two models require the ground truth image, coil sensitivity maps and the mask as the training input. We will be using the fastMRI brain dataset provided by our supervisor but it can also be downloaded from the following link:

https://fastmri.med.nyu.edu/

fastMRI provides fully sampled k-space data which is used to obtain the ground truth by converting it to the image domain via the inverse Fourier transform. The coil sensitivity maps are then computed using ESPIRiT. For the mask, all results and experiments described in this paper uses a Poisson disk undersampling mask with a 4x acceleration rate. However, different mask choices are also acceptable like Cartesian, radial, spiral, etc. For implementation purposes, the mask is shifted and end slices are removed because they do not contribute much information during training. Our dataset contains slices from all contrasts i.e. FLAIR, T1 and T2.

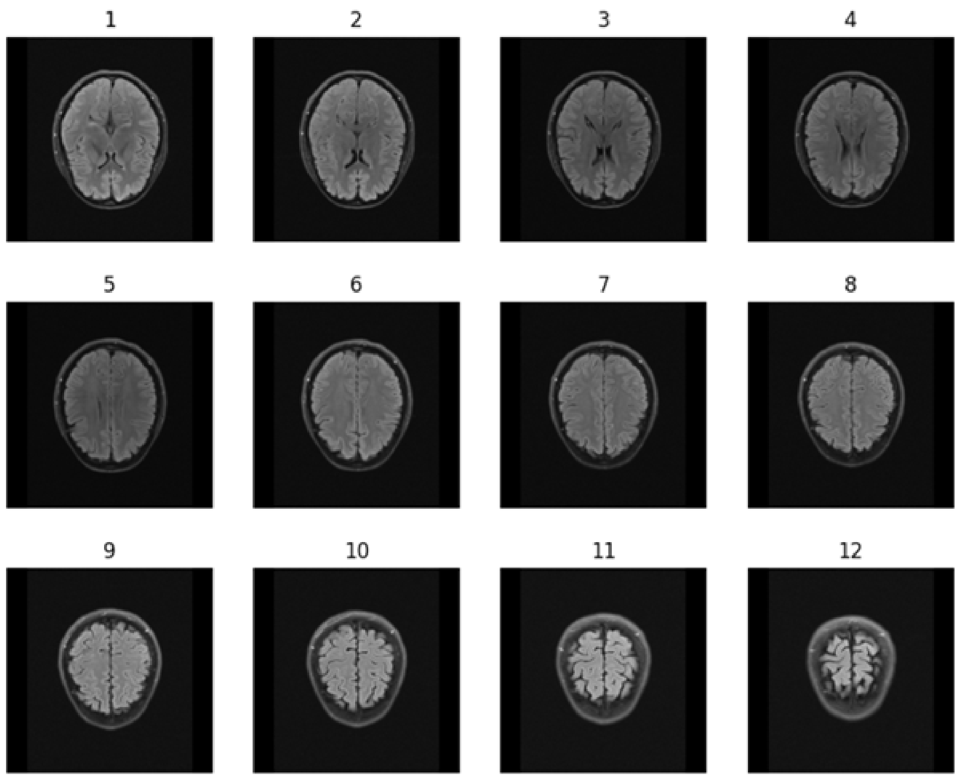

Figure 1: Sample Images, Insignificant End Slices Removed

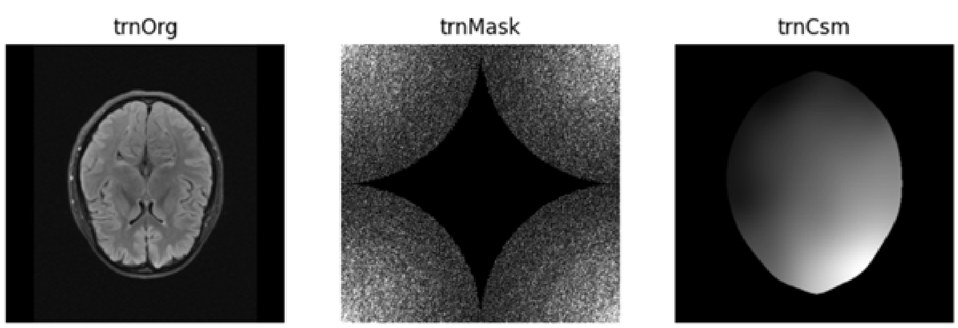

Figure 2: Sample Dataset

2.3 MoDL and VarNet Training

The models are trained with identical parameters for fair comparison.

An overview for the training, testing and inference is given below:

Hyperparameters	Outcome
Learning Rate = 0.001	Train Score = 28.494
Batch Size = 1	Train Loss = 4.624
Number of Epochs = 20	Test PSNR = 27.282
Number of Unrolls = 10	Test SSIM = 0.784

Table 1: Training and Testing Data for MoDL

Hyperparameters	Outcome
Learning Rate = 0.001	Train Score = 38.665
Batch Size = 1	Train Loss = 0.191
Number of Epochs = 20	Test PSNR = 37.844
Number of Unrolls = 10	Test SSIM = 0.936

Table 2: Training and Testing Data for VarNet

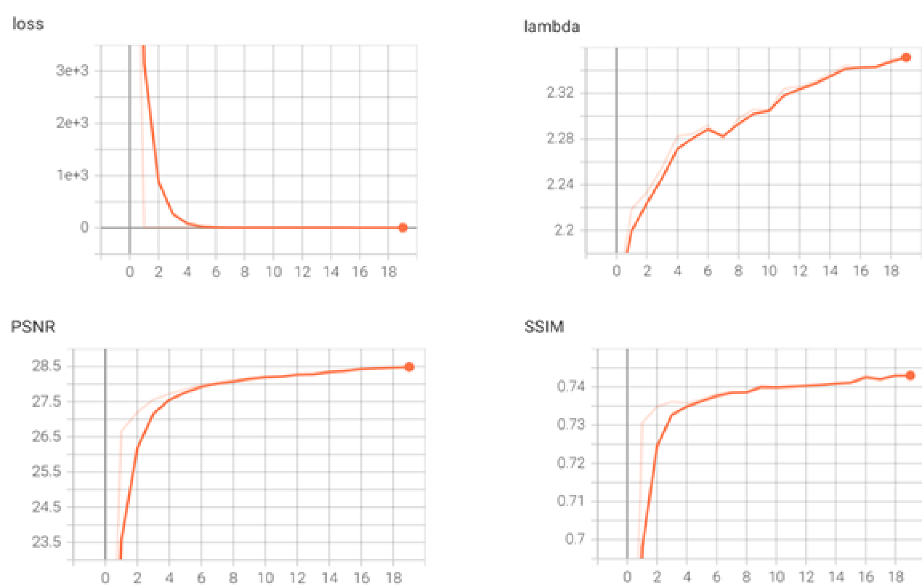

Figure 3: Loss and Score Curves for MoDL

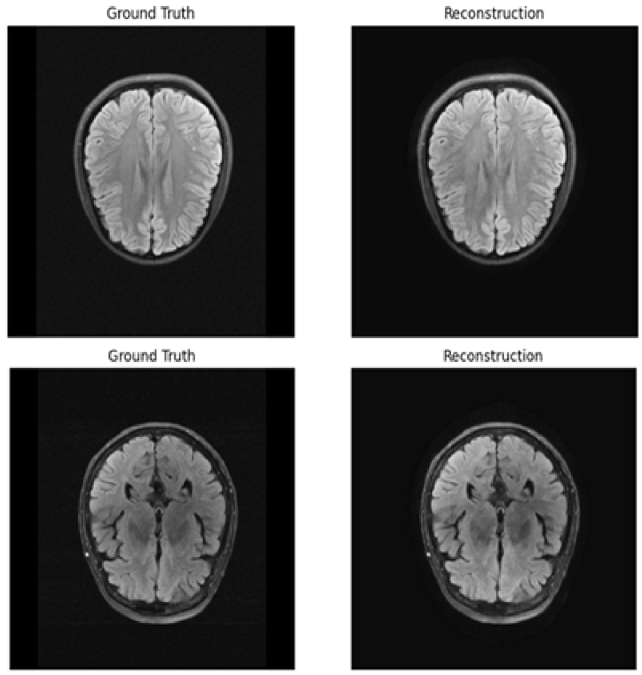

Figure 4: Sample Inference for MoDL

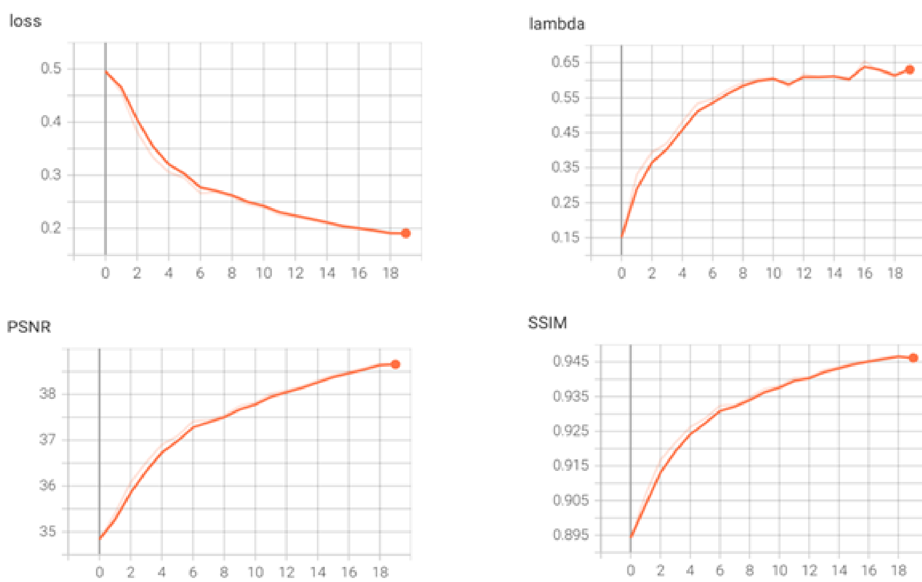

Figure 5: Loss and Score Curves for VarNet

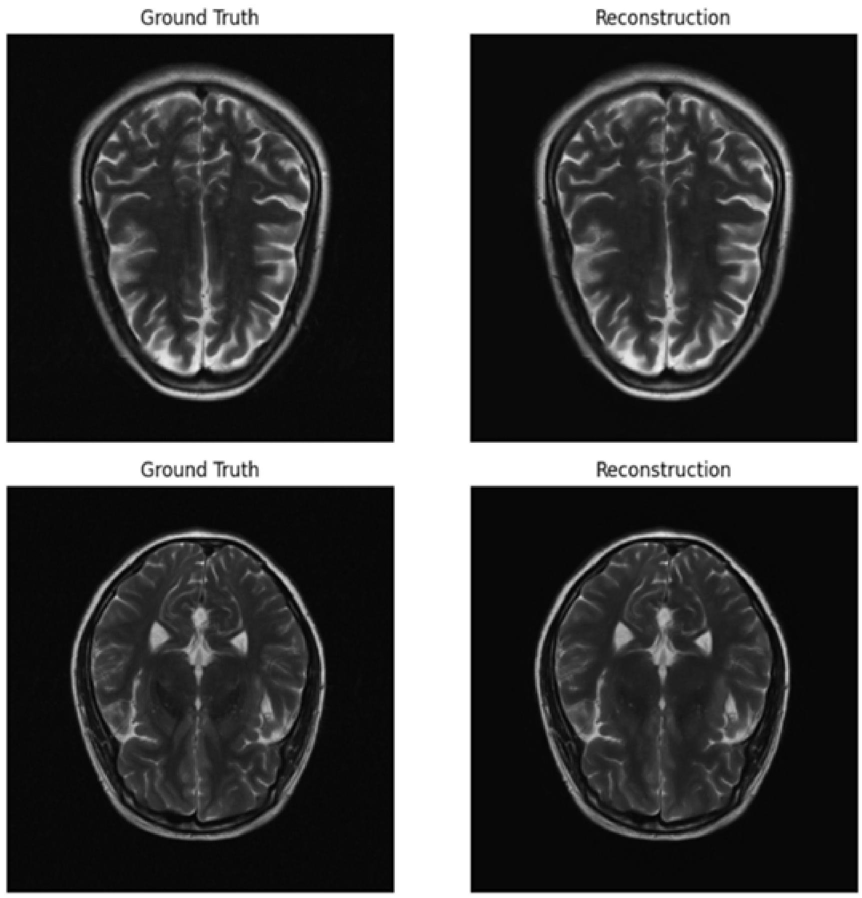

Figure 6: Sample Inference for VarNet

2.4 fastMRI dataset for SSDU

As stated earlier, SSDU works on undersampled k-space data but fastMRI provides fully sampled data. Hence, the k-space is subsampled by pointwise multiplication with a mask. We used the same Poisson disk mask but other masking options are also possible.

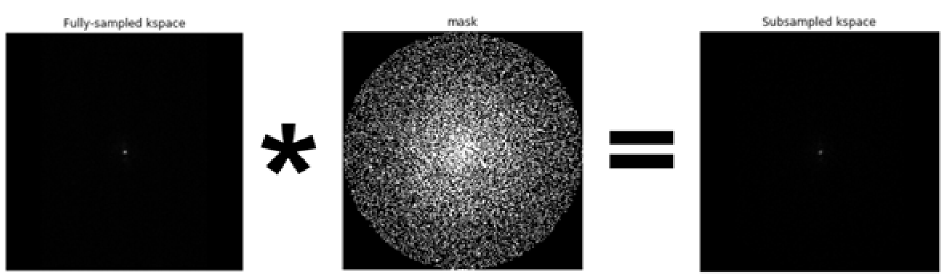

Figure 7: Undersampling fastMRI k-space

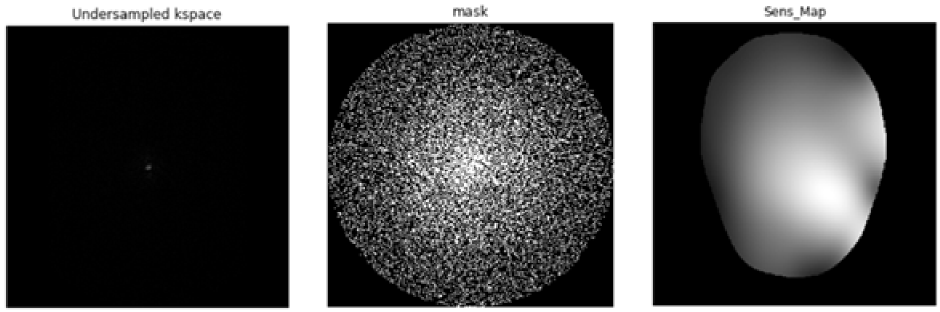

Figure 8: Sample Dataset

2.5 SSDU Training

In case of SSDU, the loss curve does not fully converge for training on 20 epochs. Therefore, for practical purposes, training should be carried out for more number of epochs but we will continue with only 20 epochs so that the training is comparable to MoDL and VarNet.

Hyperparameters	Outcome
Learning Rate = 5e-5	Train Loss = 0.367
Number of Epochs = 20	Test PSNR = 33.029
Number of Unrolls = 10	Test SSIM = 0.564

Table 3: Training and Testing Data for SSDU

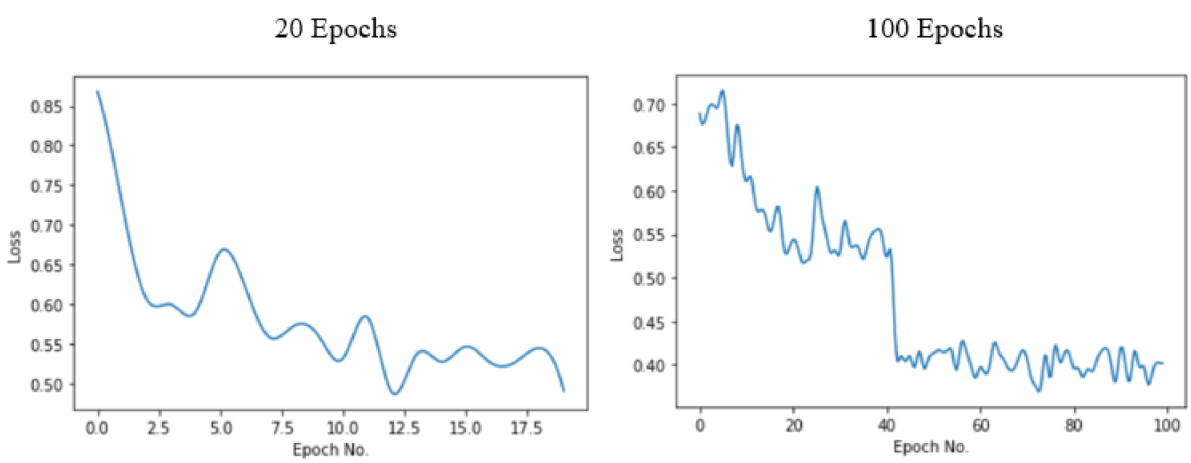

Figure 9: Loss Curves for SSDU

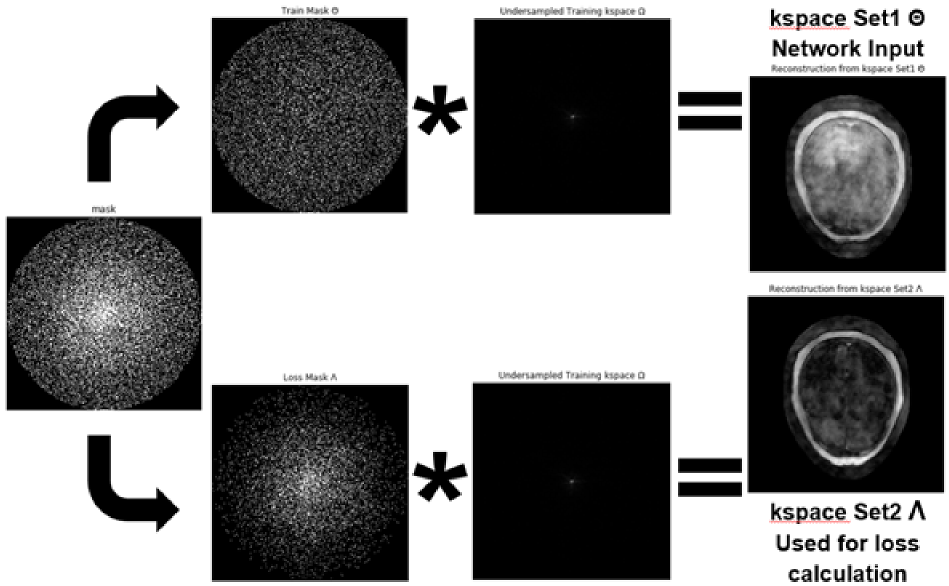

Figure 10: Inside SSDU Training

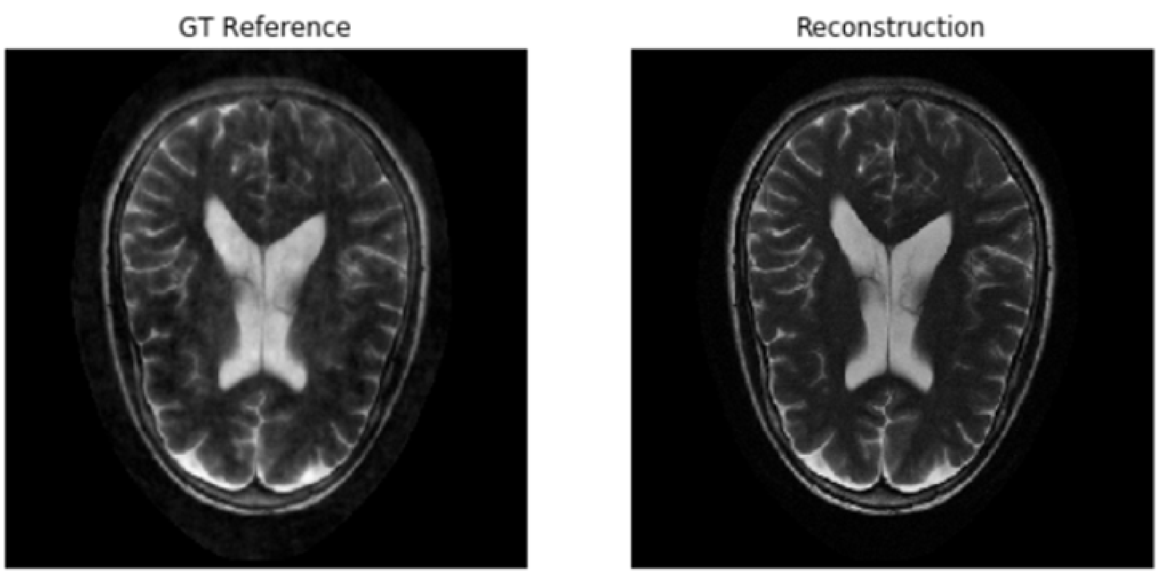

Figure 11: Sample Inference for SSDU

3 Experiments and Results

3.1 Effect of Normalization

Normalization is a key technique used in deep learning that enhances model performance and stability in the optimization process. It also eliminates the issue of vanishing or exploding gradients, allowing models to reach optimal solutions more efficiently. In our use case, normalization can be done either in the k-space or in the image space. The following table shows how the model convergence is affected by normalization.

Normalization	Train Score	Loss
No Normalization	-18.858	0.000
-1 to 1	1.010	1.243
0 to 1	9.581	0.318

Table 4: Normalization

3.2 Effect of Number of Unrolls

The general trend in all three architectures is that the model performance improves as the number of network unrolls (k) increases. A comparison between the minimum unrolls i.e. k=1 and recommended maximum unrolls i.e. k=10 is demonstrated in the following tables.

	k=1	k=10
Train Loss	0.575	4.624
Test PSNR	33.650	27.282
Test SSIM	0.836	0.784

Table 5: MoDL Metrics (for 20 Epochs)

For MoDL, the PSNR for k=1 is better than that of k=10 because the models are trained using identical hyperparameters without fine-tuning. This inconsistency is resolved by appropriately tuning the learning rate.

	k=1	k=10
Train Loss	0.205	0.191
Test PSNR	37.332	37.844
Test SSIM	0.931	0.936

Table 6: VarNet Metrics (for 20 Epochs)

	k=1	k=10
Train Loss	0.615	0.367
Test PSNR	32.785	33.030
Test SSIM	0.564	0.564

Table 7: SSDU Metrics (for 20 Epochs)

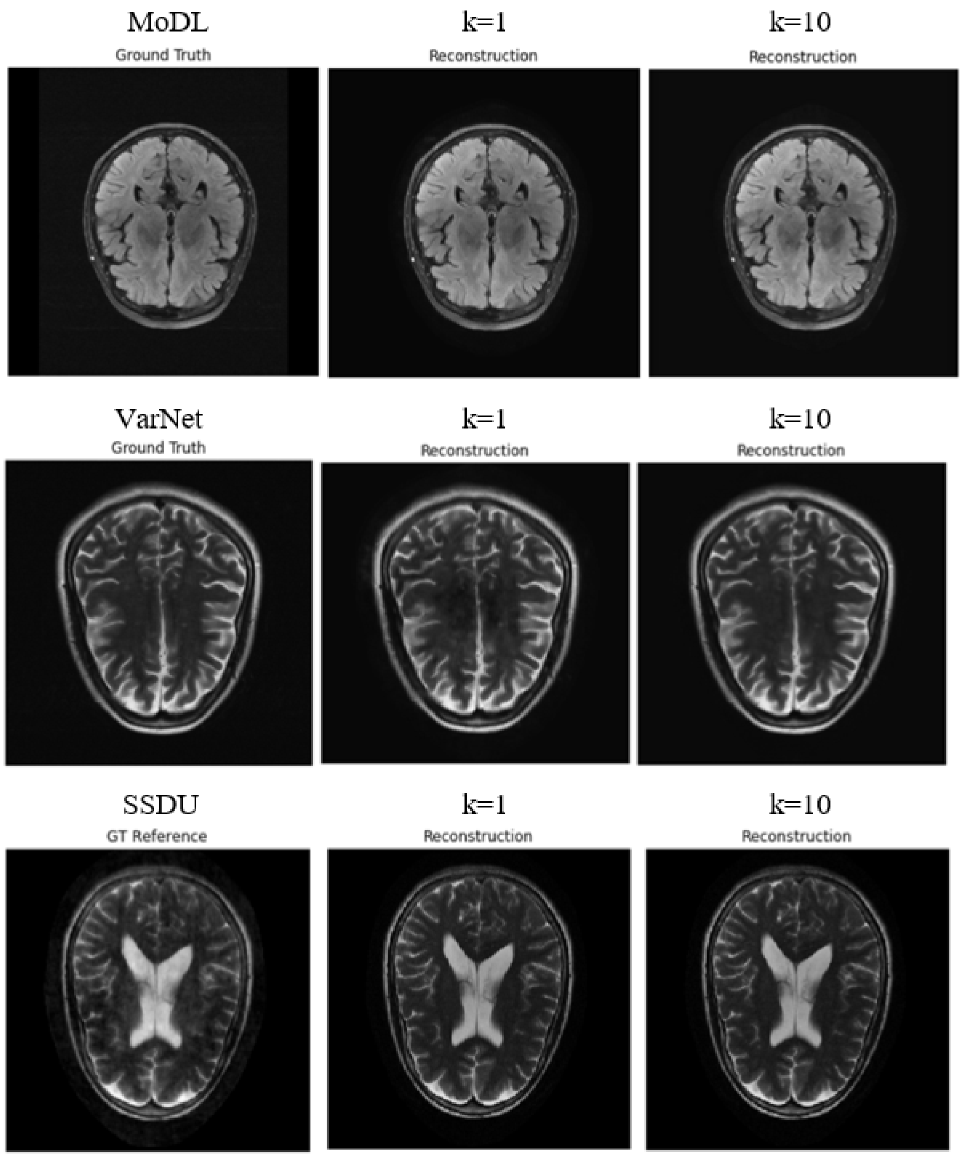

Figure 12: Inference Results for Different Unrolls

4 Discussion

As reflected in the experiments, normalization is necessary to ensure convergence in all three models. We have done normalization in image space for MoDL/VarNet and in k-space for SSDU. However, there is still a possibility to investigate the results if normalization is done in k-space. Another aspect to examine is the tuning of hyperparameters for each architecture. To make results comparable, we performed identical training without any fine-tuning. MoDL and Varnet gave good results but SSDU struggles to converge to a proper minimum. Hence, further evaluation could be done by fine-tuning the parameters. The choice of mask can also affect training. The dataset in our experiments uses a Poisson disk mask for each image slice. It would be interesting to investigate the outcome if the dataset consists of slices with different masks.

5 Conclusion

All three models give comparable results in training and inference with a minor increase in PSNR and SSIM scores as the number of unrolls of the network increases.

An overview of the three architectures is given below:

	MoDL	VarNet	SSDU
VRAM	Medium	High	High
PSNR	Better	Best	Good
SSIM	Better	Best	Good
Training Time	Good	Ok	Slow

Table 8: Comparison of the Architectures

Overall, SSDU is slow to train but is the obvious choice if only undersampled training data is available. Both MoDL and VarNet require fully-sampled data to train but differ in hardware requirements and performance. VarNet gives the best scores but requires a GPU with high memory (¿16GB) to train whereas MoDL can be trained on a GPU with a minimum 10GB VRAM. Hence, if hardware is a limiting factor, one should go more MoDL for a slight trade-off in inference results.

Acknowledgments

I would like to acknowledge support for this project from Friedrich-Alexander-Universität Erlangen-Nürnberg and my supervisor Dr. Zhengguo Tan from the Computational Imaging Lab.

Appendix A.

References

Hemant K. Aggarwal, Merry P. Mani, and Mathews Jacob. Modl: Model-based deep learning architecture for inverse problems. *IEEE TRANSACTIONS ON MEDICAL IMAGING, VOL. 38, NO. 2,*, 2019.

Kerstin Hammernik, Teresa Klatzer, Erich Kobler, Michael P. Recht, Daniel K. Sodickson, Thomas Pock, and Florian Knoll. Learning a variational network for reconstruction of accelerated mri data. *Magnetic Resonance in Medicine 79:3055 3071*, 2017.

Burhaneddin Yaman, Seyed Amir Hossein Hosseini, Steen Moeller, Jutta Ellermann, Kâmil Uğurbil, and Mehmet Akçakaya1. Self-supervised learning of physics guided reconstruction neural networks without fully sampled reference data. *University of Minnesota, Minneapolis, Minnesota*, 2020.