A study on MoDL: Model Based Deep Learning Architecture for Inverse Problems

Raj Mohan Tumarada
ratu00001@stud.uni-saarland.de

Abstract

This work is a brief study on deriving deep network-based architectures for inverse problems with any random structure based on [46]. The proposed method is a model-based reconstruction of an image with CNN (Convolution Neural Network) as regularization prior. The forward model with only a few parameters is enough to capture image information as opposed to other inversion approaches that require large training data and times. The main highlight of the proposed formulation is the use of numerical optimization like conjugate gradients(CG) to enforce data consistency within the network. Also the weight sharing mechanism of end-to-end training across iterations significantly boosts the performance by adapting CNN weights to the forward model compared to using pre-trained denoisers as regularization priors. The experiments from 3 also support in faster convergence and improved performance.

1 Introduction

MODEL-based image recovery from sparse and noisy multichannel measurements is now a focused area with fruition in several application areas such as MRI [1], CT [2],PET [3], microscopy [4]. These methods rely on a forward model which is a numerical form of measurement system. Image recovery is then presented as optimizing the consistency between the the measurements obtained and true data from the forward model. Since this recovery is ill-posed, we modify the cost function by incorporating priors as a regulariser. These priors penalise the measurements that lie outside the domain of natural images. Some of the priors include total variation [5], patch-based non-local methods, low-rank penalties [6], [7], as well as priors learned from exemplary data [8] or the measurements themselves [9]. Many recent works[12]-[14] proposed to use existing deep CNN's[10], [11] as a single framework to invert both forward model and capture the redundancy in the images. Another alternative is iterative algorithms that alternate between data consistency(DC) and pre-trained CNN denoisers[15]-[17]. End-to-end training methods [18], [19] also have an equivalent architecture as pre-trained models but here the CNN parameters are learned at every iteration and the resulting deep network recovers images from undersampled images. We use undersampling masks for reconstruction in MRI by following an iterative process rather than using fully sampled k-space data which is time consuming.

The proposed architecture, coined as (MoDL) by the authors, incorporates the strengths of model-based reconstruction methods with deep learning. By using variational framework we enforce the data-consistency term along with learned CNN to capture redundancy in images. The interleaved CNN blocks within the network learns the image set, and DC blocks enforces consistency with the measurements. The DC block comprises of a sub-problem which is quadratic that has analytical solutions for uncomplicated scenarios like single channel MRI recovery [18]. Scenarios like in multichannel MRI, where the forward model is complex, we solve the quadratic sub-problem by conjugate gradient (CG) optimization. Since the forward model lies within the network, a less-complex plug-and-play CNN as prior is enough to capture the image set as opposed

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

to some existing CNNs that do not require data consistency term. The network is trained to different forward models (e.g. sampling patterns), which makes it robust to a variety of sampling conditions.

The ability of the methods[12]-[14] to learn the inverse is exceptional while having some practical drawbacks like the point spread function has to match the receptive field of the CNN. In applications such as tomography the Fourier recovery from undersampled measurements has large point spread function that demand huge networks with many parameters such UNET. Learning such huge networks efficiently requires excessive amounts of training data and time, which is often challenging in biomedical imaging. The proposed method is similar to [18], [19] interms of the use of end-to-end training. The parameters of the network are trained for the specific task in end-to-end training which boosts the performance than using pre-trained denoisers. Especially, the CNN parameters are trained to learn the noise and alias artifacts at each iteration.

2 Methodology

2.1 Image formation and forward model

The image acquisition is like an operator $\mathcal{A}$ that transforms continuous image $x : \mathbb{R}^2 \to \mathbb{C}$ to a vector of measurements $\mathcal{A}(x) = \mathbf{b} \in \mathbb{C}^N$. The aim of reconstruction is to recover the discrete approximation represented by $\mathbf{x} \in \mathbb{R}^p$ from $\mathbf{b}$ where the approximation of $\mathcal{A}$, is denoted by the matrix $\mathbf{A}$, that maps $\mathbf{x}$ to $\mathbf{b}$;

$$\mathbf{b} = \mathbf{A}(\mathbf{x}). \tag{1}$$

While $\mathbf{A} = \mathbf{SF}$, where $\mathbf{S}$ is the sampling matrix that pick rows of the Fourier matrix, while $\mathbf{F}$ is the 2-D discrete Fourier transform,. The general approach in model-based imaging is to present the recovery as a regularized optimization problem:

$$\mathbf{x} = \arg\min_{\mathbf{x}} \underbrace{\|\mathbf{A}\,\mathbf{x} - \mathbf{b}\|_2^2}_{\text{data consistency}} + \lambda \underbrace{\mathcal{R}(\mathbf{x})}_{\text{regularization}} \tag{2}$$

The regularization prior $\mathcal{R} : \mathbb{C}^n \to \mathbb{R}_{>0}$ is choosen carefully to restrict the solutions within the image domain. $\mathcal{R}(\mathbf{x})$ has a small value when $\mathbf{x}$ is noise free and artifact-free image, while its value is large for noisy images.

2.2 Deep learned image reconstruction

Many of the deep learning algorithms recover the images as

$$\mathbf{x}_{\text{rec}} = \mathcal{T}_{\mathbf{w}}\left(\mathbf{A}^H\,\mathbf{b}\right), \tag{3}$$

where $\mathcal{T}_{\mathbf{w}}$ is a learned CNN[33]. The operator $\mathbf{A}^H(\cdot)$ transforms the measurement data from the forward model to the image domain, as CNNs work on the image domain.

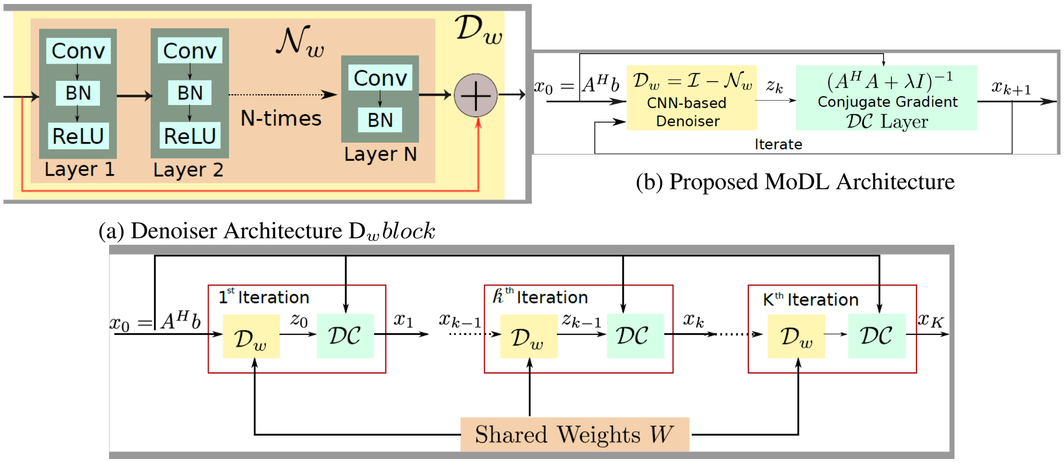
$$\mathbf{x}_{\text{rec}} = \mathcal{T}_{\mathbf{w}}\left(\mathbf{A}^H\mathbf{A}\mathbf{x}\right), \tag{4}$$

The CNN learns to invert the operator $\mathbf{A}^H\mathbf{A}$; i.e., $\mathcal{T}_{\mathbf{w}} \approx \left(\mathbf{A}^H\mathbf{A}\right)^{-1}$ for signals within the image set. The Matrix $\mathbf{A}^H\mathbf{A}$ is translation-invariant for simple measurement operators like Fourier sampling, blurring, projection imaging which makes the CNNs to solve such problems [34]. But the Matrix $\mathbf{A}^H\mathbf{A}$ is not translation-invariant for cases such as parallel MRI where we need an alternative approach involving a CNN-based regularizer like [16][35][36].

2.3 Proposed Method

We reformulate the reconstruction of the image $\mathbf{x} \in \mathbf{C}^n$ (As MRI data is captured in K-space(Frequency domain) it has both real and complex parts) as the optimization problem:

$$\mathbf{x}_{\text{rec}} = \arg\min_{\mathbf{x}} \underbrace{\|\mathcal{A}(\mathbf{x}) - \mathbf{b}\|_2^2}_{\text{data consistency}} + \lambda \underbrace{\|\mathcal{N}_{\mathbf{w}}(\mathbf{x})\|^2}_{\text{regularization}}\ . \tag{5}$$



(a) Denoiser Architecture $\mathrm{D}_w block$

(b) Proposed MoDL Architecture

(c) Unrolled Architecture

Here, $\mathcal{N}_{\mathrm{w}}$ is CNN that learns noise and alias patterns that depend on parameters $\mathbf{w}$ which is learned during training. $\mathcal{N}_{\mathbf{w}}(\mathbf{x})$ can be expressed as

$$\mathcal{N}_{\mathbf{w}}(\mathbf{x}) = (\mathcal{I} - \mathcal{D}_{\mathrm{w}})(\mathbf{x}) = \mathbf{x} - \mathcal{D}_{\mathbf{w}}(\mathbf{x}). \tag{6}$$

where $\mathcal{D}_{\mathbf{w}}(\mathbf{x})$ acts as denoiser of $\mathbf{x}$, by removing noise and alias artifacts. Substituting from (6), in (5), we obtain

$$\mathbf{x}_{\text{rec}} = \arg\min_{\mathbf{x}} \|\mathcal{A}(\mathbf{x}) - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x} - \mathcal{D}_{\mathbf{w}}(\mathbf{x})\|^2 \tag{7}$$

While these mechanisms depend on forward model, the receptive field of the framework need not match the full size of image.

The non-linear mapping $\mathcal{D}_{\mathbf{w}}(\mathbf{x}_n + \Delta\mathbf{x})$ while training can be approximated by Taylor series expansion around the n^{th} iterate as

$$\mathcal{D}_{\mathbf{w}}(\mathbf{x}_n + \Delta\mathbf{x}) \approx \underbrace{\mathcal{D}_{\mathbf{w}}(\mathbf{x}_n)}_{\mathbf{z}_n} + \mathbf{J}_n^T \nabla\mathbf{x}, \tag{8}$$

where $\mathbf{J}_n$ is the Jacobian matrix. Setting $\mathbf{x}_n + \Delta\mathbf{x} = \mathbf{x}$, the regulariser can be approximated as

$$\|\mathbf{x} - \mathcal{D}_{\mathbf{w}}(\mathbf{x}_n + \nabla\mathbf{x})\|^2 \approx \|\mathbf{x} - \mathbf{z}_n\|^2 + \|\mathbf{J}_n \Delta\mathbf{x}\|^2 \tag{9}$$

Also the second term above tends to be zero for small perturbations. Since the above approximation is applicable in the neighbourhood of $\mathbf{x}_n$, we achieve an alternating algorithm that approximates (7):

$$\mathbf{x}_{n+1} = \arg\min_{\mathbf{x}} \|\mathcal{A}(\mathbf{x}) - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x} - \mathbf{z}_n\|^2, \tag{10a}$$

$$\mathbf{z}_n = \mathcal{D}_{\mathbf{w}}(\mathbf{x}_n) \tag{10b}$$

The sub-problem (10a) can be solved using the normal equations :

$$\mathbf{x}_{n+1} = \underbrace{\left(\mathcal{A}^H \mathcal{A} + \lambda\, \mathcal{I}\right)^{-1}}_{\mathcal{Q}} \left(\mathcal{A}^H(\mathbf{b}) + \lambda\, \mathbf{z}_n\right) \tag{11}$$

The algorithm is initialized with $\mathbf{z}_0 = \mathbf{0}$. The layout of the iterative framework is depicted in Fig. 1b. Once we fix the number of iterations, the updates rules are unrolled to form deep CNN network, as shown in Fig. 1c, whose weights are shared across iterations. The unrolled architecture has similarities to [16], [18], [19] but we use the same denoiser $\mathcal{D}_{\mathbf{w}}$ and trainable regularization parameter λ for every iteration to achieve consistency with (10a) & (10b). The key benefit of the proposed method lies in reduction of network parameters. Even the experiments demonstrated the robustness of training procedure and the improved reconstruction quality.

For complex cases like multichannel MRI, the operator $\left(\mathcal{A}^H \mathcal{A} + \lambda\, \mathcal{I}\right)$ is not analytically invertible. In such scenarios, we solve (11) using conjugate gradient optimization. The CG sub-blocks result in more accurate enforcement of the DC constraint at each iteration, hence contributing to faster reduction of cost per iteration. The CG block has no trainable parameters so the gradients are backpropagated by another CG block as mentioned in (15). This suggests that large number of CG steps can be carried out with negligible memory overhead during training at each iteration.

2.4 End-to-end Training

The motto behind End-to-end training is to determine the weights $\mathbf{w}$ that need to be shared across the iterations. By using chain rule, we find the gradients of cost function with respect to the shared weights. The advantage of end-to-end training is no manual tuning of noise variance/regularization parameter is required i.e., everything is learned within the framework.

$$(\nabla_{\mathbf{w}}\mathcal{C}) = \sum_{k=0}^{K-1} \mathbf{J}_{\mathbf{w}}(\mathbf{z}_k)^T \ (\nabla_{\mathbf{z}_k}\mathcal{C}), \tag{12}$$

where the Jacobian matrix $\mathbf{J}_{\mathbf{w}}(\mathbf{z})$ has entries $[\mathbf{J}_{\mathbf{w}}(\mathbf{z})]_{i,j} = \partial z_i/\partial w_j$ and $\mathbf{z}_k$ is the output of the CNN at the k^{th} iteration.

The terms $\nabla_{\mathbf{z}_k}\mathcal{C}, k = 0, .., K-1$ are evaluated recursively using backpropagation like in any deep learning network. The main difference difference here compared to CNN training is the usage of numerical optimization blocks within the deep network. The backpropagation through conjugate gradient (CG) blocks is as follows:

$$\nabla_{\mathbf{z}_{k-1}}\mathcal{C} = \mathbf{J}_{\mathbf{z}_{k-1}}(\mathbf{x}_k)^T \ \nabla_{\mathbf{x}_k}\mathcal{C} \tag{13}$$

where the Jacobian matrix $\mathbf{J}_{\mathbf{z}}(\mathbf{x})$ has entries $[\mathbf{J}_{\mathbf{z}}(\mathbf{x})]_{i,j} = \partial x_i/\partial z_j$. Note from (11) that $\mathbf{x}_k = \mathcal{Q}\,\mathbf{z}_{k-1} + \mathbf{q}$, where $\mathcal{Q} = \left(\mathcal{A}^H\mathcal{A} + \lambda\,\mathcal{I}\right)$ and $\mathbf{q} = \mathcal{A}^H(\mathbf{b})$. The Jacobian matrix of $\mathcal{Q}$ is given by

$$\mathbf{J}_{\mathbf{z}}(\mathbf{x}) = \left(\mathcal{A}^H\mathcal{A} + \lambda\,\mathcal{I}\right)^{-1} \tag{14}$$

Since $\mathcal{Q}$ is symmetric, we have

$$(\nabla_{\mathbf{z}_{k-1}}\mathcal{C}) = \underbrace{\left(\mathcal{A}^H\mathcal{A} + \lambda\,\mathcal{I}\right)^{-1}}_{\mathcal{Q}} \ (\nabla_{\mathbf{x}_k}\mathcal{C}) \tag{15}$$

We then optimize the above expression using a CG algorithm until convergence. Note that the above gradient calculation is only valid if we implement $\left(\mathcal{A}^H\mathcal{A} + \lambda\,\mathcal{I}\right)^{-1}$ or let CG algorithm converge. Thus, the gradients can be back-propagated within CG block using CG algorithm.

2.5 Implementation

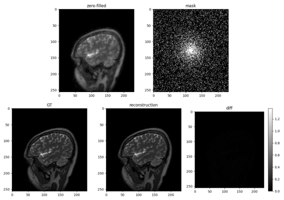
PyTorch was used to implement the above methodology. The denoiser mentioned in Fig. 1a used a 5 layer CNN architecture with 64 filters at each layer followed by Batch Normalization [41] and ReLU activation. The last layer does not have ReLU activation. The noise learnt while training is again added as input to $\mathcal{N}_w$ block to reconstruct the image. This entire process is termed as $\mathcal{D}_w$ block. The output of $\mathcal{D}_w$ block is given as input to DC block as described in Fig. 1b. This recursive model is unrolled by K iterations set to 10 for the alternating strategy (10b). The DC layer works with complex inputs i.e., undersampled K-space data and the respctive coil sensitive maps(csm), as we use Multi channel MRI image data. Here we take an image and apply Fourier Transform using coil semsitive maps and transform it to K-space. Then undersample it in K-space using sampling mask and apply Inverse Fourier Transform to bring it back to spatial domain. This way we reduce the computational burden while training as we work on low dimensional data. The CNN handles complex data by concatenating the real and imaginary part as channels i.e. we convert from $\mathbb{C}^{m\times n}$ space into $\mathbb{R}^{m\times n\times 2}$ space.
The training was done based on a two-step approach, by first training the model for one iteration with random initialization of parameters. In order to facilitate weight sharing across iterations of the unrolled network, we use the weights of single iteration trained model at every iteration. This approach significantly accelerates and stabilizes the training process rather than initialising the whole network with random weights every iteration.

3 Experiments

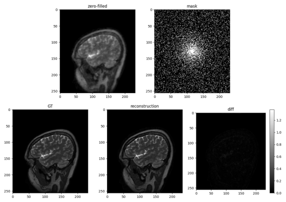
3.1 Dataset

The MRI data used for this study is the same as mentioned in [46]. The images were acquired using a 3D T2 CUBE sequence with Cartesian readouts using a 12-channel sensitive maps (CSM). We selected only 90 slices out of 208 because they had parts of anatomy for training. The sampling masks were acquired using variable-density cartesian random sampling which is generally reconstructed using a technique called ESPRiT [42]. The same sampling mask was used for the data from all 12 coils of the same slice. The training data had dimensions in rows × columns × slices × coils as $256 \times 232 \times 360 \times 12$ and testing data had dimensions $256 \times 232 \times 164 \times 12$.

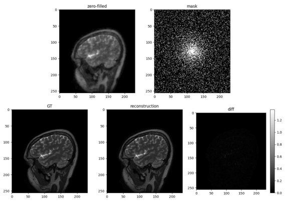
3.2 Experimental Setup



(a) Reconstruction from training on CG-ET-WS



(b) Reconstruction from training on CG-PD-NS



(c) Reconstruction from training on CG-ET-NS

The experiments were carried out based on the training strategy and network architecture mentioned in 3.2. Only Conjugate Gradient(CG) optimization was used in all experiments within MoDL Architecture. The experiments yielded the following 3 configurations CG-ET-WS, CG-PD-NS, CG-ET-NS. All the experiments were performed on the same dataset mentioned in 3.1. For *CG-ET-WS* as proposed in the framework, I used CG to optimize the network with end-to-end training by sharing the same weights across all iterations. The regularization parameter λ is set to trainable, so no need of hyperparameter tuning here. By sharing weights the numbers of parameters reduced significantly. The configuration *CG-PD-NS* is another strategy wherein we use pre-trained denoisers based on CNN without end-to-end training. The denoisers were trained separately for different noise levels within the same architecture by freezing the DC block. Had to use this strategy mentioned in [9] as there are no readily available pre-trained denoisers for the dataset. A total of 10 such models were trained to match the number of iterations between noise levels 0.02 and 0.2 with 0.02 step size. The particular noise values used are: 0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.14, 0.16, 0.18, 0.2. The regularization parameter λ was set to 0.05 here. The Configuration *CG-ET-NS* was same as the setting *CG-ET-WS* except that we are not sharing the weights across iterations.

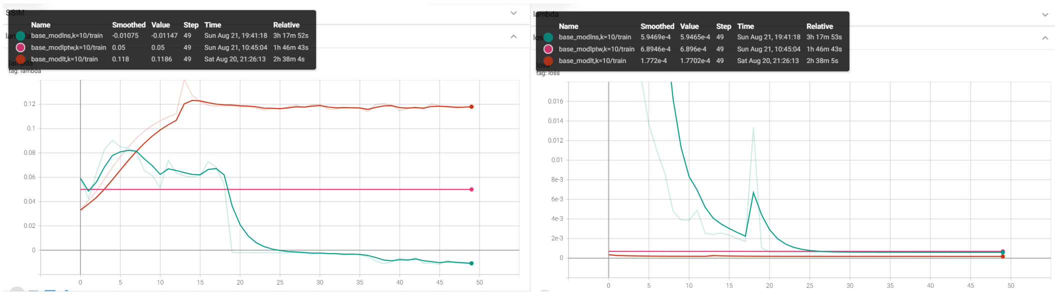
All the experiments were trained for 50 epochs each with batch size 1. ADAM optimizer with learning rate of 0.001 is used to optimize the framework along with CNN denoiser. Learning rate scheduler hasn't helped in this case as there are no significant gains. The total number of parameters and training time were shown in Table 3.2. Especially, the number of parameters reduced by a factor of the number of iterations. The difference in parameters between CG-PD-NS and CG-ET-NS is that λ is not trainable for PD case. For a given zero-filled frame($\mathbf{A}^H$ $\mathbf{b}$) and sampling mask the reconstructions while training were as shown in 2a, 2b, 2c (Please zoom in for details).

Training Strategy	**Network Architecture**
End-to-End Training(ET)	With Sharing(WS)
Pre-trained Denoiser(PD)	No Sharing(NS)

Table 1: Experimental Categories

Configuration	**Parameters(K=10)**	**Training Time(hrs)**
CG-ET-WS	113671	2.38
CG-PD-NS	1136700	1.46
CG-ET-NS	1136710	3.17

Table 2: Parameters and Training time



	Name	Smoothed	Value	Step	Time	Relative
●	base_modlns,k=10/train	-0.01075	-0.01147	49	Sun Aug 21, 19:41:18	3h 17m 52s
●	base_modlptw,k=10/train	0.05	0.05	49	Sun Aug 21, 10:45:04	1h 46m 43s
●	base_modlt,k=10/train	0.118	0.1186	49	Sat Aug 20, 21:26:13	2h 38m 4s

	Name	Smoothed	Value	Step	Time	Relative
●	base_modlns,k=10/train	5.9469e-4	5.9465e-4	49	Sun Aug 21, 19:41:18	3h 17m 53s
●	base_modlptw,k=10/train	6.8946e-4	6.896e-4	49	Sun Aug 21, 10:45:04	1h 46m 43s
●	base_modlt,k=10/train	1.772e-4	1.7702e-4	49	Sat Aug 20, 21:26:13	2h 38m 5s

(a) λ while training

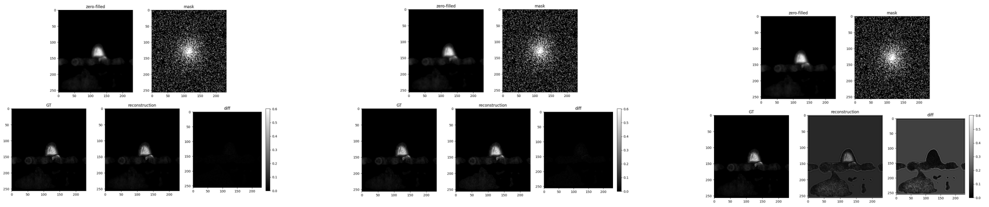
(b) Training loss

Table 3: Results. The best values are coded green and the worst in red

Configuration	PSNR	SSIM
CG-ET-WS	39.9917	0.9844
CG-PD-NS	34.5346	0.9562
CG-ET-NS	30.2538	0.8119

4 Results

To show the quantitative performance of the models, used PSNR(peak signal to noise ratio) and SSIM(Structural similarity index measure) which are the commonly used metrics for testing reconstruction quality (the higher the better for both metrics). As can be seen from 3 the CG-ET-WS configuration gave us the best results on both PSNR and SSIM as expected. From the plot 3a, we can see in CG-ET-NS setting, the training is unstable and pushed λ, the regularization parameter, towards negative. In Machine Learning terminology, this can be inferred as overfitting as we are making the weights as high as possible which led to poor results on the test set. See4c the fifth image in the block shows that the difference of reconstructed image from the Ground Truth. The image is not blank which can be inferred as poor reconstruction quality in this setting compared to other approaches where there is no risk of overfitting. On the other hand, CG-PD-NS setup did not give any significant gains compared to CG-ET-WS.



(a) Recontruction on test set from CG-ET-WS

(b) Recontruction on test set CG-PD-NS

(c) Recontruction on test set CG-ET-NS

5 Conclusion

The proposed MoDL framework merges model-driven approaches with data-hungry deep learning. This framework is powerful and acts as a precursor for deriving deep architerctures for inverse problems. The CG algorithm within DC blocks allows to optimize even the complex forward models where it is hard to compute the analytical inverse. This also allows us to incorporate additional image priors wherever applicable. The weight sharing mechanism of end-to-end strategy benefits from decoupling convergence from complexity of architecture. The scalability of the network to perform more iterations without the necessity of more data mitigates the risk of overfitting. The results of the model are encouraging even with less number of parameters.

6 References

[1] J. A. Fessler, “Model-Based Image Reconstruction for MRI,” IEEE Signal Process. Mag., vol. 27, no. 4, pp. 81–89, 2010.

[2] I. A. Elbakri and J. A. Fessler, “Statistical image reconstruction for polyenergetic x-ray computed tomography,” IEEE Trans. Med. Imag., vol. 21, no. 2, pp. 89–99, 2002.

[3] J. Verhaeghe, D. Van De Ville, I. Khalidov, Y. D’Asseler, I. Lemahieu, and M. Unser, “Dynamic PET reconstruction using wavelet regularization with adapted basis functions,” IEEE Trans. Med. Imag., vol. 27, no. 7, pp. 943–959, July 2008.

[4] F. Aguet, D. Van De Ville, and M. Unser, “Model-based 2.5-D deconvolution for extended depth of field in brightfield microscopy,” IEEE Trans. Image Process., vol. 17, no. 7, pp. 1144–1153, July 2008.

[5] S. Ma, W. Yin, Y. Zhang, and A. Chakraborty, “An Efficient Algorithm for Compressed MR Imaging using Total Variation and Wavelets,” in Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[6] S. G. Lingala, Y. Hu, E. DiBella, and M. Jacob, “Accelerated dynamic MRI exploiting sparsity and low-rank structure: kt SLR,” IEEE Trans. Med. Imag., vol. 30, no. 5, pp. 1042–1054, 2011.

[7] S. G. Lingala and M. Jacob, “A blind compressive sensing frame work for accelerated dynamic mri,” in IEEE Int. Symp. Bio. Imag. IEEE, 2012, pp. 1060–1063.

S. Ravishankar and Y. Bresler, “L0 Sparsifying Transform Learning With Efficient Optimal Updates and Convergence Guarantees,” IEEE Trans. Signal Process., vol. 63, no. 9, pp. 2389–2404, 2015.

[9] S. G. Lingala and M. Jacob, “Blind Compressive Sensing Dynamic MRI,” IEEE Trans. Med. Imag., vol. 32, no. 6, pp. 1132–1145, 2013.

[10] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer, 2015, pp. 234–241.

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[12] H. Chen, Y. Zhang, M. K. Kalra, F. Lin, Y. Chen, P. Liao, J. Zhou, and G. Wang, “Low-Dose CT with a Residual Encoder-Decoder Convolutional Neural Network,” IEEE Trans. Med. Imag., vol. 36, no. 12, pp. 2524–2535, 2017.

[13] G. Wang, “A Perspective on Deep Imaging,” IEEE Access, vol. 4, no. nn, pp. 8914–8924, 2016.

[14] K. Zhang, W. Zuo, S. Member, Y. Chen, D. Meng, and L. Zhang, “Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising,” IEEE Trans. Image Process., vol. 26, no. 7, pp. 3142– 3155, 2017.

[15] L. Zhang and W. Zuo, “Image Restoration: From Sparse and Low-Rank Priors to Deep Priors,” IEEE Signal Process. Mag., vol. 34, no. 5, pp. 172–179, 2017.

[16] K. Zhang, W. Zuo, S. Gu, and L. Zhang, “Learning Deep CNN Denoiser Prior for Image Restoration,” in IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2808–2817.

[17] J. H. R. Chang, C.-L. Li, B. Poczos, B. V. K. V. Kumar, and A. C. Sankaranarayanan, “One Network to Solve Them All — Solving Linear Inverse Problems using Deep Projection Models,” in IEEE International Conference on er Vision, 2017, pp. 1–12.

[18] J. Schlemper, J. Caballero, J. V. Hajnal, A. Price, and D. Rueckert, “A Deep Cascade of Convolutional Neural Networks for MR Image Reconstruction,” in Information Processing in Medical Imaging, 2017, pp. 647–658.

[19] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll, “Learning a Variational Network for Reconstruction of Accelerated MRI Data,” Magnetic resonance in Medicine, vol. 79, no. 6, pp. 3055–3071, 2017.

[20] M. Mardani, H. Monajemi, V. Papyan, S. Vasanawala, D. Donoho, and J. Pauly, “Recurrent Generative Adversarial Networks for Proximal Learning and Automated Compressive Image Recovery,” in IEEE Conference on Computer Vision and Pattern Recognition, 2018, p. na.

[21] P. Putzky and M. Willing, “Recurrent Inference Machines for Solving Inverse Problems,” in arXiv, 2017, pp. 1–12.

[22] K. Gregor and Y. LeCun, “Learning fast approximations of sparse coding,” in International Conference on Machine Learning. Omnipress, 2010, pp. 399–406.

[23] S. Poddar and M. Jacob, “Dynamic mri using smoothness regularization on manifolds (storm),” IEEE Trans. Med. Imag., vol. 35, no. 4, pp. 1106– 1115, 2016.

[24] S. Biswas, H. K. Aggarwal, S. Poddar, and M. Jacob, “ Modelbased free-breathing cardiac MRI reconstruction using deep learned and STORM priors: MoDL-STORM,” in IEEE International Conference on Acoustics, Speech, and Signal Processing, 2018, p. NA.

[25] G. Ongie and M. Jacob, “Super-resolution mri using finite rate of innovation curves,” in IEEE Int. Symp. Bio. Imag. IEEE, 2015, pp. 1248–1251.

[26] G. Ongie, S. Biswas, and M. Jacob, “Convex recovery of continuous domain piecewise constant images from non-uniform Fourier samples,” IEEE Trans. Signal Process., vol. 66, no. 1, pp. 236–250, 2017.

[27] M. A. T. Figueiredo, R. D. Nowak, S. Member, and R. D. Nowak, “An EM Algorithm for Wavelet-Based Image Restoration,” IEEE Trans. Image Process., vol. 12, no. 8, pp. 906–916, 2003.

[28] Y. Hu and M. Jacob, “Higher Degree Total Variation (HDTV) Regularization for Image Recovery,” IEEE Trans. Image Process., vol. 21, no. 5, pp. 2259–2271, 2012.

[29] G. Ongie and M. Jacob, “Off-the-Grid Recovery of Piecewise Constant Images from Few Fourier Samples,” SIAM on Imag. Sci., vol. 9, no. 3, pp. 1004—-1041, 2016.

[30] D. Lee, K. H. Jin, E. Y. Kim, S.-H. Park, and J. C. Ye, “Acceleration of mr parameter mapping using annihilating filter-based low rank hankel matrix (aloha),” Magnetic resonance in medicine, vol. 76, no. 6, pp. 1848–1864, 2016.

[31] J. P. Haldar, “Low-Rank Modeling of Local k-Space Neighborhoods (LORAKS) for Constrained MRI,” IEEE Trans. Med. Imag., vol. 33, no. 3, pp. 668–681, 2014.

[32] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering,” IEEE Trans. Image Process., vol. 16, no. 8, pp. 2080–2095, 2007.

[33] A. Mousavi and R. G. Baraniuk, “Learning to Invert: Signal Recovery via Deep Convolutional Networks,” in IEEE Intel. Conf.Acou., Speech, Sig Proces, 2017, pp. 2272–2276.

[34] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, “Deep Convolutional Neural Network for Inverse Problems in Imaging,” IEEE Trans. Image Process., vol. 29, pp. 4509–4522, 2017.

[35] J. Schlemper, J. Caballero, J. V. Hajnal, A. N. Price, and D. Rueckert, “A deep cascade of convolutional neural networks for dynamic mr image reconstruction,” IEEE Trans. Med. Imag., vol. 37, no. 2, pp. 491–503, 2018.

[36] S. Diamond, V. Sitzmann, F. Heide, and G. Wetzstein, “Unrolled Optimization with Deep Priors,” in arXiv:1705.08041, 2017, pp. 1–11.

[37] S. H. Chan, X. Wang, and O. A. Elgendy, “Plug-and-Play ADMM for Image Restoration: Fixed Point Convergence and Applications,” IEEE Transactions on Computational Imaging, vol. 3, no. 1, pp. 84–98, 2017.

[38] y. yang, J. Sun, H. Li, and Z. Xu, “Deep admm-net for compressive sensing mri,” in Advances in Neural Information Processing Systems 29, 2016, pp. 10–18.

[39] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, “Numerical recipes in c,” Cambridge University Press, vol. 1, p. 3, 1988.

[40] D. P. Kingma and J. L. Ba, “Adam: a Method for Stochastic Optimization,” International Conference on Learning Representations 2015, pp.1–15, 2015.

[41] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in IEEE International Conference on Machine Learning, 2015, pp. 448–456. [Online]. Available: http://arxiv.org/abs/1502.03167

[42] M. Uecker, P. Lai, M. J. Murphy, P. Virtue, M. Elad, J. M. Pauly, S. S.Vasanawala, and M. Lustig, “ESPIRiT - An eigenvalue approach to autocalibrating parallel MRI: Where SENSE meets GRAPPA,” Magnetic Resonance in Medicine, vol. 71, no. 3, pp. 990–1001, 2014.

[43] E. Kang, J. Min, and J. C. Ye, “A deep convolutional neural network using directional wavelets for low-dose x-ray ct reconstruction,” Medical Physics, vol. 44, no. 10, pp. e360–e375, 2017.

[44] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers,” Foundations and Trends R in Machine Learning, vol. 3, no. 1, pp. 1–122, 2010.

[45] A. Beck and M. Teboulle, “A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems,” SIAM J. Imag. Sci., vol. 2, no. 1, pp. 183–202, 2009.

[46]Aggarwal, Hemant K., Merry P. Mani, and Mathews Jacob. "MoDL: Model-based deep learning architecture for inverse problems." IEEE transactions on medical imaging 38, no. 2 (2018): 394-405.

A Appendices