Energy Dissipation with Plug-and-Play Priors

Hendrik Sommerhoff[1] **Andreas Kolb**[1] **Michael Moeller**[2]
[1]Computer Graphics Group, University of Siegen
[2]Computer Vision Group, University of Siegen
{hendrik.sommerhoff, andreas.kolb, michael.moeller}@uni-siegen.de

Abstract

Neural networks have reached outstanding performance for solving various ill-posed inverse problems in imaging. However, drawbacks of end-to-end learning approaches in comparison to classical variational methods are the requirement of expensive retraining for even slightly different problem statements and the lack of provable error bounds during inference. Recent works tackled the first problem by using networks trained for Gaussian image denoising as generic plug-and-play regularizers in energy minimization algorithms. Even though this obtains state-of-the-art results on many tasks, heavy restrictions on the network architecture have to be made if provable convergence of the underlying fixed point iteration is a requirement. More recent work has proposed to train networks to output descent directions with respect to a given energy function with a provable guarantee of convergence to a minimizer of that energy. However, each problem and energy requires the training of a separate network. In this paper we consider the combination of both approaches by projecting the outputs of a plug-and-play denoising network onto the cone of descent directions to a given energy. This way, a single pre-trained network can be used for a wide variety of reconstruction tasks. Our results show improvements compared to classical energy minimization methods while still having provable convergence guarantees.

1 Introduction

In many image processing tasks an observed image f is modeled as the result of the transformation of a clean image $\hat{u}$ under a known (linear) operator A and unknown noise ξ,

$$f = A\hat{u} + \xi. \tag{1}$$

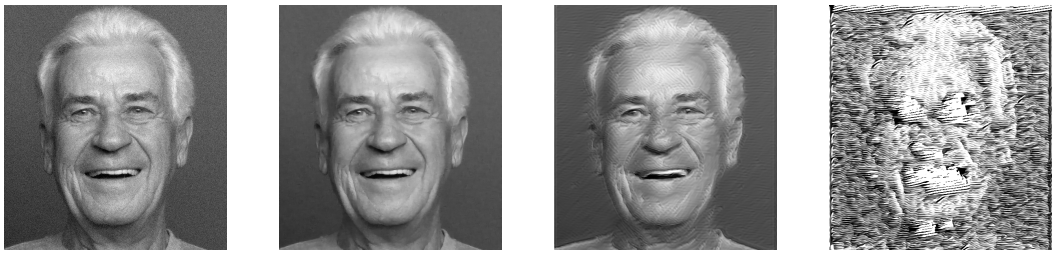
In most cases, the problem of reconstructing $\hat{u}$ from f and A is ill-posed and can thus not be solved by a simple inversion of A, giving rise to the field of regularization theory with iterative or variational methods, see e.g. [2] for an overview. In recent years neural networks were very successful in learning a direct mapping $\mathcal{G}(f) \approx \hat{u}$ for a variety of problems such as deblurring [32, 28], denoising [34], super-resolution [8], demosaicing [9] and MRI- or CT-reconstruction [33, 14]. Even though this works well in practice, there are rarely any guarantees on the behaviour of neural networks on unseen data, making them difficult to use in safety-critical applications. Moreover, for each problem and type of noise a separate network has to be trained.

In contrast, classical variational methods try to find the solution by the minimization of a suitable energy function of the form

$$\hat{u} = \operatorname{argmin}_u H_f(u) + \alpha R(u) \tag{2}$$

where H_f is a data fidelity term, for example commonly chosen as $H_f(u) = \frac{1}{2}||Au - f||^2$, and R is a regularization function that models prior knowledge about the solution, e.g. the popular total variation (TV) regularization, $R(u) = \|\nabla u\|_1$, [24]. While minimizers of (2) come with many desirable theoretical guarantees, regularizations like the TV often cannot perfectly capture the complex structure of the space of natural images.

NeurIPS 2019 Workshop on Solving Inverse Problems with Deep Networks



(a) Noisy (b) Iteration 1 (c) Iteration 100 (d) Iteration 800

Figure 1: Example for a diverging algorithmic scheme using (6) with $H_f = 0$ and a DnCNN denoising network $\mathcal{G}$ (for blind denoising). Pixel values are clipped to [0,1] for visualization. Image from the FACES dataset `https://faces.mpdl.mpg.de`.

To combine the advantages of powerful feed-forward networks and model-based approaches like (2), authors have considered various hybrid models like learning regularizers (e.g. [23, 1, 11, 5]), designing networks architectures that resemble the structure of minimization algorithms or differential equations, e.g. [25, 36, 15, 6], interleaving networks with classical optimization steps [16, 17], or using the parametrization of networks as a regularization for (2), see e.g. [29, 12].

A particularly flexible approach arises from [7, 37, 30, 13], where proximal operators with respect to the regularizer are replaced by arbitrary denoising operators, with recent works focusing on the use of denoising networks [18, 4, 35]. While such approaches allow to tackle different inverse problems with the same neural network, the derivation of theoretical guarantees - even in terms of the convergence of the resulting algorithmic scheme - remains difficult, see [3, 27] or some discussion in [20], unless the denoiser satisfies particular properties [22].

2 Provably dissipating model-based energies with plug-and-play priors

The starting point of the above-mentioned algorithmic schemes that utilize denoising networks to regularize model-based inverse problems are methods for the minimization of (2). While most works focus on primal-dual / ADMM approaches, their convergence analysis is quite delicate even in a setting in which one still minimizes (nonconvex) energies, such that we turn to two simpler methods, gradient descent and proximal gradient methods,

$$u^{k+1} = u^k - \tau \left(\nabla H_f(u^k) + \alpha \nabla R(u^k) \right), \tag{3}$$

$$u^{k+1} = \operatorname{prox}_{\tau R}(u^k - \tau \nabla H_f(u^k)), \tag{4}$$

where $\operatorname{prox}_{\tau R}(v) = \operatorname{argmin}_u \frac{1}{2}\|u - v\|^2 + \tau R(u)$. Following the idea of [7, 37, 30, 13], considering either a gradient descent or a proximal step on the regularization as a generic denoising operation gives rise to the following two algorithmic schemes,

$$u^{k+1} = \frac{1}{2}\rho(u^k; 2\tau) + \frac{1}{2}\mathcal{G}(u^k), \tag{5}$$

$$u^{k+1} = \mathcal{G}(\rho(u^k; \tau)), \tag{6}$$

where $\mathcal{G}$ denotes any kind of denoiser, e.g. a convolutional neural network, and we define $\rho(u^k; \tau) = u^k - \tau \nabla H_f(u^k)$ for the sake of brevity of notation. We refer to [20] for a more detailed derivation.

Algorithmic schemes like (5) or (6) combine the model-based flexibility of energy minimization methods (i.e. explicit modelling of H_f) with the expressive power of deep neural networks $\mathcal{G}$.

Unfortunately – despite their success in various practical applications – schemes like (5) or (6) remain dangerous to be used: Figure 1 shows the result of running the iteration (6) with $H_f = 0$ on a noisy input image $f = u^0$ for 100 and 800 iterations using a DnCNN [34] preimplemented in Matlab as the denoiser $\mathcal{G}$. As we can see the image gets completely distorted. Even more strikingly, the range of the image increased from values in $[0, 1]$ to an interval of $[-185, 218]$ within the first 1000 iterations. Clearly, the algorithmic scheme diverges.

A natural condition for the provable convergence of a scheme like (6) (at least along subsequences) would be a 1-Lipschitz continuous operator $\mathcal{G}$. There has been previous work on computing upper

bounds for the best Lipschitz constant of a network and using it to enforce a user defined Lipschitz constant L during training time [10, 26, 19] but we found that enforcing non-expansiveness drastically decreased the denoising performance. The problem of computing the best Lipschitz constant, in hope of improving those results, was recently proved to be NP-hard [31] and thus is infeasible.

Therefore, we adapt the recent idea proposed in [21] to safeguard neural networks by forcing them to predict a descent direction to a given model-based energy, such that it can be used within a line search algorithm to guarantee convergence. More precisely, at any given estimate u and model-based energy E the authors use the Euclidean projection onto the half space

$$\mathcal{C}(\gamma, \nabla E(u)) = \{d | \langle d, \nabla E(u) \rangle \geq \gamma ||\nabla E(u)||\}, \quad \gamma > 0 \tag{7}$$

as the last layer of their network. Even though the resulting algorithm converges to the minimizer of E, experiments showed significantly higher peaks of the PSNR value in early iterations compared to classical gradient descent on E. Intuitively, the descent direction proposed by the network pushes the iteration closer towards the distribution of the training data than a usual gradient descent step.

While the approach of [21] has to train a separate network for each inverse problem and each type of noise, we investigate the combination of the flexible algorithmic schemes (5) and (6) with the idea from [21] to project onto the half-space of descent directions to safeguard the underlying algorithm.

In the following $\mathcal{G}$ will always refer to a generic denoising network, like DnCNN [34]. We assume that $E(u) = H_f(u) + R(u)$ is a continuously differentiable, strictly convex and coercive energy function. As a first step, we simply rewrite the algorithmic schemes (5) and (6) in such a way that they resemble a gradient descent iteration, i.e.,

$$u^{k+1} = u^k - \left(u^k - \frac{1}{2}\rho(u^k; 2\tau) - \frac{1}{2}\mathcal{G}(u^k) \right), \tag{8}$$

$$u^{k+1} = u^k - \left(u^k - \mathcal{G}(\rho(u^k; \tau)) \right), \tag{9}$$

such that we can interpret $u^k - \frac{1}{2}\rho(u^k; 2\tau) - \frac{1}{2}\mathcal{G}(u^k)$ or $u^k - \mathcal{G}(\rho(u^k; \tau))$ as "update directions" of the respective algorithmic schemes. Because the plain iterations (8) and (9) can easily be divergent, we safeguard them by projecting them onto the half-space of descent directions $\mathcal{C}(\gamma, \nabla E(u^k))$, i.e.,

$$d^k = \text{proj}_{\mathcal{C}(\gamma, \nabla E(u^k))} \left(u^k - \alpha\rho(u^k; 2\tau) - (1-\alpha)\mathcal{G}(u^k) \right), \tag{conv}$$

$$\text{or} \quad d^k = \text{proj}_{\mathcal{C}(\gamma, \nabla E(u^k))} \left(u^k - \mathcal{G}(\rho(u^k; \tau)) \right). \tag{prox}$$

Note that we replaced the averaging of the gradient descent and denoising step in (8) by an abitrary convex combination using a parameter α to determine the respective influence of the data term and the denoising more flexibly. After computing the above directions d^k, we update our iterates using

$$u^{k+1} = u^k - t^k d^k \tag{10}$$

with a step size t^k chosen based on a backtracking line-search mechanism similar to [21]. Under weak additional conditions, the latter guarantees the convergence of the proposed scheme to the minimizer of E. Such a minimizer could of course be determined by any classical algorithm, but we hope for (10) to yield a better path towards the true minimizer, and consider a *discrepancy principle* for stopping the iteration before convergence. More precisely, we terminate (10) as soon as

$$H_f(u^k) \leq \beta H_f(\hat{u}) \tag{11}$$

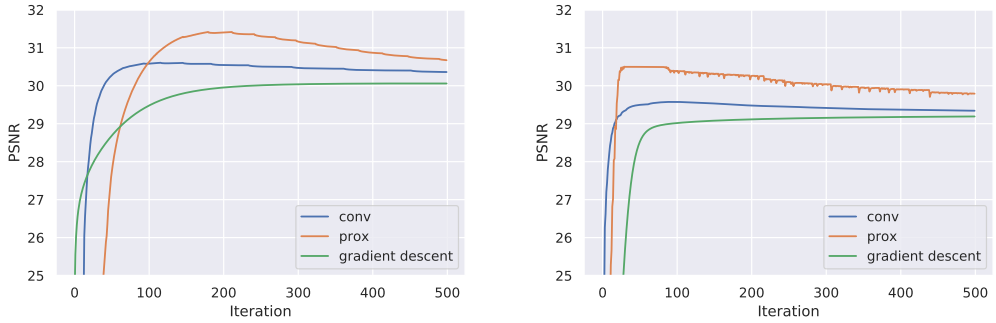
for $H_f(\hat{u})$ being an estimate on the (data-term-dependent measure of the) noise level of the considered problem, and β being a scaling factor (typically close to 1).

3 Results

We tested our implementation with the image reconstruction tasks of Gaussian deblurring with standard deviation 1 and $4\times$ single image super resolution. In both cases we added Gaussian noise with standard deviation 0.02 to the corrupted image. We chose the PyTorch implementation[1] of DnCNN [34] pre-trained on a noise level of 0.1 as our denoising network.

Our surrogate energy uses a TV regularization with Huber-norm instead of the ℓ^1-norm. As our data term we choose $H_f(u) = \frac{1}{2}||Au - f||^2$ with A either being the blurring or downsampling

[1] `https://github.com/SaoYan/DnCNN-PyTorch`



(a) deblurring (b) super resolution

Figure 2: Exemplary PSNR values per iteration for the deblurring and super resolution experiments.

operator. The best hyperparameters for all methods were found with a grid search. In all experiments, for scheme (conv), $\alpha = 0$ was the best choice for any τ, indicating that the gradient descent step on the data term does not yield much additional information, assumably because the projection onto $\mathcal{C}(\gamma, \nabla E(u))$ which depends on the gradient of the data term anyway. When using (prox), we empirically found $\tau = 30$ to be the best choice. For the projection onto the half-space of descent directions, we we used $\gamma = 5$ for both methods for deblurring, and $\gamma = 50$ in (conv) and $\gamma = 1$ in (prox) for super resolution. For fairness, the classical gradient descent was also implemented using backtracking line search.

Figure 2 shows the reconstruction quality of the current iterate compared to ground truth over a span of 500 iterations. The PSNR quickly peaks before slowly converging to the fixed point of the surrogate energy which is consistent with the results of [21]. Notably, the convex combination method peaks earlier but not as high as the prox method. Tables 1 and 2 show results using early stopping using a discrepancy principle. On all test images our prox scheme beats gradient descent.

Method	Camerman	House	Peppers	Starfish	Butterfly	Plane	Bird	Lena	Barbara	Boat	Man	Couple
conv, $\beta = 1$	27.47	32.76	29.08	29.45	30.45	27.53	28.45	32.52	25.21	30.04	30.78	29.84
conv, $\beta = 0.9$	28.40	33.32	30.57	29.79	31.21	28.58	29.30	33.49	25.58*	30.79	31.34	30.50
conv, best	28.44	33.36	30.67	29.86	31.23	28.62	29.32	33.76	25.61	30.84	31.41	30.62
prox, $\beta = 1$	28.01	32.17	27.72	29.44	29.96	27.46	29.14	32.73	**25.85**	30.71	30.78	30.50
prox, $\beta = 0.9$	**29.47**	**34.21**	**31.05**	**31.33**	**32.17**	**29.12**	**30.05**	**34.16**	25.64*	**31.64**	**31.87**	**31.41**
prox, best	29.48	34.33	31.40	31.33	32.18	29.16	30.05	34.30	26.88	31.66	31.88	31.42
GD, best	28.19	32.90	30.15	29.55	30.76	28.23	28.95	33.24	25.63	30.64	31.18	30.48

Table 1: PSNR values for deblurring for varying images and stopping criteria. The algorithm was stopped when $H_f(u^k) < \beta H_f(\hat{u})$. *best* refers to the highest PSNR over 500 iterations and a "*" means that the stopping criterion was not triggered such that the last iteration was used instead.

Method	Cameraman	House	Peppers	Starfish	Butterfly	Plane	Bird	Lena	Barbara	Boat	Man	Couple
conv, $\beta = 1$	21.56	24.10	22.13	22.06	22.20	20.74	20.85	24.88	21.97	25.22	26.35	24.94
conv, $\beta = 0.9$	21.56	24.10	22.13	22.06	22.20	21.27	21.37	28.36	23.40	25.31	26.48	25.02
conv, best	23.00	27.50	23.67	23.60	23.53	22.34	22.50	28.49	23.45	25.32	26.56	25.06
prox, $\beta = 1$	23.43	28.57	24.61	24.41	24.98	22.86	23.27	29.21	23.60	25.87	26.88	25.39
prox, $\beta = 0.9$	**23.50**	**28.78**	**24.62**	**24.54**	**25.09**	**22.91**	**23.35**	**29.38**	**23.67**	**25.96**	**27.05**	**25.40**
prox, best	23.52	28.87	24.72	24.58	25.18	22.93	23.46	29.45	23.72	26.00	27.16	25.53
GD, best	22.21	26.55	22.94	23.01	21.85	21.87	21.40	27.65	23.22	24.94	26.08	24.82

Table 2: PSNR values for super resolution for varying images and stopping criteria. The algorithm was stopped when $H_f(u^k) < \beta H_f(\hat{u})$. *best* refers to the highest PSNR over 500 iterations.

4 Conclusion

We combine deep learning and energy minimization methods for solving inverse problems in image reconstruction into a provably convergent algorithmic scheme. Still, our approach is able to generalize to different problems with a single denoising network and without the need to retrain if that problem changes. We were able to reach better results than the energy minimization baseline in our experiments, and are happy to elaborate on the above aspects in the NeurIPS workshop.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[2] M. Benning and M. Burger. Modern regularization methods for inverse problems. *Acta Numerica*, 27:1–111, 2018.

[3] 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*, 3:84–98, 2016.

[4] J. R. Chang, C.-L. Li, B. Poczos, B. V. Kumar, and A. C. Sankaranarayanan. One network to solve them all — solving linear inverse problems using deep projection models. In *International Conference on Computer Vision (ICCV)*, 2017.

[5] Y. Chen, R. Ranftl, and T. Pock. Insights into analysis operator learning: From patch-based sparse models to higher order mrfs. *IEEE Transactions on Image Processing*, 23(3):1060–1072, 2014.

[6] Y. Chen, W. Yu, and T. Pock. On learning optimized reaction diffusion processes for effective image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[7] A. Danielyan, V. Katkovnik, and K. Egiazarian. Image deblurring by augmented lagrangian with bm3d frame prior. *Workshop on Information Theoretic Methods in Science and Engineering*, 01 2010.

[8] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2016.

[9] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand. Deep joint demosaicking and denoising. *ACM Trans. Graph.*, 35(6):191:1–191:12, 2016.

[10] H. Gouk, E. Frank, B. Pfahringer, and M. Cree. Regularisation of neural networks by enforcing lipschitz continuity. *arXiv preprint arXiv:1804.04368*, 2018.

[11] S. Hawe, M. Kleinsteuber, and K. Diepold. Analysis operator learning and its application to image reconstruction. *IEEE Transactions on Image Processing*, 22(6):2138–2150, 2013.

[12] R. Heckel and P. Hand. Deep decoder: Concise image representations from untrained non-convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[13] F. Heide, M. Steinberger, Y.-T. Tsai, M. Rouf, D. Pająk, D. Reddy, O. Gallo, J. L. abd Wolfgang Heidrich, K. Egiazarian, J. Kautz, and K. Pulli. Flexisp: A flexible camera image processing framework. In *SIGGRAPH Asia*, volume 33(6). ACM, 2014.

[14] E. Kan, J. Min, and J. C. Ye. WaveNet: a deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction. *Medical Physics*, 44:e360–e375, 10 2016.

[15] E. Kobler, T. Klatzer, K. Hammernik, and T. Pock. Variational networks: Connecting variational methods and deep learning. In *German Conference on Pattern Recognition (GCPR)*, 2017.

[16] R. Liu, S. Cheng, X. Liu, L. Ma, X. Fan, and Z. Luo. A bridging framework for model optimization and deep propagation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[17] R. Liu, L. Ma, Y. Wang, and L. Zhang. Learning converged propagations with deep prior ensemble for image enhancement. *IEEE Transactions on Image Processing*, 28(3):1528–1543, 2019.

[18] T. Meinhardt, M. Moeller, C. Hazirbas, and D. Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *International Conference on Computer Vision (ICCV)*, 2017.

[19] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

[20] M. Moeller and D. Cremers. Image denoising—old and new. In M. Bertalmío, editor, *Denoising of Photographic Images and Video: Fundamentals, Open Challenges and New Trends*, pages 63–91. Springer International Publishing, 2018.

[21] M. Möller, T. Möllenhoff, and D. Cremers. Controlling neural networks via energy dissipation. In *ICCV*, 2019.

[22] Y. Romano, M. Elad, and P. Milanfar. The little engine that could: Regularization by denoising (RED). *SIAM Journal on Imaging Sciences*, 10:1804–1844, 2017.

[23] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(205), 2009.

[24] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.

[25] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[26] H. Sedghi, V. Gupta, and P. M. Long. The singular values of convolutional layers. In *ICLR*, 2019.

[27] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman. Plug-and-play priors for bright field electron tomography and sparse interpolation. *IEEE Transactions on Computational Imaging*, 2(4):408–423, Dec 2016.

[28] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia. Scale-recurrent network for deep image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[29] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Deep image prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[30] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg. Plug-and-play priors for model based reconstruction. In *IEEE Global Conference on Signal and Information Processing*, pages 945–948, Dec 2013.

[31] A. Virmaux and K. Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems*, pages 3835–3844, 2018.

[32] L. Xu, J. S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[33] G. Yang, S. Yu, H. Dong, G. Slabaugh, P. L. Dragotti, X. Ye, F. Liu, S. Arridge, J. Keegan, Y. Guo, and D. Firmin. DAGAN: Deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction. *IEEE Transactions on Medical Imaging*, 37(6):1310–1321, 2018.

[34] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[35] 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 (CVPR)*, 2017.

[36] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *International Conference on Computer Vision (ICCV)*, 2015.

[37] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *International Conference on Computer Vision (ICCV)*, pages 479–486, 2011.