IMAGE DENOISING WITH DEEP UNFOLDING AND NORMALIZING FLOWS

Xinyi Wei[1] *Hans van Gorp*[1] *Lizeth Gonzalez Carabarin*[1]
Daniel Freedman[2] *Yonina C. Eldar*[3] *Ruud J.G. van Sloun*[1]

[1] Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands
[2]Verily Research, Tel Aviv, Israel
[3]Department of Math and Computer Science, Weizmann Institute of Science, Rehovot, Israel

ABSTRACT

Many application domains, spanning from low-level computer vision to medical imaging, require high-fidelity images from noisy measurements. State-of-the-art methods for solving denoising problems combine deep learning with iterative model-based solvers, a concept known as deep algorithm unfolding or unrolling. By combining a-priori knowledge of the forward measurement model with learned proximal image-to-image mappings based on deep networks, these methods yield solutions that are both physically feasible (data-consistent) and perceptually plausible (consistent with prior belief). However, current proximal mappings based on (predominantly convolutional) neural networks only implicitly learn such image priors. In this paper, we propose to make these image priors fully explicit by embedding deep generative models in the form of normalizing flows within the unfolded proximal gradient algorithm, and training the entire algorithm in an end-to-end fashion. We demonstrate that the proposed method outperforms competitive baselines on image denoising.

***Index Terms*—** image denoising, inverse problems, deep unfolding, generative modeling, normalizing flows

1. INTRODUCTION

Image recovery from noisy measurements is an important problem in applications spanning from medical imaging[1] to photography[2]. Denoising can be posed as a linear inverse problem with many potential solutions satisfying the measurements. Recovery of a meaningful and plausible solution thus requires adequate statistical priors. Formulating such priors for natural or medical image recovery tasks is however not trivial.

Traditionally, one might approach this problem from a compressed sensing point of view [3, 4], where the signal image is assumed to be sparse in some transform domain. However, choosing the appropriate sparse domain is highly dependent on the application and requires careful analysis of, e.g., wavelet or total variation-based regularizers that are hard to tune in practice.

Deep learning[5] methods are increasingly being adopted as alternatives to compressed sensing in image denoising [6, 7, 8, 9], but also in deconvolution [10], inpainting [11, 12] and end-to-end signal recovery [13, 14]. Moreover, recent works have shown that, using variable splitting techniques [15], any preferred denoiser can be used within (plugged into) classical model-based optimization methods (so called "plug-and-play" approaches). Permitting the use of architectures based on convolutional autoencoders, U-Nets [16], or residual networks (ResNets) [17].

Deep generative models (DGMs), such as generative adversarial networks (GANs) [18], variational autoencoders (VAEs) [19] and normalizing flows [20], can also be used as priors for inverse problems [21, 22, 23, 24, 25]. Such DGMs are pre-trained on large datasets of clean images, learning to map simple (often Gaussian) latent distributions into the complex distribution of images. After pre-training, DGMs are then used to solve inverse problems by performing gradient-based optimization in their simpler latent space.

While all of these approaches improve upon the hand-crafted sparsity-based priors and exhibit great empirical success, they do not accelerate the optimization process and still rely on time-consuming iterative algorithms. Moreover, their strength, being agnostic of the task and merely concerned with modeling the general image prior, is also a limitation: these approaches do not exploit task-specific statistical properties that can aid the optimization.

Deep algorithm unfolding [26, 27, 28] aims to address these problems by unrolling the iterative optimization algorithm as a feed forward deep neural network. The result is a deep network that takes the structure of the iterations in proximal-gradient methods, but allows for learning the parameters and/or successive "neural" proximal mappings directly from training data.

Deep algorithm unfolding is a powerful technique; nevertheless, from an analytic point of view, these learned proximal operators are lacking in a certain regard. In particular, although these models perform excellent representation learning, there is no analytic form to express or even approximate what is actually learned. This fact motivated the current work: we set

out to determine whether a more analytically rigorous formulation of denoising, combined with deep unrolling, could yield a more effective algorithm.

In this paper, we propose an end-to-end deep algorithm unfolding framework that combines neural proximal gradient descent with generative normalizing flow priors. Our approach first pre-trains a generic flow-based model on natural images by direct likelihood maximization, and subsequently fine-tunes the entire pipeline and priors to adapt to specific image reconstruction tasks, in our case image denoising. Our main contributions are as follows:

- We propose a new framework for solving image denoising problems based on deep algorithm unfolding and pre-trained normalizing flows priors that adapt to the data.
- We leverage the generative probabilistic nature of our model to yield a strong initial guess: the maximum likelihood solution of the learned flow prior.
- We demonstrate a superior performance in comparison with the state-of-the-art neural proximal gradient descent baselines.

The remainder of this paper is organized as follows. In section 2, we first introduce the image denoising problem, then we present our method of learning the prior over possible images using normalizing flows, after which we will show our unrolled proximal gradient scheme that uses the normalizing flows prior as a prox. In section 3, we list the experimental setup and the obtained results. We end the paper with a conclusion and discussion of possible future works in section 4.

2. METHODS

2.1. Image denoising

Image denoising can be cast in the following form:

$$\mathrm{y} = \mathrm{x} + \eta, \tag{1}$$

where y is the noisy observed image and x is the desired image, both expressed in vector form, and η is an additive white Gaussian noise (AWGN) vector with zero mean and standard deviation σ_n. The ultimate goal is to remove noise while preserving all the image characteristics (adhering to data consistency). To this end, we employ maximum *a-posteriori* (MAP) estimation:

$$\hat{\mathrm{x}}_{MAP} \coloneqq \arg\max_x p(\mathrm{x}|\mathrm{y}) \propto \arg\max_x p(\mathrm{y}|\mathrm{x})\, p_\theta(\mathrm{x}), \tag{2}$$

where $p(\mathrm{y}|\mathrm{x})$ is the likelihood according to the observed image, and $p_\theta(\mathrm{x})$ is the image prior. Since noise follows a Gaussian distribution with zero mean, we have $p(\mathrm{y}|\mathrm{x}) \sim \mathcal{N}(\mu = \mathrm{x}, \sigma_n^2)$. MAP optimization leads to the following (negative log posterior) minimization problem:

$$\hat{\mathrm{x}} = \arg\min_{\mathrm{x}} \frac{1}{2\sigma_n^2}\|\mathrm{y} - \mathrm{x}\|_2^2 - \log p_\theta(\mathrm{x}). \tag{3}$$

We solve (3) in the following manner. First, we shall use normalizing flows to learn an adequate prior $p_\theta(\mathrm{x})$ over possible images. Second, we employ deep algorithm unfolding to accelerate and improve upon standard gradient descent or quasi-newton based methods for (3) while at the same time improve the learned prior throughout the iterations.

2.2. Normalizing flows priors

Normalizing flows [20] are generative models, that transform a base probability distribution $p(\mathrm{z}) \sim \mathcal{N}(0, I)$ into a more complex, possibly multi-modal distribution by a series of composable, bijective, and differentiable mappings. Normalizing flows can operate in two directions: the generative direction which transforms a point in the (Guassian) latent space into the more complex image space ($\mathrm{x} = g_\theta(\mathrm{z})$), and the flow direction, which maps images to the latent space ($\mathrm{z} = f_\theta(\mathrm{x})$). To create a normalizing flow of sufficient capacity, many layers of bijective functions can be composed together:

$$\mathrm{z} = f_\theta(\mathrm{x}) = (\mathrm{f}_1 \circ \mathrm{f}_2 \circ ... \circ \mathrm{f}_\mathrm{i})(\mathrm{x}), \tag{4}$$

where '$\circ$' denotes the composition of two functions, and θ are the parameters of the model.

Exact density evaluation of $p_\theta(\mathrm{x})$ is possible through the use of the change of variables formula, leading to:

$$\log p_\theta(\mathrm{x}) = \log p(\mathrm{z}) + \log \mid \mathtt{det} D f_\theta(\mathrm{x}) \mid, \tag{5}$$

where D is the Jacobian. The determinant of the Jacobian is added here as we are working with probability density functions, and we thus need to account for the change in density caused by the transformation f_θ. Because we choose z to follow a Gaussian distribution with zero mean and unity variance, and the fact that $\mathrm{x} = g_\theta(\mathrm{z})$, we can then perform proximal update in z space:

$$\begin{aligned} \hat{\mathrm{z}} &= \arg\min_{\mathrm{z}} \frac{1}{2\sigma_n^2}\|\mathrm{y} - g_\theta(\mathrm{z})\|_2^2 - \log p(\mathrm{z}), \\ &= \arg\min_{\mathrm{z}} \|\mathrm{y} - g_\theta(\mathrm{z})\|_2^2 + \lambda\|\mathrm{z}\|_2^2, \end{aligned} \tag{6}$$

where λ is a parameter that balances the importance of adhering to the measurements (data consistency) and the prior. Note that we can also choose other distributions for z, e.g., a Laplace distribution, in which case the ℓ_2 norm becomes an ℓ_1 norm.

In the rest of this paper we employ GLOW as our choice for the normalizing flows prior. For details regarding its architecture and implementations (coding) we refer the reader to the original paper by Kingma and Dhariwal [20] as well as the paper by Asim et al.[25].

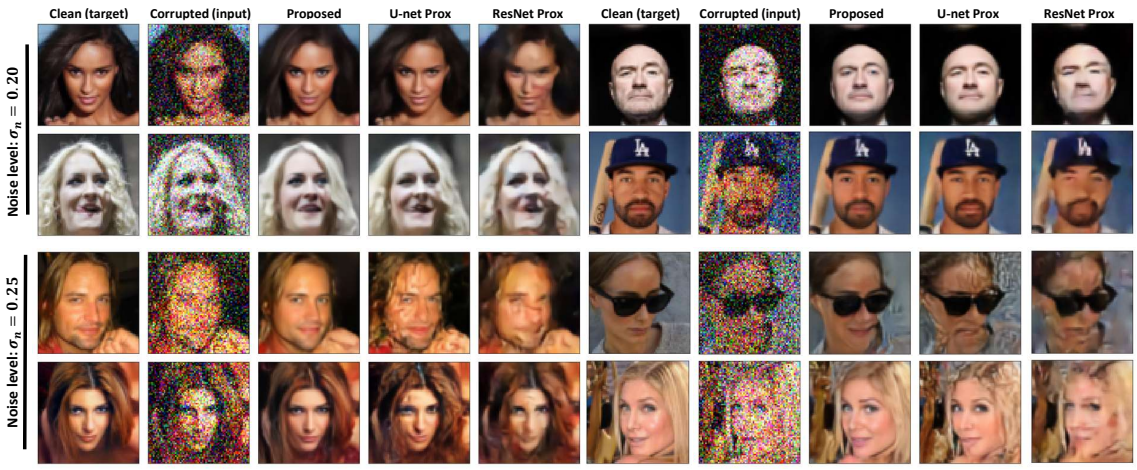

Fig. 1. Comparison of the proposed flow-based proximal mapping with baselines based on standard U-net or ResNet proximal mappings at noise levels $\sigma_n = 0.20$ (in-distribution) and $\sigma_n = 0.25$ (out-of-distribution).

2.3. Unrolled proximal gradient iterations

To solve the optimization problem in (6) we make use of an iterative proximal-style algorithm that alternates between gradient updates in the direction of the data consistency term and pushing the solution in the proximity of the prior.

To derive our iterative scheme, we will alternate between solving in x-space and z-space. Firstly, we will perform a data consistency step in x-space:

$$\tilde{\mathrm{x}}^{(k+1)} = \mathrm{x}^{(k)} - \mu^{(k)}(\mathrm{y} - \mathrm{x}^{(k)}), \tag{7}$$

where superscript (k) denotes the current fold and $\mu^{(k)}$ is the trainable step size. The image is then converted to the latent space using:

$$\tilde{\mathrm{z}}^{(k+1)} = f_\theta^{(k+1)}(\tilde{\mathrm{x}}^{(k+1)}). \tag{8}$$

The purpose of this conversion to latent space is so that we may perform the proximal update $\mathcal{P}(\cdot)$ using the z-space formulation in (6):

$$\mathrm{z}^{(k+1)} = \mathcal{P}^{(k+1)}(\tilde{\mathrm{z}}^{(k+1)}) = \frac{\tilde{\mathrm{z}}^{(k+1)}}{1 + \lambda^{(k+1)}}, \tag{9}$$

where $\lambda^{(k+1)}$ is a trainable shrinkage parameter. Intuitively, this can be understood as pushing solutions into a high likelihood regime (i.e. closer to the origin in z). Finally, we convert from latent space back to signal space

$$\mathrm{x}^{(k+1)} = g_\theta^{(k+1)}(\mathrm{z}^{(k+1)}), \tag{10}$$

and then continue on to the next iteration.

This iterative algorithm is unfolded as a K-fold feedforward neural network that is trained in an end-to-end fashion. After K folds the final estimate $\hat{\mathrm{x}}$ is produced from the latent space after data consistency:

$$\hat{\mathrm{x}} = g_\theta^{(K)}(\tilde{\mathrm{z}}^{(K)}). \tag{11}$$

2.4. Pre-training and initial guess

To aid with the stability during end-to-end training we first pre-train GLOW on a set of clean images to learn a generic density function using 5. After pre-training, we embed these generic priors into the unrolled architecture, and untie their parameters. By then training the model using end-to-end supervised learning, we allow the GLOW model at each fold to be distinct and adapt to the denoising task. Moreover, we use the pre-trained GLOW model to yield a powerful initial guess for $\mathrm{x}^{(0)}$. As we know that the most likely image lives at the origin of the Gaussian latent space, we set $\mathrm{x}^{(0)} = g_\theta^{(0)}(\mathrm{z}^{(0)} = 0)$.

3. EXPERIMENTS

We assess our framework's performance for the denoising task on images of human faces, using the CelebA-HQ dataset[29] that consists of 27,000 training, 1,500 validation, and 1,500 test images. The images are resized to 64×64 pixels with 3 (RGB) channels. We manually corrupt these images using AWGN and train our unfolded proximal gradient network for $k = 4$ folds using a Mean Square Error (MSE) loss.

We employ the Adam optimizer with (lr = $1e^{-5}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e-8$). Moreover, we train the learnable step size $\mu^{(k)}$, and shrinkage factor $\lambda^{(k)}$ with a higher learning rate, namely $1e^{-2}$. Early stopping is used if the validation loss does not decrease for 5 consecutive epochs. Leveraging the invertible nature of the GLOW model, we strongly reduce train-time memory of the full unfolded architecture using the approach by Putzky and Welling [30]; instead of storing all intermediate activations for back-propagation, we recalculate them during backpropagation.

We compare our normalizing flows prior, to two alternative neural proximal mappings; one based on ResNet [26], and one

Table 1. Denoising results of deep unfolding with flow-based proximal mapping compared to two strong baselines. Values reported are mean Peak Signal to Noise Ratio (PSNR) of the reconstructed images across the CelebA-HQ[29] test set.

Denoising	**ResNet Prox** [26]	**U-Net Prox** [16]	**Glow Prox (ours)**
$\eta \sim \mathcal{N}(\mu_n = 0,\, \sigma_n = 0.10)$	28.567 dB	29.228 dB	**29.423** dB
$\eta \sim \mathcal{N}(\mu_n = 0,\, \sigma_n = 0.15)$	28.029 dB	28.920 dB	**29.009** dB
$\eta \sim \mathcal{N}(\mu_n = 0,\, \sigma_n = 0.20)$	27.054 dB	28.180 dB	**28.236** dB
$\eta \sim \mathcal{N}(\mu_n = 0,\, \sigma_n = 0.25)$	25.489 dB	25.770 dB	**26.633** dB

based on a U-net. Note that for a fair and direct comparison, we focus on typical alternatives *within* the unfolded proximal gradient framework. This allows a straightforward assessment of the proposed (task-adapted) normalizing flows priors beyond the architectural advantages of unfolding the proximal gradient algorithm itself.

The ResNet proximal baseline follows the structure proposed by [26]. Each residual block consists of two convolutional layers with 3×3 kernels and 128 feature maps, followed by batch normalization and ReLU activations. These were followed by three convolutional layers with 1×1 kernels, where the first two made use of ReLU activations. The second proximal baseline is a standard U-net [16]. The U-net is a convolutional neural network that follows a typical encoding-decoding architecture, with extra skip connections between each input and output at every encoding level. Here we make use of a Pytorch U-Net implementation.

We trained all three proximal methods, based on ResNet, U-Net, and the proposed GLOW prox, for AWGN with a standard deviation of $\sigma_n = 0.20$. We then analyzed performance for four different standard deviations, ranging from $\sigma_n = 0.10$ to $\sigma_n = 0.25$, (see Table 1). Our proposed method outperforms the baselines not only on the in-distribution noise levels, but also on the out-of-distribution noise levels. Qualitatively, this also becomes apparent from the examples displayed in Fig. 1. The reconstructions when using our GLOW prox are sharper, and details (for example, the hair) are better preserved.

4. CONCLUSION

In this paper, we proposed an unfolded neural proximal gradient descent framework with a normalizing flow prior for image denoising. We demonstrated that our proposed framework outperforms the two strong baselines on both in-distribution and out-of-distribution noise levels. While unfolding and end-to-end training enables fitting to (and exploiting) a specific data distribution, it also makes it more sensitive to out of distribution measurements. We show that generative flow proximal operators suffer less from this problem than standard discriminative U-Net or ResNet ones, and thus have advantages in real world applications of unfolding. This does not mean that these denoisers work poorly when used in a plug-and-play setting.

In this manuscript we only explored the use of our method on image denoising. However, multiple challenges can be cast in the same way as equation 1 by adding a measurement matrix $A \neq I$, e.g., inpainting or deblurring. Future work would include experiments on these other types of problems. Moreover, future work could also include analysis on out-of-distribution performance and the impact of pre-training the GLOW prior.

5. REFERENCES

[1] Nicola Pezzotti et al., "An adaptive intelligence algorithm for undersampled knee mri reconstruction: Application to the 2019 fastmri challenge," *arXiv preprint arXiv:2004.07339*, 2020.

[2] Francisco Duarte Moura Neto and Antônio José da Silva Neto, *An introduction to inverse problems with applications*, Springer Science & Business Media, 2012.

[3] David L Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[4] Yonina C Eldar and Gitta Kutyniok, *Compressed sensing: theory and applications*, Cambridge university press, 2012.

[5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[6] H.C. Burger, C.J. Schuler, and Stefan Harmeling, "Image denoising: Can plain neural networks compete with bm3d?," 06 2012, pp. 2392–2399.

[7] Junyuan Xie, Linli Xu, and Enhong Chen, "Image denoising and inpainting with deep neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, Red Hook, NY, USA, 2012, NIPS'12, p. 341–349, Curran Associates Inc.

[8] Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong xu, Wangmeng Zuo, and Chia-Wen Lin, "Deep learning on

image denoising: An overview," *Neural Networks*, vol. 131, 08 2020.

[9] V.Jain and H.S.Seung, "Natural image denoising with convolutional networks," 2008, pp. 769–776.

[10] Li Xu, Jimmy S. J. Ren, Ce Liu, and Jiaya Jia, "Deep convolutional neural network for image deconvolution," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, Cambridge, MA, USA, 2014, NIPS'14, p. 1790–1798, MIT Press.

[11] Ugur Demir and Gözde B. Ünal, "Patch-based image inpainting with generative adversarial networks," *CoRR*, vol. abs/1803.07422, 2018.

[12] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z. Qureshi, and Mehran Ebrahimi, "Edgeconnect: Generative image inpainting with adversarial edge learning," *CoRR*, vol. abs/1901.00212, 2019.

[13] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "Reconnet: Non-iterative reconstruction of images from compressively sensed measurements," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 449–458.

[14] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, 2017.

[15] 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 in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[16] 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*. Springer, 2015, pp. 234–241.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[18] Ian J.Goodfellow, J.Pouget-Abadie, M.Mirza, B.Xu, D.Warde-Farley, S.Ozair, A.Courville, and Y.Bengio, Eds., *Generative Adversarial Nets*, Curran Associates, Inc., 2014.

[19] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," 2014.

[20] Durk P Kingma and Prafulla Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. 2018, vol. 31, Curran Associates, Inc.

[21] P.Hand and V.Voroninski O.Leong, "Phase retrieval under a generative prior," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2018, NIPS'18, p. 9154–9164, Curran Associates Inc.

[22] P.Hand and V.Voroninski, "Global guarantees for enforcing deep generative priors by empirical risk," *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 401–418, 2020.

[23] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G. Dimakis, "Compressed sensing using generative models," 2017.

[24] Muhammad Asim, Fahad Shamshad, and Ali Ahmed, "Solving bilinear inverse problems using deep generative priors," *CoRR*, vol. abs/1802.04073, 2018.

[25] Muhammad Asim, Max Daniels, Oscar Leong, Paul Hand, and Ali Ahmed, "Invertible generative models for inverse problems: mitigating representation error and dataset bias," in *Proceedings of Machine Learning and Systems 2020*, pp. 4577–4587. 2020.

[26] M.Mardani, Q.Y.Sun, S.Vasawanala, V.Papyan, H.Monajemi, J.Pauly, and D.Donoho, "Neural proximal gradient descent for compressive imaging," *CoRR*, vol. abs/1806.03963, Feb 2018.

[27] Vishal Monga, Yuelong Li, and Yonina C Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.

[28] Yuelong Li, Mohammad Tofighi, Junyi Geng, Vishal Monga, and Yonina C Eldar, "Deep algorithm unrolling for blind image deblurring," *arXiv preprint arXiv:1902.03493*, 2019.

[29] T.Karras, T.Aila, S.Laine, and J.Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *ICLR 2018*, vol. abs/1710.10196, Feb 2018.

[30] Patrick Putzky and Max Welling, "Invert to learn to invert," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., pp. 444–454. Curran Associates, Inc., 2019.