MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Learning Plug-and-Play Proximal Quasi-Newton Denoisers

Al-Shabili, Abdullah; Mansour, Hassan; Boufounos, Petros T.

TR2020-045 April 11, 2020

Abstract

Plug-and-play (PnP) denoising for solving inverse problems has received significant attention recently thanks to its state of the art signal reconstruction performance. However, the performance improvement hinges on carefully choosing the noise level of the Gaussian denoiser and the descent step size in every iteration. We propose a strategy for training a Gaussian denoiser inspired by an unfolded proximal quasi-Newton algorithm, where the noise level of the input signal to the denoiser is estimated in each iteration and at every entry in the signal. Our scheme deploys a small convolutional neural network (mini-CNN) to estimate an element-wise noise level, mimicking a diagonal approximation of the Hessian matrix in quasi-Newton methods. Empirical simulation results on image deblurring demonstrate that our proposed approach achieves approximately 1dB improvement over state of the art methods, such as, BM3D-PnP and proximal gradient descent-PnP that are supplied with the true noise level, as well as over an end-to-end retrained FFDNet architecture that was trained to estimate the noise level and recover the deblurred images.

IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)

LEARNING PLUG-AND-PLAY PROXIMAL QUASI-NEWTON DENOISERS

Abdullah H. Al-Shabili[*]

Tandon School of Engineering
New York University
Brooklyn, NY 11209
abdullah.alshabili@nyu.edu

Hassan Mansour, Petros T. Boufounos

Mitsubishi Electric Research Laboratories
Cambridge, MA 02139
{mansour, petrosb}@merl.com

ABSTRACT

Plug-and-play (PnP) denoising for solving inverse problems has received significant attention recently thanks to its state of the art signal reconstruction performance. However, the performance improvement hinges on carefully choosing the noise level of the Gaussian denoiser and the descent step size in every iteration. We propose a strategy for training a Gaussian denoiser inspired by an unfolded proximal quasi-Newton algorithm, where the noise level of the input signal to the denoiser is estimated in each iteration and at every entry in the signal. Our scheme deploys a small convolutional neural network (mini-CNN) to estimate an element-wise noise level, mimicking a diagonal approximation of the Hessian matrix in quasi-Newton methods. Empirical simulation results on image deblurring demonstrate that our proposed approach achieves approximately 1dB improvement over state of the art methods, such as, BM3D-PnP and proximal gradient descent-PnP that are supplied with the true noise level, as well as over an end-to-end retrained FFDNet architecture that was trained to estimate the noise level and recover the deblurred images.

***Index Terms*—** Inverse problems, deep learning, plug-and-play, proximal methods, quasi-Newton.

1. INTRODUCTION

Recent work has demonstrated that Deep learning can be very effective in solving linear inverse problems encountered in signal processing applications [1–3]. This success is due to the ability of deep network architectures to provide nonlinear and differentiable models for classes of signals that are not well-characterized using conventional signal models, such as subspace, manifold, or sparse models.

A linear inverse problem can be described by the linear system of equations

$$y = Ax + v,$$

in which a target signal (or image) $x \in \mathbb{R}^N$ is observed through a linear and often under-determined forward operator $A \in \mathbb{R}^{m \times N}$ to produce the measurements $y \in \mathbb{R}^m$. The measurement noise, $v \in \mathbb{R}^m$, is often assumed to be independent Gaussian distributed. Given the measurements y and observation matrix A, the task is to recover the target signal x by incorporating some prior knowledge on the structure of the class of signals being acquired. Consequently, established frameworks for tackling such problems set up the following optimization problem:

$$\min_x f(x) + \lambda \rho(x), \tag{1}$$

where $f(x) := \frac{1}{2}\|y - Ax\|_2^2$ is the objective function, $\rho(x)$ is regularizing penalty function—typically a non-smooth—that restricts the space of solutions to the appropriate class of signals x, and λ is the regularization parameter controlling the trade-off. Traditional signal processing methods explicitly model the regularizer $\rho(x)$ and have derive efficient algorithms for solving (1). One of the most computationally efficient and effective iterative methods is the proximal gradient descent (PGD) method [4, 5], that splits the update in each iteration t into the following two steps:

$$z^{t+1} = x^t - \alpha^t A^T (Ax - y) \tag{2a}$$

$$x^{t+1} = \arg\min_u \frac{1}{2} \left\|u - z^{t+1}\right\|_2^2 + \alpha^t \lambda \rho(u) \tag{2b}$$

The first step is a gradient descent update with respect to the objective function $f(\cdot)$ using the step size α^t, while the second step computes a proximal mapping with respect to the penalty function.

However, in some applications, the penalty function $\rho(\cdot)$ may be difficult to determine or describe analytically, but examples of the class of signals x may be readily available. In this context, the Plug-and-Play (PnP) framework [6] replaces the proximal mapping with respect to $\rho(\cdot)$ with a signal denoiser that can be trained from the data. In this framework, the update equations become

$$z^{t+1} = x^t - \alpha^t A^T (Ax - y) \tag{3a}$$

$$x^{t+1} = \mathcal{D}\left(z^{t+1}\right) \tag{3b}$$

where $\mathcal{D}$ denotes a denoising deep neural network (DNN) that plays the role of a proximal operator. This approach resembles the classical PGD, but lacks a regularization parameter that is tightly related to the noise level in the observations. In other words, once the denoiser is trained on a certain noise level, it becomes a black box, and we cannot control the denoiser strength. The resulting algorithm has fundamental generalization issues, that become evident when the observation noise level is outside the range of pre-trained noise levels.

In this paper, we seek to equip the deep learning PnP approach with a mechanism to estimate the noise standard deviation at each of the N entries in x, and subsequently control the denoiser strength. To that end, we adopt the FFDNet [7] architecture for the PnP denoiser which allows us to input the noise standard deviation at every iteration. We then re-train the FFDNet denoiser by unfolding the proximal quasi-Newton (PQN) algorithm over T iterations and introducing a small Convolutional Neural Network (mini-CNN) to estimate the element-wise noise level affecting the iterates x^t at every iteration $t \in \{1 \dots T\}$ in (3a). The proposed framework mimics a diagonal approximation of the Hessian matrix in the PQN algorithm, thus reducing the computation complexity of the algorithm. Moreover, the diagonal approximation provides a direct intuition of the noise standard deviation affecting the elements of x^t.

[*] A. H. Al-Shabili performed this work during an internship at MERL.

Learning deep signal denoisers is a rich research area that has generated state of the art preformance, ranging from CNN architectures [8], Encoder-Decoder [9], Deep Image Prior [10], and many more. Most of these denoisers do not have a mechanism of controlling the denoising after the training phase is finished. Thus, [11] uses the half quadratic splitting algorithm and the DnCNN denoiser in the PnP framework by training a set of denoisers on a wide range of noise levels. This approach faces fundamental issues of practicality as well as the challenges of training and storing multiple denoiser networks. The same issue of controlling the deep learning denoiser strength has been flagged in the literature, e.g., see [12, 13]. Recently, the FFDNet [7] was proposed, which has an input layer that represents the estimation of the standard deviation of the noise level. This feature makes the FFDNet architecture a good candidate PnP denoiser for our problem.

The paper is organized as follows: Section 2 provides a background of the model-based optimization methods and deep learning denoisers in inverse problems. In Section 3, we present the proposed proximal quasi-Newton method. Simulation results are reported in Section 4 and we conclude the paper in Section 5, where we also address some open questions.

2. BACKGROUND

2.1. Model-based Optimization Methods

The classical model based optimization formulates the linear inverse problem as minimizing a composite cost function as

$$\hat{x} = \arg\min_{x} \underbrace{\|y - Ax\|^2}_{f(x)} + \lambda\rho(x) \tag{4}$$

where the first term, $f(\cdot)$, measures the discrepancy between the estimation and the observations, and $\rho(\cdot)$ captures the structure of the signal x.

The Proximal Newton Method (PNM) is a non-smooth extension of the Newton-Raphson method [14]. The method first generates a quadratic approximation of the smooth term at each iteration. Then a non-separable proximal operator is computed as follows

$$z^{t+1} = x^t - \left[\nabla_x^2 f(x^t)\right]^{-1} \nabla f(x^t) \tag{5a}$$

$$x^{t+1} = \arg\min_{x} \frac{1}{2} \left\|x - z^{t+1}\right\|^2_{\nabla_x^2 f(x^t)} + \lambda\rho(x) \tag{5b}$$

in case $\nabla_x^2 f(x^t)$ is replaced with a scaled identity, the PNM reduces to the Proximal Gradient Descent (PGD) method. Aside from inverting the Hessian matrix, the major computational challenge in using PNM is evaluating the non-seperable proximal operator, which depends on both $\rho(\cdot)$ and the structure of the Hessian matrix.

2.2. Deep Neural Networks as Denoisers

The plug-and-play (PnP) method relies on the notion that the proximal mapping is in fact a constrained Gaussian denoiser

$$\mathbf{Prox}_\rho(z, \lambda) := \arg\min_{x} \frac{1}{2\lambda} \|x - z\|^2 + \rho(x). \tag{6}$$

To exploit DNNs, that can capture highly complex signal representations, the Gaussian denoiser problem may be reformulated as

$$\hat{\Theta} = \arg\min_{\Theta} \sum_{i=1}^{M} \|x_i - \mathrm{NN}_\Theta(y_i)\|^2 \tag{7a}$$

$$\hat{x}_{\text{learn}} = \mathrm{NN}^\star_\Theta(y) = \mathcal{D}(y), \tag{7b}$$

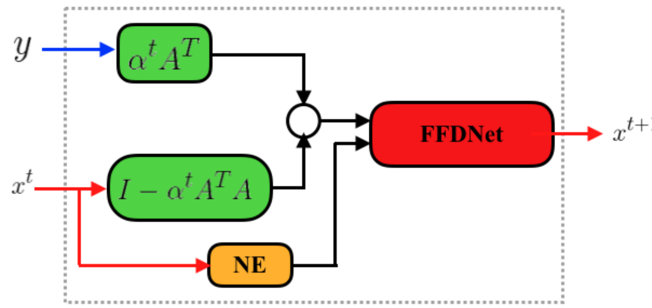

Fig. 1. A single iteration block model of the proposed proximal quasi-Newton method. The core denoiser is based on a retrained FFDNet architecture with a noise estimator (NE) block for estimating a spatially variant input noise standard deviation.

where M is number of training samples, and Θ models the neural network parameters. The reconstructed image $\hat{x}_{\text{learn}}$ is produced at the output of the neural network $\mathrm{NN}^\star_\Theta(y)$ that acts as a denoiser and takes as input the blurry image y. The PnP framework uses this network in place of the proximal mapping in the proximal algorithm. Our goal is to use this framework to obtain a PnP version of the quasi-PNM with a denoiser that is versatile to the noise level.

3. LEARNED PROXIMAL QUASI-NEWTON DENOISER

The challenge in designing a noise-versatile PnP denoiser for proximal quasi-Newton iterations arises when the forward operator A is column-rank deficient. In such cases, the inverse of the Hessian does not exist and every iteration introduces noise that belongs in the null space of A.

Many neural network architectures have been proposed in the literature as Gaussian denoisers of natural images. For example, DnCNN uses 17 layers with a residual connection [8], multi-level wavelet CNN (MWCNN) uses a modified U-Net architecture and wavelet transform concepts [15], and RED-Net uses encoder-decoder with symmetric skip connections [9]. Despite their promising performance, these approaches cannot be applied if the noise distribution and variance differs from that of the training dataset. Instead, the FFDNet [7] architecture adds an input layer that represents the noise standard deviation, thus resolving the lack of explicit dependence on the noise level.

We adopt an algorithm unfolding strategy in order to train a denoiser based on the main FFDNet architecture. In order to handle the spatially varying noise that arises from the contribution of the null space of A, we add a noise estimator (NE) module to the input of the FFDNet denoiser that behaves as a diagonal Hessian estimator $\hat{H}(x^t)$ and determines the element-wise noise standard deviation affecting the entries of x^t. The NE module is composed of a small convolutional neural network (mini-CNN) formed from two 2-D convolution layers with 24 filters in each layer of size 3×3, and a ReLU activation function. The output of the NE module produces the noise standard deviation input layer of the FFDNet denoiser.

By unfolding the proximal quasi-Newton method over T iterations, we produce a learned PQN method (PQNM) where each iter-

Table 1. The average PSNR (dB) and SSIM results on different datasets and methods.

Dataset	Blurred	BM3D (true σ)	PGD (true σ, fine)	PGD (true σ, Gauss)	PQNM (1-iter $\equiv$ E-to-E)	PQNM
Set5	26.05 / 0.78	29.34 / 0.86	**32.73 / 0.90**	30.20 / 0.87	31.52 / 0.89	32.62 / **0.90**
Set14	24.64 / 0.71	27.43 / 0.81	28.69 / 0.84	26.32 / 0.77	29.08 / 0.85	**30.74 / 0.87**
CBSD68	25.01 / 0.71	27.39 / 0.81	30.09 / 0.88	27.42 / 0.81	29.61 / 0.87	**30.92 / 0.88**

ation is composed of the following update equations:

$$z^{t+1} = x^t - \alpha^t \; A^T \left(Ax - y\right) \tag{8a}$$

$$x^{t+1} = \text{FFDNet}\left(z^{t+1}, \lambda\, \alpha^t\, \hat{H}(x^t)\right) \tag{8b}$$

Note that equation (8a) is missing the inverse of the approximate Hessian component $\left[\hat{H}(x^t)\right]^{-1}$. This is due to a numerical instability that we observed during the training phase that prevents the training from converging. A block diagram of one iteration of the proposed PQNM is presented in Fig. 1. The Noise Estimator module represents a mini-CNN that is shared with all the iterations. The task of this network is to process a noisy image and output the estimated noise level in the range of $[0, 1/(\lambda\alpha^t)]$.

We use $\mathcal{T}^T_{H_\Theta}(\cdot)$ to denote the T unfolded iterations of the simplified PQNM-PnP, where H_Θ represents the NE network parameters. Therefore the overall optimization problem can be formulated as

$$\hat{\Theta} = \arg\min_{\Theta} \sum_{i=1}^{M} \left\| x_i - \mathcal{T}^T_{H_\Theta}(y_i) \right\|^2 \tag{9}$$

$$\text{s.t.} \quad 0 \leqslant \text{NN}_{H_\Theta}(y_i) \leqslant \frac{1}{\lambda\alpha^t} \quad \forall i = 1, \cdots, M$$

In order to simplify the training process, the box constraint is embedded into the model architecture by adding a clipping layer.

4. NUMERICAL RESULTS

We evaluate the performance of our learned proximal quasi-Newton denoiser on an image deblurring application where the blurring kernels are known.

4.1. Denoiser Pre-training

The pre-trained FFDNet Gaussian denoiser is obtained from [16], which is trained with $M = 128 \times 8000$ patches from the Waterloo Exploration Database [17]. The FFDNet was trained by adding white Gaussian noise of standard deviation in the range of $[0, 75]/255$. During the pre-training phase, the noise standard deviation input layer is assumed to be known and uniform. During the unfolded training stage, the assumption of a known noise level will be dropped and the mini-CNN will be used to estimate this noise input layer.

4.2. Unfolded training of PQNM-PnP

Once the PQNM-PnP algorithm is unfolded over $T = 10$ iterations, the mini-CNN is trained along with fine tuning of the last 3 layers of the FFDNet. The goal of this fine tuning is to be able to adapt the FFDNet denoiser to handle the changing distribution of the noise that arises from the null space of the forward operator A in each iteration. Using the same procedure and in order to evaluate the performance relative to a comparable end-to-end learning approach, we retrain the PQNM for a single iteration with all the layers retrained except the first 2 convolution layers. In both of the scenarios, the simulation settings are as follows: we extract $100,000$ patches of size 128×128 from the Waterloo Exploration Database. Motion blur kernels are generated by uniformly sampling 6 angles in $[0, \pi]$ and 6 lengths in a range of $[5, 15]$ pixels with 15×15 pixel size kernels. After convolving an image with each kernel, a Gaussian noise of 0.01 variance is added to the blurry images.

The step size parameter of each iteration, α^t, is a learnable parameter. The training is performed using the Adam solver, with an initial learning rate of 10^{-4}. For the validation dataset, we use the Kodak24 dataset [18]. The Adam learning rate is decayed by a factor of 0.1 after two epochs of non-increasing average PSNR of the validation dataset, i.e. when the training reaches a plateau phase. The model is trained for 50 epochs.

4.3. Evaluation

We use different datasets to evaluate the performance of the proposed algorithm, namely CBSD68 [19], Set5, and Set14 [20]. Each image is convolved with 4 kernels with different angles and lengths. We compare the performance of our proposed PQNM-PnP method with several algorithms such as the BM3D-PnP approach. Moreover, we compare with two PGD-PnP versions that lack a noise estimator module. In one of the PGD methods, the Gaussian pre-trained FFDNet denoiser is kept fixed and the true noise standard deviation is given to the denoiser in every iteration. The other method, uses a fine tuned FFDNet denoiser where the last 3 layers are updated along with the true noise standard deviation. Furthermore, We also present the performance of an end-to-end trained FFDNet denoiser network with a mini-CNN to estimate the noise standard deviation using the same image deblurring training dataset to demonstrate the advantage of algorithm unfolding.

Table 1 illustrates the deblurring performance of the above mentioned methods. The comparison is based on the average PSNR and the Structural Similarity Index (SSIM). Despite the fact that our proposed PQNM is blind to the noise level, it achieves the highest PSNR and SSIM in almost all the experiments compared to the PGD-PnP and BM3D-PnP that are supplied with the true noise level. Moreover, the proposed PQNM outperform the single iteration end-to-end learning approach with FFDNet by a consistent 1 dB in the average PSNR value.

Fig 2 illustrates sample results for the proposed approach using a Gaussian blur kernel with standard deviation 1.6, and similar noise level to the previous experiment. Note that Gaussian kernel was not among the fine tuning process, therefore this experiment should indicate the generalization ability of our simple fine tuning scheme. The proposed approach outperforms both the BM3D-PnP and PGD-PnP algorithms that use the true noise level. This could be attributed

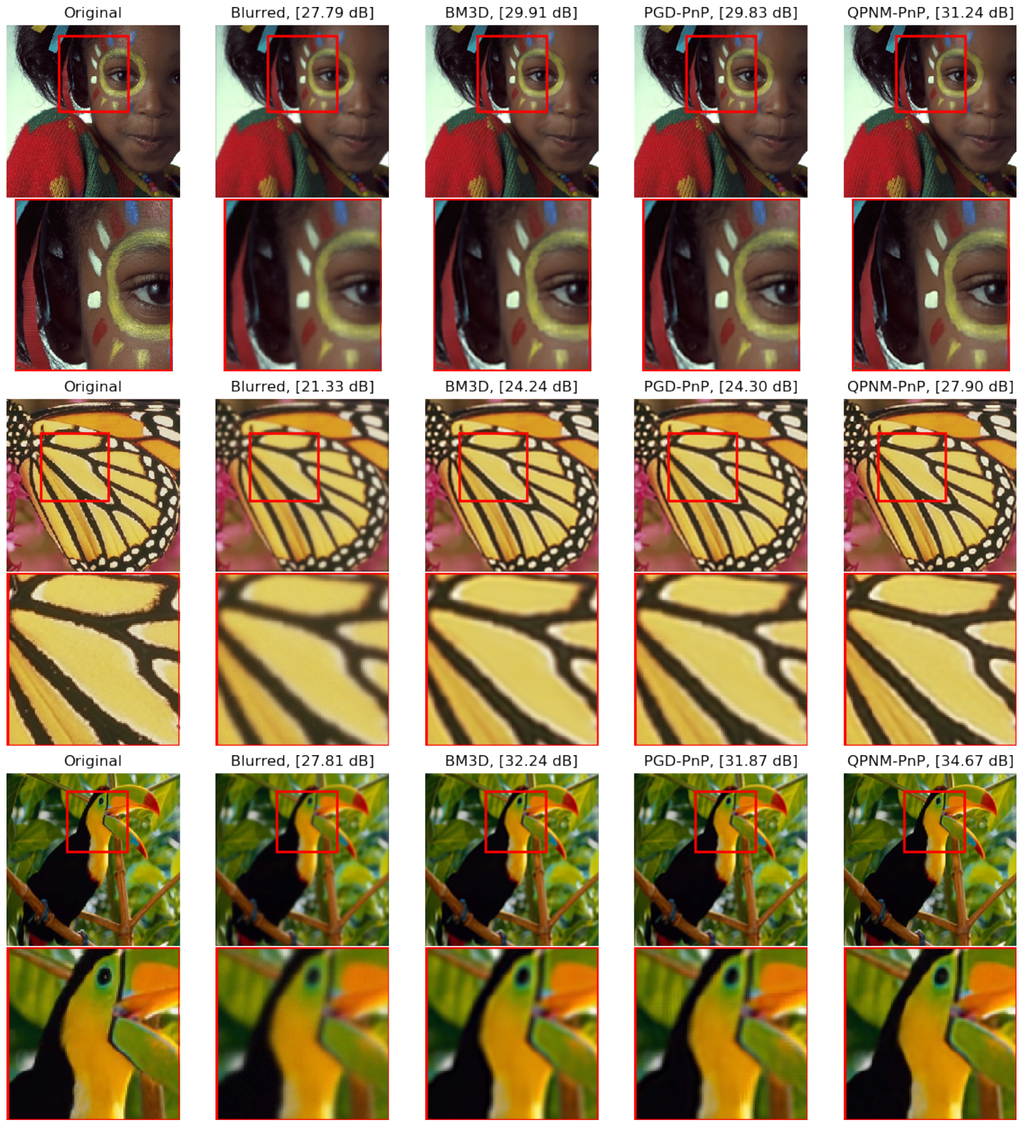

Fig. 2. Image deblurring performance(PSNR) of the proposed PQNM-PnP, using the common Gaussian blur with standard deviation 1.6.

to the flexibility of the mini-CNN that is capable of accounting for spatially-variant noise level that arises during the iterative framework.

5. CONCLUSION

The integration between deep learned denoisers and the Plug-and-play (PnP) approach has shown promising state-of-the-art results. Nonetheless, deep learned denoisers fundamentally suffer from a lack of a mechanism to control the denoising strength. In this work, we proposed a diagonal approximation of the Proximal Quasi Newton Method (PQNM), which is implemented via a mini-CNN. This approximation can be interpreted as estimating the noise standard deviation from a statistical perspective. The unfolding of the PQNM faces numerical challenges with inverting the estimated Hessian matrix during the learning process. Despite these challenges, the proposed approach delivers promising simulation results that encourages further investigation. In summary, we propose an approach that can control the strength of the deep learned denoiser by estimating the noise levels across multiple iterations of an unfolded algorithm.

6. REFERENCES

[1] Alice Lucas, Michael Iliadis, Rafael Molina, and Aggelos K Katsaggelos, "Using deep neural networks for inverse problems in imaging: beyond analytical methods," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 20–36, 2018.

[2] Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, 2017.

[3] Yaniv Romano, Michael Elad, and Peyman Milanfar, "The little engine that could: Regularization by denoising (red)," *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1804–1844, 2017.

[4] Patrick L Combettes and Jean-Christophe Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*, pp. 185–212. Springer, 2011.

[5] Neal Parikh, Stephen Boyd, et al., "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[6] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg, "Plug-and-play priors for model based reconstruction," in *2013 IEEE Global Conference on Signal and Information Processing*. IEEE, 2013, pp. 945–948.

[7] Kai Zhang, Wangmeng Zuo, and Lei Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, 2018.

[8] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[9] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Advances in neural information processing systems*, 2016, pp. 2802–2810.

[10] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Deep image prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.

[11] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang, "Learning deep cnn denoiser prior for image restoration," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3929–3938.

[12] JH Rick Chang, Chun-Liang Li, Barnabas Poczos, BVK Vijaya Kumar, and Aswin C Sankaranarayanan, "One network to solve them all–solving linear inverse problems using deep projection models," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5888–5897.

[13] Tim Meinhardt, Michael Moller, Caner Hazirbas, and Daniel Cremers, "Learning proximal operators: Using denoising networks for regularizing inverse imaging problems," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1781–1790.

[14] Jason D Lee, Yuekai Sun, and Michael A Saunders, "Proximal newton-type methods for minimizing composite functions," *SIAM Journal on Optimization*, vol. 24, no. 3, pp. 1420–1443, 2014.

[15] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo, "Multi-level wavelet-cnn for image restoration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 773–782.

[16] Matias Tassano, Julie Delon, and Thomas Veit, "An analysis and implementation of the ffdnet image denoising method," *Image Processing On Line*, vol. 9, pp. 1–25, 2019.

[17] Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang, "Waterloo exploration database: New challenges for image quality assessment models," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 1004–1016, 2016.

[18] Eastman Kodak, "Kodak lossless true color image suite (photocd pcd0992)," *URL http://r0k. us/graphics/kodak*, 1993.

[19] David Martin, Charless Fowlkes, Doron Tal, Jitendra Malik, et al., "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *ICCV*. Iccv Vancouver:, 2001.

[20] Radu Timofte, Vincent De Smet, and Luc Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Asian conference on computer vision*. Springer, 2014, pp. 111–126.