



**HAL**  
open science

# Learned Image Deblurring by Unfolding a Proximal Interior Point Algorithm

Marie-Caroline Corbineau, Carla Bertocchi, Emilie Chouzenoux, Marco Prato,  
Jean-Christophe Pesquet

► **To cite this version:**

Marie-Caroline Corbineau, Carla Bertocchi, Emilie Chouzenoux, Marco Prato, Jean-Christophe Pesquet. Learned Image Deblurring by Unfolding a Proximal Interior Point Algorithm. ICIP 2019 - 26th IEEE International Conference on Image Processing, IEEE, Sep 2019, Taipei, Taiwan. 10.1109/ICIP.2019.8803438 . hal-02303511

**HAL Id: hal-02303511**

**<https://hal.science/hal-02303511v1>**

Submitted on 2 Oct 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# LEARNED IMAGE DEBLURRING BY UNFOLDING A PROXIMAL INTERIOR POINT ALGORITHM

M.-C. Corbineau<sup>1</sup>, C. Bertocchi<sup>2</sup>, E. Chouzenoux<sup>1</sup>, M. Prato<sup>2</sup>, J.-C. Pesquet<sup>1</sup>

<sup>1</sup> CVN, CentraleSupélec, INRIA Saclay, University of Paris-Saclay, Gif-Sur-Yvette, France

<sup>2</sup> Università di Modena e Reggio Emilia, Modena, Italy

<sup>1</sup>first.last@centralesupelec.fr, <sup>2</sup>first.last@unimore.it

## ABSTRACT

Image restoration is frequently addressed by resorting to variational methods which account for some prior knowledge about the solution. The success of these methods, however, heavily depends on the estimation of a set of hyperparameters. Deep learning architectures are, on the contrary, very generic and efficient, but they offer limited control over their output. In this paper, we present iRestNet, a neural network architecture which combines the benefits of both approaches. iRestNet is obtained by unfolding a proximal interior point algorithm. This enables enforcing hard constraints on the pixel range of the restored image thanks to a logarithmic barrier strategy, without requiring any parameter setting. Explicit expressions for the involved proximity operator, and its differential, are derived, which allows training iRestNet with gradient descent and backpropagation. Numerical experiments on image deblurring show that the proposed approach provides good image quality results compared to state-of-the-art variational and machine learning methods.

**Index Terms**— Image deconvolution, proximity operator, interior point method, neural network, deep unfolding

## 1. INTRODUCTION

In this paper we focus on non-blind image deblurring problems matching the following model

$$y = H\bar{x} + \omega, \quad (1)$$

where  $\bar{x} \in \mathbb{R}^n$  is the ground-truth vectorized color image,  $y \in \mathbb{R}^n$  is a blurred noisy version of it,  $\omega \in \mathbb{R}^n$  is a realization of an additive zero-mean white Gaussian noise with standard deviation  $\sigma > 0$ , and  $H \in \mathbb{R}^{n \times n}$  models the circular convolution with a known blur kernel for each channel of the color image. Finding an appropriate estimate of  $\bar{x}$  is an ill-posed problem. An efficient way to tackle it is to adopt a penalized variational approach introducing prior knowledge about the sought image. For instance, one can impose bound constraints on the pixel intensity range, or use a spatial regularization function, such as the total variation and its various

extensions [1]. This leads to the following constrained minimization problem,

$$\underset{x \in [0, \beta]^n}{\text{minimize}} \frac{1}{2} \|Hx - y\|^2 + \lambda \mathcal{R}(x) \quad (2)$$

where  $\beta > 0$  is an upper bound on the pixel intensity,  $\lambda > 0$  is a regularization parameter, and  $\mathcal{R} : \mathbb{R}^n \rightarrow \mathbb{R}$  is a penalization function. However, solving (2) may require advanced algorithms, which can be too slow for real-time applications. Moreover, the optimal choice for  $\lambda$  strongly depends on the data, and is usually obtained using time-consuming techniques such as cross validation, discrepancy principle [2], or Stein unbiased risk estimates (SURE) [3].

Deep Neural Networks (DNNs), in particular Convolutional Neural Networks (CNNs), provide good performance for restoration problems such as denoising [4], CT reconstruction [5], or deblurring [6, 7, 8]. However, as detailed in [9], DNNs are very often preceded by a pre-processing step, which again amounts to solving a problem of the form (2), where the setting of the regularization weight requires an accurate knowledge of the noise level, *e.g.* [7, 10]. Furthermore, since explicit prior information about the output can hardly be incorporated into a DNN, the explainability and reliability of such methods could be questioned [11].

One straightforward way to combine the benefits of variational approaches and DNNs is to unfold an iterative method and to untie the parameters of both the model and the algorithm across the layers of the network [12]. For instance, FISTA, ISTA and ADMM optimizers can be unfolded to perform sparse coding [13] and image reconstruction [14, 15, 16], respectively. However, in the aforementioned works, all operators are learned, which weakens the link between the resulting network and the original algorithm. Up to our knowledge, only a few works so far have considered combining interior point methods (IPMs) with deep learning [17, 18, 19].

In this paper, we propose a novel neural network architecture called iRestNet, which results from unfolding a proximal IPM, over a finite number of iterations identifying with the network layers. One key feature of the proposed algorithm is that it produces iterates inside the feasible domain  $[0, \beta]^n$ , thanks to the introduction of a logarithmic barrier. Hence,

prior knowledge can directly and explicitly be incorporated into the network. Finally, it allows differentiation and gradient backpropagation throughout the network, which enables the stepsize, barrier parameter, and regularization weight to be untied across the network and learned for each layer.

This paper is organized as follows: in Section 2, we describe the proposed proximal IPM and derive expressions for the proximity operator of the barrier and its gradient. In Section 3, we present the proposed neural network architecture, while Section 4 is dedicated to numerical experiments and comparisons to state-of-the-art methods for image deblurring; finally, some conclusions are drawn in Section 5.

## 2. PROXIMAL INTERIOR POINT ALGORITHM

### 2.1. Notation and assumptions

Let  $\Gamma_0(\mathbb{R}^n)$  denote the set of functions which take values in  $\mathbb{R} \cup \{+\infty\}$  and are proper, convex, lower semicontinuous on  $\mathbb{R}^n$ . The regularization function  $\mathcal{R} \in \Gamma_0(\mathbb{R}^n)$  is assumed to be twice-differentiable. For every  $g \in \Gamma_0(\mathbb{R}^n)$ ,  $\gamma > 0$  and  $x \in \mathbb{R}^n$ , the proximity operator [20] of  $\gamma g$  at  $x$  is defined as

$$\text{prox}_{\gamma g}(x) = \underset{z \in \mathbb{R}^n}{\text{argmin}} \frac{1}{2} \|x - z\|^2 + \gamma g(z). \quad (3)$$

### 2.2. Interior point method

In the interior point framework, problem (2) is solved through a sequence of unconstrained subproblems of the form

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \frac{1}{2} \|Hx - y\|^2 + \lambda \mathcal{R}(x) + \mu \mathcal{B}(x) \quad (4)$$

where  $\mu > 0$  is the so-called barrier parameter, which vanishes along the minimization process, and  $\mathcal{B} \in \Gamma_0(\mathbb{R}^n)$  is the logarithmic barrier with unbounded derivative at the boundary of the feasible domain:  $(\forall x \in \mathbb{R}^n) \mathcal{B}(x) = \sum_{i=1}^n -\ln(\beta - x_i) - \ln(x_i)$  if  $x \in ]0, \beta[^n$ ,  $+\infty$  otherwise. Although standard IPMs require the inversion of an  $n \times n$  linear system at every iteration, these computationally expensive steps can be avoided by combining the interior point method with a proximal strategy, leading to very competitive solvers [21, 22]. We propose to combine the proximal IPM in [23] with a forward-backward step, which leads to the following update,

$$x_{k+1} = \text{prox}_{\gamma_k \mu_k \mathcal{B}}(x_k - \gamma_k (H^\top (Hx_k - y) + \lambda_k \nabla \mathcal{R}(x_k))) \quad (5)$$

$$= \mathcal{A}(x_k, \mu_k, \gamma_k, \lambda_k) \quad (6)$$

It can be noted that the regularization parameter is allowed to vary along iterations for more flexibility. The proximity operator of the barrier in (5) can be deduced from Proposition 1 below, using separability.

**Proposition 1** *Let  $b$  defined as  $(\forall u \in \mathbb{R}) b(u) = -\ln(\beta - u) - \ln(u)$  if  $u \in ]0, \beta[$ ,  $+\infty$  otherwise. Then,  $(\forall u \in \mathbb{R})$*

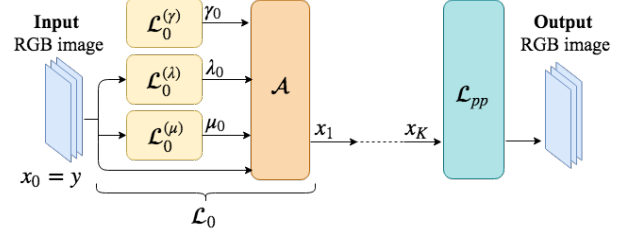


Fig. 1. iRestNet global architecture.

$(\forall \alpha > 0) \varphi(u, \alpha) = \text{prox}_{\alpha b}(u)$  is the unique solution in  $]0, \beta[$ , of the following cubic equation,

$$0 = v^3 - (\beta + u)v^2 + (\beta u - 2\alpha)v + \beta \alpha. \quad (7)$$

In addition, the partial derivatives of  $\varphi$  are given by

$$\frac{\partial \varphi}{\partial u}(u, \alpha) = \frac{\varphi(u, \alpha)(\varphi(u, \alpha) - \beta)}{3\varphi(u, \alpha)^2 - 2\varphi(u, \alpha)(\beta + u) + \beta u - 2\alpha}, \quad (8)$$

$$\frac{\partial \varphi}{\partial \alpha}(u, \alpha) = \frac{2\varphi(u, \alpha) - \beta}{3\varphi(u, \alpha)^2 - 2\varphi(u, \alpha)(\beta + u) + \beta u - 2\alpha}. \quad (9)$$

For sake of length, the proof of Proposition 1 is not provided here, but the main ingredients rely on [24, Example 4.15] and on the implicit function theorem.

## 3. PROPOSED ARCHITECTURE

We propose to unfold (6) over  $K$  iterations, and to use a supervised learning strategy in order to determine, from a training set of images, an optimal setting for the hyperparameters  $(\mu_k, \gamma_k, \lambda_k)_{0 \leq k \leq K-1}$ . For every  $k \in \{0, \dots, K-1\}$ , the  $k$ -th layer  $\mathcal{L}_k$  is built as the association of three hidden structures,  $\mathcal{L}_k^{(\mu)}$ ,  $\mathcal{L}_k^{(\gamma)}$  and  $\mathcal{L}_k^{(\lambda)}$ , used to infer the barrier parameter  $\mu_k$ , the stepsize  $\gamma_k$  and the regularization weight  $\lambda_k$ , respectively; which are followed by the update  $\mathcal{A}$ . Note that our framework allows the use of a post-processing step, that will be denoted as  $\mathcal{L}_{pp}$ . The resulting architecture is depicted in Figure 1.

### 3.1. Hidden structures

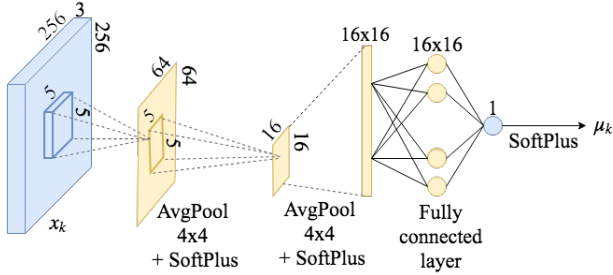
For every  $k \in \{0, \dots, K-1\}$ , the outputs  $(\mu_k, \gamma_k, \lambda_k)$  of the structures  $\mathcal{L}_k^{(\mu)}$ ,  $\mathcal{L}_k^{(\gamma)}$ , and  $\mathcal{L}_k^{(\lambda)}$  are made positive thanks to the Softplus function [25], defined below, which can be viewed as a smooth approximation of the ReLU activation function,

$$(\forall u \in \mathbb{R}) \text{Softplus}(u) = \ln(1 + \exp(u)). \quad (10)$$

Unlike ReLU, the gradient of Softplus is never strictly equal to zero, which helps propagating the gradient through the network. The stepsize is estimated as follows,

$$\gamma_k = \mathcal{L}_k^{(\gamma)} = \text{Softplus}(a_k), \quad (11)$$

where  $a_k$  is a scalar parameter learned during training. The barrier parameter is obtained using two convolutional layers

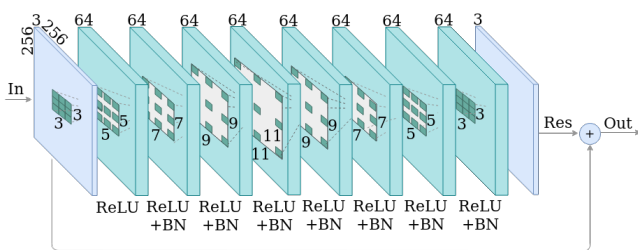


**Fig. 2.** Architecture of  $\mathcal{L}_k^{(\mu)}$ .

followed by a fully connected layer, as depicted in Figure 2. The regularization parameter  $\lambda_k$  is inferred from the image statistics and the following estimation of the noise level [26],

$$\hat{\sigma}(y) = \text{median}(|W_H y|) / 0.6745, \quad (12)$$

where  $|W_H y|$  is the vector gathering the absolute value of the diagonal coefficients of the first level Haar wavelet decomposition of the blurred image  $y$ . The proposed architecture does not require any prior knowledge about the noise level, in particular it does not have to be constant over the dataset. The expression of  $\mathcal{L}_k^{(\lambda)}$  is problem-dependent since it is built upon the chosen regularization strategy. A specific example will be given in Section 4 for the total variation regularization function. The architecture of the post-processing layer  $\mathcal{L}_{pp}$ , which is depicted in Figure 3, is inspired from [4]: it is made of 9 convolutional layers with different dilation factors, so as to widen the receptive field without creating memory issues. The artifacts that remain in the image after going through the  $K$  blocks of iRestNet map a small space, which is well captured by residual learning, hence the presence of a skip connection between the input of  $\mathcal{L}_{pp}$  and its output.



**Fig. 3.** Architecture of  $\mathcal{L}_{pp}$ . BN: batch normalization.

To train the network presented in Figure 1 using gradient descent, the chain rule, combined with automatic differentiation, can be applied for all steps in the network except for  $\mathcal{A}$ , whose gradient is directly derived from Proposition 1, since  $\nabla \mathcal{R}$  is assumed to be differentiable. For applications involving high risk and legal responsibility, it is critical to guarantee that the performance of neural networks remain acceptable when the input is perturbed. Explicit conditions securing the robustness of the proposed iRestNet architecture can be derived, using the recent work [27].

## 4. NUMERICAL EXPERIMENT

Numerical experiments are performed on a set of image deblurring problems. The total variation like function defined below is used as a regularizer,

$$(\forall x \in \mathbb{R}^n) \mathcal{R}(x) = \sum_{i=1}^n \sqrt{\frac{(D_v x)_i^2 + (D_h x)_i^2}{\delta^2} + 1}, \quad (13)$$

where  $D_v \in \mathbb{R}^{n \times n}$  and  $D_h \in \mathbb{R}^{n \times n}$  are the vertical and horizontal gradient operators, respectively, and  $\delta = 0.01$  is a smoothing parameter. The depth of iRestNet is fixed at  $K = 40$ . For every  $k \in \{0, \dots, K-1\}$ , the regularization parameter is inferred from the image statistics as follows,

$$\lambda_k = \mathcal{L}_k^{(\lambda)}(x_k) = \frac{\hat{\sigma}(x_0) \text{Softplus}(b_k)}{\eta(x_k) + \text{Softplus}(c_k)}, \quad (14)$$

where  $(b_k, c_k)$  is a pair of scalars learned by the network and  $\eta(x_k)$  is the standard deviation of the concatenated spatial gradients of  $x_k$ ,  $[(D_v x_k)^\top (D_h x_k)^\top]^\top$ .

### 4.1. Dataset and experimental settings

The training set is made of 200 RGB images from the Berkeley segmentation (BSD500) training set, and of 1000 RGB images from the COCO training set. We use the BSD500 validation set, which is made of 100 images, to monitor the training. The performance of iRestNet is evaluated on the 200 images from the BSD500 test set, which have been center-cropped using a  $256 \times 256$  window. Blurry images are produced using the following  $25 \times 25$  blur kernels and noise levels: (i) a Gaussian kernel, with a standard deviation of 1.6, and a Gaussian noise standard deviation of  $\sigma = 0.008$ ; this configuration is denoted as GaussA; (ii) the same kernel is used when  $\sigma$  is uniformly distributed between 0.01 and 0.05, the latter is denoted as GaussB; (iii) the Gaussian kernel with a standard deviation of 3, with  $\sigma = 0.04$ , denoted as GaussC; (iv) the third motion test kernel from [28], with  $\sigma = 0.01$ , denoted as Motion; (v) the square uniform kernel of size  $7 \times 7$ , with  $\sigma = 0.01$ , this configuration is referred to as Square.

### 4.2. Training

For each degradation model, one iRestNet is trained. The first 30 layers are trained individually for 40 epochs, with a decreasing learning rate multiplied by 0.9 every 10 epochs. This training strategy is chosen with regards to its low memory requirement. The rest of the network,  $\mathcal{L}_{pp} \circ \mathcal{L}_{39} \cdots \circ \mathcal{L}_{30}$ , is trained in an end-to-end fashion during 200-600 epochs, depending on the test configuration. We use the ADAM method to minimize the training loss defined as the opposite of the structural similarity measure (SSIM) [29]. The SSIM is a popular perceptual measure in image processing. Codes are implemented in Pytorch. All trainings are conducted using a

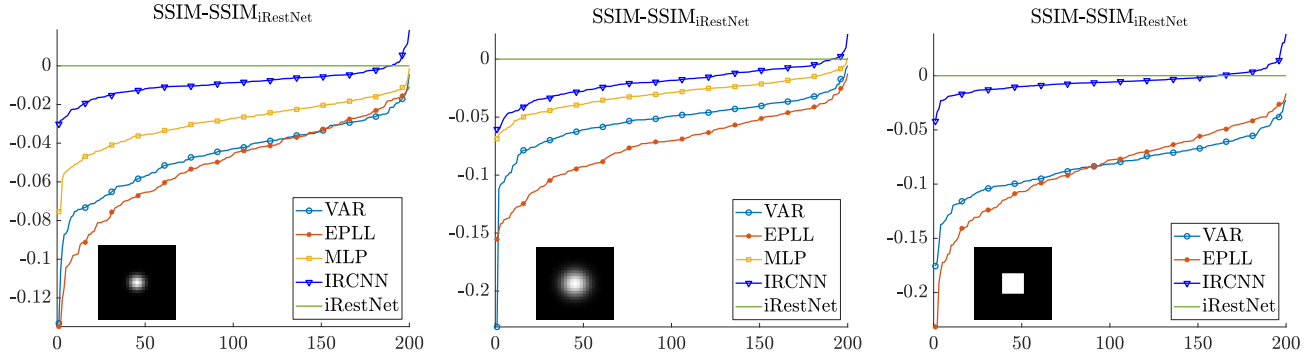


Fig. 4. Sorted SSIM gain on the test set, negative when iRestNet performs better. Left to right: GaussA, GaussC, Square.

GeForce GTX 1080 or 1070 GPU. The training takes approximately 3 to 4 days for each configuration.

### 4.3. Evaluation metrics and competitors

The restoration is evaluated based on the SSIM. The reconstruction given by iRestNet is compared with a solution to Problem (2) obtained using the projected gradient algorithm [30] with  $(\lambda, \delta)$  manually set so as to yield the best SSIM. This solution, referred to as VAR, gives an upper bound on the image quality that one can expect by solving (2). We also use the following deep learning methods for comparison: EPLL [31], MLP [7], and IRCNN [4]. When the setting is not relevant for these methods, no score is provided. Since MLP, EPLL and IRCNN require the knowledge of the noise level, we use the estimation  $\hat{\sigma}$  from (12) for GaussB, as the noise level is unknown in that case.

### 4.4. Results and discussion

The average SSIM obtained with the different methods for the various configurations on the test set can be found in Table 1. The mean SSIM achieved with iRestNet is greater than those obtained with the other methods for all configurations. For completeness, the SSIM of all images of the test set are plotted in Figure 4 for three configurations. As one can see, iRestNet performs well in terms of SSIM on almost all images. An example of the visual results obtained with the dif-

ferent methods can be found in Figure 5 for one test image degraded with GaussB. One can see from inspecting Figure 5 that the contrast and the details are better retrieved with iRestNet than with its competitors on this image.

	GaussA	GaussB	GaussC	Motion	Square
Blurred	0.675	0.522	0.326	0.548	0.543
VAR	0.804	0.724	0.585	0.829	0.756
EPLL	0.799	0.709	0.564	0.838	0.754
MLP	0.821	0.734	0.608	-	-
IRCNN	0.841	0.768	0.618	0.907	0.833
iRestNet	<b>0.850</b>	<b>0.786</b>	<b>0.638</b>	<b>0.911</b>	<b>0.839</b>

Table 1. SSIM results on the test set.

## 5. CONCLUSION

This paper presents a novel architecture based on a deep unfolded IPM with untied parameters. Range constraints are enforced on the solution thanks to a logarithmic barrier, which provides more control over the output of the network. Other contributions include the expression of the gradient of the required proximity operator. A direction for future work is to extend the method to situations in which the blur kernel is not fully known, so as to address blind deconvolution problems.

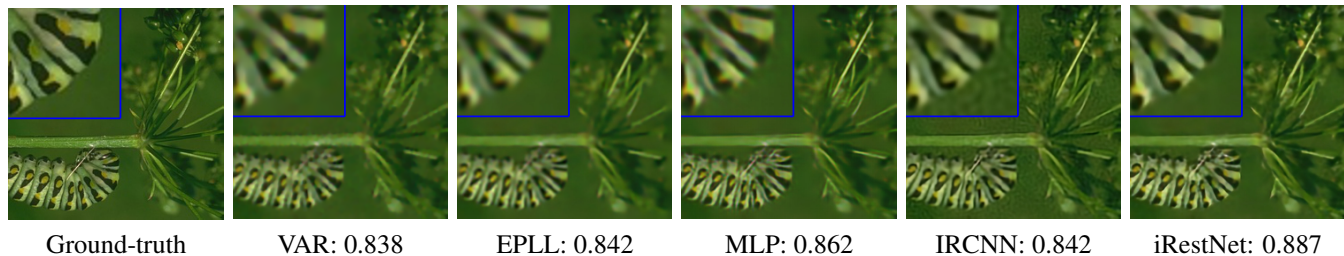


Fig. 5. Visual results and SSIM obtained with the different methods on one test image degraded with GaussB.

## 6. REFERENCES

- [1] J.-F. Aujol, “Some first-order algorithms for total variation based image restoration,” *Journal of Mathematical Imaging and Vision*, vol. 34, no. 3, pp. 307–327, 2009.
- [2] O. Scherzer, “The use of Morozov’s discrepancy principle for Tikhonov regularization for solving nonlinear ill-posed problems,” *Computing*, vol. 51, no. 1, pp. 45–60, 1993.
- [3] C.-A. Deledalle, S. Vaiteer, J. Fadili, and G. Peyré, “Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection,” *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, pp. 2448–2487, 2014.
- [4] 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)*, Jul 2017, vol. 2.
- [5] 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.
- [6] L. Xu, J. S. J. Ren, C. Liu, and J. Jia, “Deep convolutional neural network for image deconvolution,” in *Conference on Advances in Neural Information Processing Systems (NIPS)*, Dec 2014, pp. 1790–1798.
- [7] C. Schuler, H. C. Burger, S. Harmeling, and B. Schölkopf, “A machine learning approach for non-blind image deconvolution,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2013, pp. 1067–1074.
- [8] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, “Learning to deblur,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1439, 2016.
- [9] M. T. McCann, K. H. Jin, and M. Unser, “Convolutional neural networks for inverse problems in imaging: A review,” *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 85–95, 2017.
- [10] D. Boublil, M. Elad, J. Shtok, and M. Zibulevsky, “Spatially-adaptive reconstruction in computed tomography using neural networks,” *IEEE Transactions on Medical Imaging*, vol. 34, no. 7, pp. 1474–1485, 2015.
- [11] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” *arXiv preprint arXiv:1312.6199*, 2013.
- [12] J. R. Hershey, J. Le Roux, and F. Weninger, “Deep unfolding: model-based inspiration of novel deep architectures,” *arXiv preprint arXiv:1409.2574*, 2014.
- [13] K. Gregor and Y. LeCun, “Learning fast approximations of sparse coding,” in *International Conference on Machine Learning (ICML)*, Jun 2010, pp. 399–406.
- [14] J. Zhang and B. Ghanem, “ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2018, pp. 1828–1837.
- [15] X. Chen, J. Liu, Z. Wang, and W. Yin, “Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds,” in *Conference on Advances in Neural Information Processing Systems (NIPS)*, Dec 2018, pp. 9061–9071.
- [16] J. Sun, H. Li, Z. Xu, and Y. Yang, “Deep ADMM-Net for compressive sensing MRI,” in *Conference on Advances in Neural Information Processing Systems (NIPS)*, Dec 2016, pp. 10–18.
- [17] B. Amos and J. Z. Kolter, “OptNet: Differentiable optimization as a layer in neural networks,” in *International Conference on Machine Learning (ICML)*, Aug 2017, pp. 136–145.
- [18] T. B. Trafalis, T. A. Tutunji, and N. P. Couëllan, “Interior point methods for supervised training of artificial neural networks with bounded weights,” in *Network Optimization*, P. M. Pardalos, D. W. Hearn, and W. W. Hager, Eds., Berlin, Heidelberg, 1997, pp. 441–470, Springer Berlin Heidelberg.
- [19] P. T. Krasopoulos and N. G. Maratos, *An Interior Point Recurrent Neural Network for Convex Optimization Problems*, pp. 409–427, Springer New York, New York, NY, 2014.
- [20] H. H. Bauschke and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*, Springer, 2017.
- [21] M.-C. Corbineau, E. Chouzenoux, and J.-C. Pesquet, “PIPA: a new proximal interior point algorithm for large-scale convex optimization,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr 2018.
- [22] M.-C. Corbineau, E. Chouzenoux, and J.-C. Pesquet, “Geometry-texture decomposition/reconstruction using a proximal interior point algorithm,” in *IEEE Sensor Array and Multichannel Signal processing Workshop (SAM)*, Jul 2018.
- [23] A. Kaplan and R. Tichatschke, “Proximal methods in view of interior-point strategies,” *Journal of Optimization Theory and Applications*, vol. 98, no. 2, pp. 399–429, 1998.
- [24] C. Chaux, P. L. Combettes, J.-C. Pesquet, and V. R. Wajs, “A variational formulation for frame-based inverse problems,” *Inverse Problems*, vol. 23, no. 4, pp. 1495–1518, June 2007.
- [25] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia, “Incorporating second-order functional knowledge for better option pricing,” in *Conference on Advances in Neural Information Processing Systems (NIPS)*, Dec 2001, pp. 472–478.
- [26] A. Ramadhan, F. Mahmood, and A. Elci, “Image denoising by median filter in wavelet domain,” *arXiv preprint arXiv:1703.06499*, 2017.
- [27] P. L. Combettes and J.-C. Pesquet, “Deep neural network structures solving variational inequalities,” *arXiv preprint arXiv:1808.07526*, 2018.
- [28] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, “Understanding and evaluating blind deconvolution algorithms,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2009, pp. 1964–1971.
- [29] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [30] A. N. Iusem, “On the convergence properties of the projected gradient method for convex optimization,” *Computational Applied Mathematics*, vol. 22, no. 1, pp. 37–52, 2003.
- [31] D. Zoran and Y. Weiss, “From learning models of natural image patches to whole image restoration,” in *IEEE International Conference on Computer Vision (ICCV)*, Nov 2011, pp. 479–486.