ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10446244

IMAGE RESTORATION WITH GENERALIZED L2 LOSS AND CONVERGENT PLUG-AND-PLAY PRIORS

*Kartheek Kumar Reddy Nareddy**, *Abijith Jagannath Kamath**, *Chandra Sekhar Seelamantula*

Department of Electrical Engineering, Indian Institute of Science, Bengaluru-560012
Email:{nareddyreddy, abijithj, css}@iisc.ac.in

ABSTRACT

Image restoration involves solving an optimization problem where the objective function is the sum of a data-fidelity term and a regularization functional that incorporates a desired image prior. Solving the optimization problem using proximal methods results in iterative algorithms that require computing a gradient step corresponding to the data-fidelity loss and a proximal update corresponding to enforcing the image prior. In this paper, we develop a novel formulation for image restoration considering a generalized data-fidelity loss and a convex regularization function that enforces a desired image prior, and we solve the problem using proximal gradient method. The choice of the data-fidelity loss is such that the adjoint operator is reminiscent of Wiener filtering when the forward operator is a convolutional operator (for instance, a shift-invariant blur kernel). The proposed gradient update ensures that the iterates remain in the solution-space of the linear measurement constraints. We further propose the plug-and-play counterpart of the restoration technique, which allows one to leverage off-the-shelf data-driven denoisers in place of the proximal operator. Experimental validations carried out on BSD500, Brodatz, Urban100, and DIV2K datasets show that the proposed technique gives rise to superior image reconstruction quality compared with the state-of-the-art techniques, with the performance measured in terms of peak signal-to-noise ratio (PSNR) and structural similarity index metric (SSIM), with comparable computational complexity.

Index Terms— Image restoration, inverse problems, proximal methods, plug-and-play methods, incoherent adjoint operators.

1. INTRODUCTION

Inverse problems are pivotal in contemporary image processing and computational imaging, with wide-ranging applications encompassing denoising, deconvolution, super-resolution, and compressed sensing [1, 2]. The goal is to recover/restore the ground-truth image (vectorized) $\mathbf{x}^* \in \mathbb{R}^n$ from noisy, linear measurements

$$\mathbf{y} = \mathbf{H}\mathbf{x}^* + \mathbf{w} \in \mathbb{R}^m, \quad m \leq n, \tag{1}$$

where $\mathbf{H} \in \mathbb{R}^{m\times n}$ is the *forward operator* of the measurement process, and $\mathbf{w} \in \mathbb{R}^m$ denotes additive noise. Image restoration can be posed as an optimization problem of the type

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \ \frac{1}{2}\|\mathbf{H}\mathbf{x}-\mathbf{y}\|_2^2 + \lambda\, g(\mathbf{x}), \tag{$\mathbf{P}_1$}$$

*Equal contribution.

This work is supported by the Prime Minister's Research Fellowship.

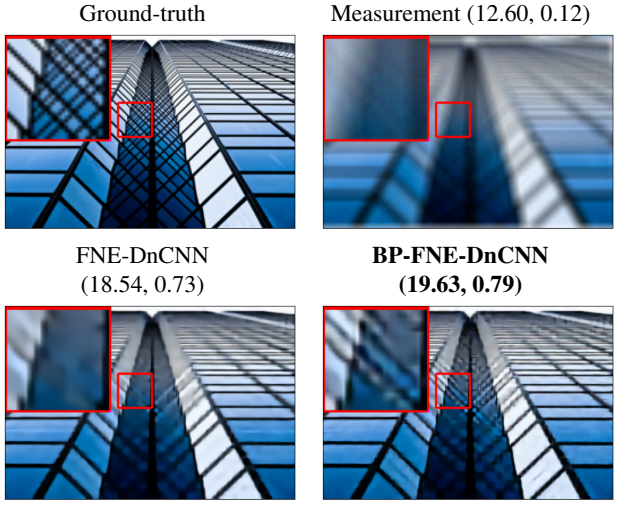

Fig. 1. Recovery from square blurred measurements using the proposed technique compared with the benchmark techniques. The numbers indicate (PSNR (in dB), SSIM).

where $\lambda > 0$ is the *regularization parameter* and $g : \mathbb{R}^n \to \mathbb{R}$ is the regularization functional that enforces the desired image prior. The solutions to the minimization problem in ($\mathbf{P}_1$) are typically iterative, i.e., the goal is to design a sequence of iterates $\{\mathbf{x}_k\}_{k\in\mathbb{N}}$ that converges to a limit point as close to $\mathbf{x}^*$ as possible. The archetypal choice for the image prior g is the ℓ_1-norm of the wavelet transform of $\mathbf{x}$ that exploits sparsity in the detail coefficients of natural images [1–3]. Correspondingly, the proximal gradient (ISTA) updates [4, 5] $\mathbf{x}_k \mapsto \mathbf{x}_{k+1}$ are given via wavelet thresholding [3] as follows:

$$\mathbf{x}_{k+1} = \Psi^\dagger \mathcal{T}_{\eta\lambda}\left\{\Psi(\mathbf{x}_k - \eta\mathbf{H}^\top(\mathbf{H}\mathbf{x}_k - \mathbf{y}))\right\},$$

where Ψ denotes the 2D dyadic wavelet transform, $\Psi^\dagger$ denotes the left-inverse of Ψ, and $\mathcal{T}_{\eta\lambda}(\mathbf{x})$ denotes the soft-thresholding or shrinkage-thresholding operator. The operator $(\Psi^\dagger\mathcal{T}_{\eta\lambda}\Psi)$ is the proximal operator of $\lambda\|\Psi\mathbf{x}\|_1$. A generic form of the update equation, where g in ($\mathbf{P}_1$) is a proper, closed and convex function, is given below:

$$\mathbf{x}_{k+1} = \text{prox}_{\eta g}(\mathbf{x}_k - \eta\mathbf{H}^\top(\mathbf{H}\mathbf{x}_k - \mathbf{y})), \tag{2}$$

where $\text{prox}_{\eta g}$ denotes the proximal operator [4] of g and $\eta > 0$ denotes the step-size parameter in the gradient update. The form given

in Eq. (2) encompasses a wide range of image priors [6,7], including those that promote sparsity [5, 8]. Fast counterparts of Eq. (2) can be obtained using Nesterov's momentum [9–11] and pre-conditioned gradient methods [12, 13]. Venkatkrishnan *et al.* [14] interpreted the proximal operation as equivalent to denoising, and introduced the plug-and-play (PnP) framework [15], which allows one to incorporate a more powerful denoiser $\mathcal{D}_\theta : \mathbb{R}^n \to \mathbb{R}^n$ instead of the proximal operator, where θ denotes the parameters of the denoiser. The denoiser may be model-based [16, 17] or data-driven/deep-learning-based [18]. The corresponding iterates take the form:

$$\mathbf{x}_{k+1} = \mathcal{D}_\theta \left\{ \mathbf{x}_k - \eta \mathbf{H}^\top (\mathbf{H}\mathbf{x}_k - \mathbf{y}) \right\}. \tag{3}$$

These iterates do not necessarily correspond to an optimization problem of the form given in ($\mathbf{P}_1$), and are not always provably convergent making them susceptible to adversarial attacks [19, 20]. Recently, several works have proposed provably-convergent PnP methods for solving linear inverse problems [21–24]. The denoiser $\mathcal{D}_\theta$ in [24] relies on DnCNN [18], with the added constraint that it should be firmly nonexpansive (FNE) during training. The resulting FNE–DnCNN ensures that the iterates in Eq. (3) converge to a fixed-point, which is characterized via monotone inclusion [24].

1.1. Contributions of This Paper

Although a lot of research has gone into the choice of g considering various priors and the associated reconstruction algorithms, the choice of the data-fidelity term has remained relatively untouched. To start with, we utilize a generalized data-fidelity loss with a convex regularization functional (with the objective function being semi-smooth). The generalized loss is the back-projection loss [25–27]. The choice of the data-fidelity loss is such that the adjoint operator is reminiscent of Wiener filtering when the forward operator is a convolutional operator (for instance, a shift-invariant blur kernel). Hence, we refer to the PGM algorithm as BP-PGM. We then consider the plug-and-play counterpart using Pesquet *et al.*'s FNE–DnCNN as the denoiser [24], and the resulting technique is dubbed "BP-FNE-DnCNN". We show that the iterates of BP-PGM converge to the minimizer with the same rate as ISTA, and that the iterates of BP-FNE-DnCNN converge to a fixed-point. The proposed technique applied to image deconvolution improves the reconstruction performance in terms of PSNR and SSIM (Section 3), compared with the existing state-of-the-art techniques. To give a flavor of the results to come, Fig. 1 shows the image restoration performance on an image taken from the Urban100 dataset [28] using BP-FNE-DnCNN vis-à-vis the benchmark techniques. BP-FNE-DnCNN has the best performance in terms of PSNR and SSIM, and has fewer visual artifacts.

2. PROBLEM FORMULATION

The starting point is the model considered in Eq. (1) where the forward linear operator $\mathbf{H}$ has full row-rank. As opposed to ($\mathbf{P}_1$), consider the optimization with the ℓ_2-based data-fidelity term replaced by the generalized ℓ_2 loss counterpart resulting in the objective function

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \; \underbrace{\frac{1}{2}\left\|\mathbf{H}\mathbf{x}-\mathbf{y}\right\|_{\mathbf{B}}^2}_{f(\mathbf{x})} + \lambda g(\mathbf{x}), \tag{$\mathbf{P}_2$}$$

where $\mathbf{B} = (\mathbf{H}\mathbf{H}^\top)^{-\frac{1}{2}}$ is a positive-definite matrix, and g is a proper, closed, and convex regularization function that enforces a desired image prior. This loss is the back-projection loss [25–27] and has shown improvements over the least-squares loss in applications such as deblurring, super-resolution, and compressed sensing. The gradient of the data-fidelity loss is given by

$$\nabla f(\mathbf{x}) = \mathbf{H}^\top (\mathbf{H}\mathbf{H}^\top)^{-1}(\mathbf{H}\mathbf{x}-\mathbf{y}), \tag{4}$$

which is β-Lipschitz, with $\beta = \left\|\mathbf{H}^\top (\mathbf{H}\mathbf{H}^\top)^{-1}\mathbf{H}\right\|_2 = 1$, i.e.,

$$\forall \mathbf{x}, \tilde{\mathbf{x}} \in \mathbb{R}^n, \; \left\|\nabla f(\mathbf{x}) - \nabla f(\tilde{\mathbf{x}})\right\|_2 \leq \beta \left\|\mathbf{x}-\tilde{\mathbf{x}}\right\|_2.$$

Therefore, the objective function is semi-smooth, i.e., it is the sum of a β-smooth, data-fidelity loss, and a convex regularization function g that is not necessarily differentiable. Thence, ($\mathbf{P}_2$) can be solved using PGM, and the iterates $\mathbf{x}_k \mapsto \mathbf{x}_{k+1}$ are given by

$$\begin{aligned} \mathbf{x}_{k+1} &= \text{prox}_{\eta g}(\mathbf{x}_k - \eta \mathbf{H}^\top (\mathbf{H}\mathbf{H}^\top)^{-1}(\mathbf{H}\mathbf{x}_k - \mathbf{y})), \\ &= \text{prox}_{\eta g}(\mathbf{x}_k - \eta \mathbf{H}^\dagger (\mathbf{H}\mathbf{x}_k - \mathbf{y})), \end{aligned} \tag{5}$$

where $\mathbf{H}^\dagger$ denotes the Moore-Penrose pseudoinverse of $\mathbf{H}$, and $0 < \eta < 2$ denotes the step-size. For instance, when $\mathbf{H}$ is a convolutional operator, $\mathbf{H}^\dagger$ is its convolutional inverse. At each iteration, the residue $\mathbf{H}\mathbf{x}_k - \mathbf{y}$ is projected onto the range-space of $\mathbf{H}^\dagger$. For generic $\mathbf{H}$, the gradient update with $\eta = 1$ lies in the solution space $\{\mathbf{x} \in \mathbb{R}^n \text{ s.t. } \mathbf{y} = \mathbf{H}\mathbf{x}\}$ as verified below:

$$\begin{aligned} \mathbf{H}\left(\mathbf{x}_k - \mathbf{H}^\dagger(\mathbf{H}\mathbf{x}_k - \mathbf{y})\right) &= \mathbf{H}\mathbf{x}_k - \mathbf{H}\mathbf{H}^\dagger(\mathbf{H}\mathbf{x}_k - \mathbf{y}) \\ &= \mathbf{H}\mathbf{x}_k - \mathbf{H}\mathbf{x}_k + \mathbf{y} = \mathbf{y}. \end{aligned}$$

When the forward operator does not have full row-rank, we consider the regularized version of the norm in ($\mathbf{P}_2$) with $\mathbf{B} = (\mathbf{H}\mathbf{H}^\top + \varepsilon\mathbf{I})^{-\frac{1}{2}}$, where $\varepsilon > 0$. Within the PGM framework, the iterates take the form

$$\begin{aligned} \mathbf{x}_{k+1} &= \text{prox}_{\eta g}(\mathbf{x}_k - \eta \mathbf{H}^\top (\mathbf{H}\mathbf{H}^\top + \varepsilon\mathbf{I})^{-1}(\mathbf{H}\mathbf{x}_k - \mathbf{y})), \\ &= \text{prox}_{\eta g}(\mathbf{x}_k - \eta \mathbf{H}_\varepsilon^\dagger (\mathbf{H}\mathbf{x}_k - \mathbf{y})). \end{aligned} \tag{6}$$

When ε take the value of σ of the input noise, then the above update matches the Wiener filter. In this paper, we choose ϵ based on the performance of the proposed method on a validation dataset. This makes our method different from Wiener filter.

This update resembles gradient-preconditioned updates [12, 13]:

$$\mathbf{x}_{k+1} = \text{prox}_{\eta g}(\mathbf{x}_k - \eta \mathbf{U}_k^{-1} \mathbf{H}^\top (\mathbf{H}\mathbf{x}_k - \mathbf{y})),$$

where $\mathbf{U}_k^{-1}$ denotes the preconditioner at iteration k. The key difference is our BP-PGM method takes following form:

$$\mathbf{x}_{k+1} = \text{prox}_{\eta g}(\mathbf{x}_k - \eta \mathbf{H}^\top \mathbf{U}_k^{-1} (\mathbf{H}\mathbf{x}_k - \mathbf{y})),$$

where the placement of $\mathbf{U}_k^{-1}$ is different from preconditioned gradient method.

Algorithm 1: BP-FNE-DNCNN for image restoration

Input: Measurements $\mathbf{y}$, forward operator $\mathbf{H}$, step-size $\eta \in]0, 2[$, $\varepsilon > 0$

Initialization: $\mathbf{x}_0 = \mathbf{H}_\varepsilon^\dagger \mathbf{y}$

for $k = 1, 2, \ldots$ *until convergence* **do**

  $\mathbf{x}_{k+1} = \mathcal{D}_\theta(\mathbf{x}_k - \eta \mathbf{H}_\varepsilon^\dagger (\mathbf{H}\mathbf{x}_k - \mathbf{y}))$

Output: $\mathbf{x}_{k+1}$

2.1. BP-FNE-DnCNN

We integrate BP-PGM and the plug-and-play framework employing FNE-DnCNN as the denoiser [24]. Since the formulation comprises the back-projection loss with FNE-DnCNN as the denoiser, we refer to our technique as BP-FNE-DnCNN. Replacing the proximal operator $\text{prox}_{\eta g}$ in Eq.(6) with a more powerful off-the-shelf denoiser $\mathcal{D}_\theta$ results in iterates of the form

$$\mathbf{x}_{k+1} = \mathcal{D}_\theta(\mathbf{x}_k - \eta \mathbf{H}_\varepsilon^\dagger(\mathbf{H}\mathbf{x}_k - \mathbf{y})). \tag{7}$$

If the denoiser is firmly nonexpansive, convergence of the iterates in Eq. (7) can guaranteed, as we shall show in Proposition 2, along the lines of [24]. Akin to [24], we use the DnCNN denoiser [18], whose schematic is shown in Fig. 2. The first layer is a convolutional layer with 64 filters of size $3 \times 3 \times c$, where c denotes the number of channels in the input image. The layers 2 to $D-1$ are convolutional layers with 64 filters of size $3 \times 3 \times 64$ with batch normalization. The last layer is a convolutional layer with c filters of size $3 \times 3 \times 64$. The nonlinear activation in all the layers is the leaky rectified linear unit: $\text{LeakyReLU}(x) = \max\{\alpha x, x\}$, with $\alpha = 0.01$. Since the network is designed to predict the residue, a skip connection from the input to the last layer is required to give the denoised image. The parameters θ of the network are the filter weights in all the D convolutional layers.

Training the denoiser: The denoiser can be trained as outlined in [24]. Let $\{\mathbf{x}_i\}_{i=1}^N$ be the training set of ground-truth images. Noisy measurements $\{\mathbf{z}_i\}_{i=1}^N$ are constructed as $\mathbf{z}_i = \mathbf{x}_i + \mathbf{w}_i$, where $\mathbf{w}_i \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ denotes additive white Gaussian noise with standard deviation σ. The denoiser training problem is posed as

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} \|\mathcal{D}_\theta(\mathbf{z}_i) - \mathbf{x}_i\|_2^2 + \mu \max\left\{\|2\nabla\mathcal{D}_\theta(\tilde{\mathbf{x}}_i)\|_2^2, 1-\delta\right\},$$

where $\mu, \delta > 0$ are hyperparameters, and $\tilde{\mathbf{x}}_i = \rho\mathbf{x}_i + (1-\rho)\mathcal{D}_\theta(\mathbf{z}_i)$, where $\rho \sim \mathcal{U}[0,1]$. The loss function is a sum of the two terms: consistency between the input and the output, and the gradient penalty that promotes the network to be firmly nonexpansive. The network is trained using the Adam optimizer [29].

2.2. Objective Convergence of BP-FNE-DnCNN

The proposed technique is a convergent method for image restoration. We show that the iterates of BP-PGM given in Eq. (6) converge in the objective function. Note that this subsumes the convergence of the iterates in Eq. (5) by setting $\varepsilon = 0$.

Proposition 1 (Convergence of BP-PGM)**.** *Let $0 < \eta < 2/\beta$, let $F(\mathbf{x}) = f(\mathbf{x}) + \lambda g(\mathbf{x})$ denote the objective function, and let $\mathbf{x}^* \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} F(\mathbf{x})$ denote the set of minimizers. The minimizing sequence $\{\mathbf{x}_k\}_{k\in\mathbb{N}}$ in Eq. (6) satisfies the condition*

$$F(\mathbf{x}_k) - F(\mathbf{x}^*) \leq \frac{\beta\|\mathbf{x}_0 - \mathbf{x}^*\|_2^2}{2k}, \forall k \in \mathbb{N}.$$

The proof follows from [10]. The iterates have a rate of convergence identical to that of ISTA [10]. Next, we show that the iterates of BP-FNE-DnCNN given in Eq. (3) converge to a fixed-point characterized by monotone inclusion.

Proposition 2 (Fixed-point convergence of BP-FNE-DnCNN)**.** *Let $0 < \eta < 2/\beta$. Suppose $\mathcal{D}_\theta$ is a firmly nonexpansive operator, and let $A = (\mathcal{D}_\theta^{-1} - \text{Id})$. Assume $\mathcal{S}_\eta = \{\mathbf{x} : (\nabla f + \frac{1}{\eta}A)(\mathbf{x}) = \mathbf{0}\}$ is nonempty. Then, the sequence $\{\mathbf{x}_k\}_{k\in\mathbb{N}}$ in Eq. (7) converges to $\hat{\mathbf{x}} \in \mathcal{S}_\eta$, i.e.,*

$$\mathbf{0} \in \nabla f(\hat{\mathbf{x}}) + \frac{1}{\eta}A(\hat{\mathbf{x}}).$$

The proof follows [24]. The iterates of BP-FNE-DnCNN converge to the zeros of $(\nabla f + \frac{1}{\eta}A)$, which implies fixed-point convergence of the sequence in Eq. (7) [5].

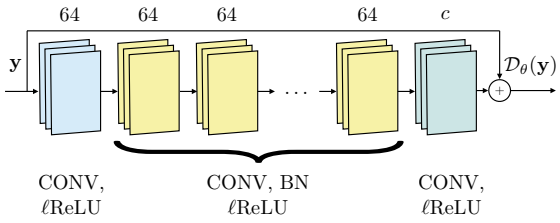

Fig. 2. Architecture of the denoiser: all layers are convolutional with LeakyReLU activation; layers 2 to $D-1$ have batch normalization. The first $D-1$ layers have 64 filters each, and the last layer has c filters, where c is the number of channels in the input image.

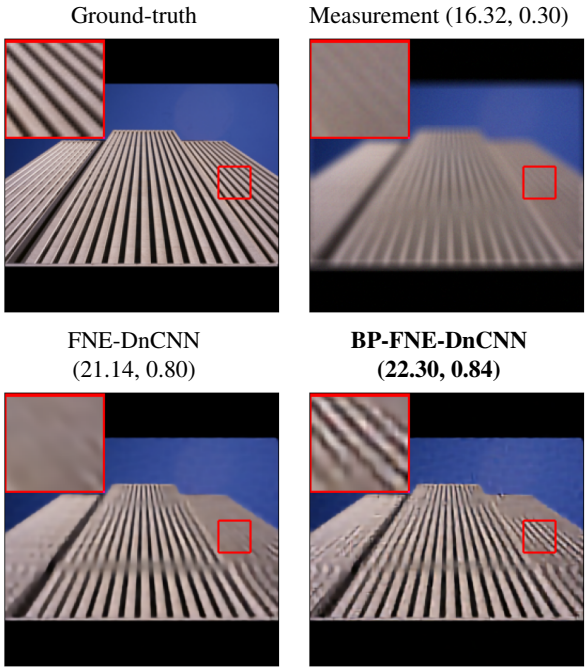

Fig. 3. Recovery from square blurred measurements using the proposed technique compared with the benchmark technique. The numbers indicate (PSNR (in dB), SSIM).

3. EXPERIMENTAL RESULTS

We demonstrate the proposed image restoration technique for image deconvolution problems. Measurements of a ground-truth image $\mathbf{x}^*$ (vectorized) are obtained as

$$\mathbf{y} = \mathbf{h} * \mathbf{x}^* + \mathbf{w},$$

where $\mathbf{h}$ is a convolutional kernel, and $\mathbf{w}$ denotes additive white Gaussian noise. We evaluate the performance of the proposed

Table 1. Performance of the proposed technique and benchmarks evaluated using PSNR↑ (in dB) and SSIM↑ on DIV2K, Urban100, BSD500, and Brodatz datasets. The best scores are shown in **bold**, and the second best scores are shown underlined.

Method	Gaussian Blur	Motion Blur	Square Blur	CPU / GPU
	DIV2K			Time (sec)
TV-ADMM [6]	$15.97 \pm 2.14, 0.3150 \pm 0.1485$	$16.31 \pm 2.13, 0.3195 \pm 0.1498$	$15.91 \pm 2.14, 0.3124 \pm 0.1483$	$\mathbf{9.74 \pm 0.46}$ / -
PnP-BM3D	$21.69 \pm 2.62, 0.5267 \pm 0.1529$	$23.36 \pm 2.62, 0.6410 \pm 0.1186$	$23.33 \pm 2.72, 0.6252 \pm 0.1279$	45.30 ± 4.10 / -
BP-BM3D [26]	$21.92 \pm 2.64, 0.5380 \pm 0.1504$	$24.34 \pm 2.71, 0.6859 \pm 0.1044$	$24.03 \pm 2.73, 0.6650 \pm 0.1139$	45.82 ± 4.09 / -
PnP-NLM [21]	$22.48 \pm 2.52, 0.5340 \pm 0.1143$	$24.14 \pm 1.47, 0.5799 \pm 0.0574$	$21.25 \pm 0.94, 0.4363 \pm 0.0712$	98.57 ± 8.86 / -
FNE-DnCNN [24]	$22.80 \pm 2.77, 0.5906 \pm 0.1455$	$26.91 \pm 2.95, 0.7927 \pm 0.0806$	$26.73 \pm 2.88, 0.7855 \pm 0.0871$	- / 12.34 ± 0.76
BP-FNE-DnCNN (Ours)	$\mathbf{23.25 \pm 2.74}, \mathbf{0.6142 \pm 0.1340}$	$\mathbf{27.14 \pm 2.90}, \mathbf{0.8039 \pm 0.0734}$	$\mathbf{27.10 \pm 2.83}, \mathbf{0.8024 \pm 0.0757}$	- / 13.60 ± 0.94
	Urban100			Time (sec)
TV-ADMM [6]	$14.26 \pm 1.96, 0.1992 \pm 0.1073$	$14.47 \pm 2.00, 0.2005 \pm 0.1073$	$14.22 \pm 1.96, 0.1995 \pm 0.1067$	$\mathbf{4.18 \pm 0.72}$ / -
PnP-BM3D	$18.34 \pm 2.90, 0.3526 \pm 0.1240$	$1971 \pm 2.86, 0.4848 \pm 0.0984$	$19.68 \pm 2.90, 0.4699 \pm 0.1002$	26.83 ± 0.95 / -
BP-BM3D [26]	$18.50 \pm 2.91, 0.3631 \pm 0.1230$	$20.46 \pm 2.84, 0.5395 \pm 0.0897$	$20.30 \pm 2.90, 0.5241 \pm 0.0908$	26.77 ± 0.82 / -
PnP-NLM [21]	$19.02 \pm 2.82, 0.3875 \pm 0.0998$	$21.32 \pm 2.09, 0.5555 \pm 0.0713$	$19.83 \pm 1.51, 0.4824 \pm 0.0947$	46.03 ± 5.05 / -
FNE-DnCNN [24]	$19.17 \pm 2.96, 0.4239 \pm 0.1266$	$22.64 \pm 2.72, 0.6861 \pm 0.0869$	$22.85 \pm 2.78, 0.6984 \pm 0.0855$	- / 6.70 ± 0.63
BP-FNE-DnCNN (Ours)	$\mathbf{19.58 \pm 2.90}, \mathbf{0.4580 \pm 0.1180}$	$\mathbf{23.02 \pm 2.64}, \mathbf{0.7124 \pm 0.0800}$	$\mathbf{23.49 \pm 2.62}, \mathbf{0.7387 \pm 0.0753}$	- / 7.31 ± 0.68
	BSD500			Time (sec)
TV-ADMM [6]	$17.24 \pm 2.43, 0.3486 \pm 0.1456$	$17.64 \pm 2.46, 0.3535 \pm 0.1469$	$17.19 \pm 2.44, 0.3444 \pm 0.1444$	$\mathbf{12.16 \pm 0.09}$ / -
PnP-BM3D	$22.82 \pm 3.62, 0.5121 \pm 0.1552$	$23.92 \pm 3.05, 0.5950 \pm 0.1356$	$23.86 \pm 3.07, 0.5845 \pm 0.1388$	49.73 ± 2.19 / -
BP-BM3D [26]	$22.80 \pm 3.05, 0.5226 \pm 0.1537$	$24.62 \pm 3.12, 0.6330 \pm 0.1262$	$24.37 \pm 3.06, 0.6160 \pm 0.1274$	50.39 ± 2.02 / -
PnP-NLM [21]	$23.05 \pm 2.80, 0.5046 \pm 0.1201$	$23.54 \pm 2.50, 0.5032 \pm 0.0579$	$21.16 \pm 2.67, 0.3785 \pm 0.0852$	132.75 ± 2.57/ -
FNE-DnCNN [24]	$23.51 \pm 3.17, 0.5575 \pm 0.1528$	$26.43 \pm 3.40, 0.7145 \pm 0.1159$	$26.45 \pm 3.27, 0.7136 \pm 0.1136$	- / 15.07 ± 0.94
BP-FNE-DnCNN (Ours)	$\mathbf{23.79 \pm 3.12}, \mathbf{0.5711 \pm 0.1448}$	$\mathbf{26.58 \pm 3.33}, \mathbf{0.7271 \pm 0.1062}$	$\mathbf{26.71 \pm 3.19}, \mathbf{0.7328 \pm 0.1125}$	- / 16.14 ± 1.07
	Brodatz			Time (sec)
TV-ADMM [6]	$16.70 \pm 4.62, 0.1760 \pm 0.1636$	$16.81 \pm 4.67, 0.1762 \pm 0.1640$	$16.66 \pm 4.62, 0.1725 \pm 0.1607$	$\mathbf{7.28 \pm 0.07}$ / -
PnP-BM3D	$20.11 \pm 5.76, 0.4048 \pm 0.2294$	$21.80 \pm 5.70, 0.5240 \pm 0.1721$	$21.58 \pm 5.84, 0.4860 \pm 0.1960$	26.93 ± 3.20 / -
BP-BM3D [26]	$20.30 \pm 5.81, 0.4128 \pm 0.2275$	$22.70 \pm 5.78, 0.5712 \pm 0.1551$	$22.27 \pm 5.81, 0.5251 \pm 0.1787$	26.27 ± 3.30 / -
PnP-NLM [21]	$20.85 \pm 5.59, 0.4366 \pm 0.1987$	$22.52 \pm 3.70, 0.5507 \pm 0.0786$	$20.30 \pm 2.62, 0.4362 \pm 0.0761$	86.94 ± 3.58 / -
FNE-DnCNN [24]	$20.76 \pm 5.97, 0.4293 \pm 0.2262$	$24.28 \pm 5.69, 0.6295 \pm 0.1421$	$23.88 \pm 5.88, 0.5834 \pm 0.1651$	- / 13.49 ± 0.92
BP-FNE-DnCNN (Ours)	$\mathbf{21.19 \pm 5.77}, \mathbf{0.4527 \pm 0.2162}$	$\mathbf{24.56 \pm 5.41}, \mathbf{0.6572 \pm 0.1212}$	$\mathbf{24.33 \pm 5.43}, \mathbf{0.6252 \pm 0.1326}$	- / 15.98 ± 0.05

technique with the benchmarks using the peak signal-to-noise ratio (PSNR) and structural similarity index metric (SSIM) as objective metrics, over the following datasets: BSD500 [30], DIV2K [31, 32], and Urban100 [28]. We run Algorithm 1 with fixed step-size parameter $\eta = 1.99$ and $\varepsilon = 0.05$ for a maximum of 300 iterations. We use the following convergent techniques as benchmarks: total-variation minimization using ADMM (TV-ADMM) [6], PnP-BM3D, BP-BM3D [26], PnP-NLM [21] and FNE-DnCNN [24] with parameters set to provide the best performance.

Training details: The network is trained similar to [24]. We train the denoiser $\mathcal{D}_\theta$ on 50,000 images from the validation set of ImageNet [33]. The architecture is fixed to have $D = 20$ convolutional layers. The measurements are corrupted with white Gaussian noise of standard deviation drawn uniformly at random over $[0, 0.1]$. The network is trained using the Adam optimizer with learning rate 10^{-4} for 200 epochs with $\mu = 10^{-5}$ and $\delta = 10^{-2}$.

Inferences: Table 1 shows the comparison in performance evaluated using PSNR and SSIM between the proposed method BP-FNE-DnCNN and the benchmarks. The proposed technique outperforms the benchmarks over all the datasets and over all the blur kernels under consideration. TV-ADMM converges the fastest, however, to a suboptimal reconstruction. PnP-BM3D method has improved performance over TV-ADMM, while the time taken is significantly higher. BP-BM3D method takes similar amount of time for reconstruction, while improving the performance by a margin. The PnP-NLM technique uses the non-local means image denoiser (NLM [16]) as a denoiser in the plug-and-play framework. PnP-NLM has a comparable performance to FNE-DnCNN, although it does not require training. However, PnP-NLM is computationally intensive. FNE-DnCNN and the proposed BP-FNE-DnCNN have similar run-times, however, BP-FNE-DnCNN has a superior performance in terms of PSNR and SSIM. Figs. 1 and 3 show examples of image reconstruction using FNE-DnCNN and BP-FNE-DnCNN obtained over the Urban100 dataset. The proposed method has superior quality reconstruction over the benchmark. The zoomed-in portion of the pictures in the red bounding box shows the difference in reconstruction quality between the benchmark and the proposed technique. BP-FNE-DnCNN is able to reconstruct the straight-line patterns from blurred images better than FNE-DnCNN. The improvement in performance is attributed to the fact that the gradient step in the proposed technique updates the reconstruction to a point that is closer to the solution-space of the linear measurement constraint.

4. CONCLUSIONS

We introduced a formulation for image restoration where the objective is the sum of a back-projection loss for data-fidelity and a regularization functional. The objective function is semi-smooth and can be solved using the proximal gradient method. We then incorporated this formulation within a plug-and-play framework where the denoiser is a learnt deep convolutional neural network, specifically a firmly nonexpansive DnCNN, thereby rendering the iterations provably convergent. Experimental results show that the proposed technique outperforms the state-of-the-art in terms of PSNR and SSIM. A careful qualitative analysis of the reconstructed images shows that proposed technique is able to reconstruct fine details from the blurred measurement unlike the benchmark techniques. The proposed technique can be used for solving inverse problems in computational imaging applications such as lensless imaging [34], diffuser-based imaging [35], coded-aperture imaging [36], etc.

5. REFERENCES

[1] A. Ribes and F. Schmitt, "Linear inverse problems in imaging," *IEEE Signal Process. Mag.*, vol. 25, no. 4, pp. 84–99, 2008.

[2] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb, "Solving inverse problems using data-driven models," *Acta Numerica*, vol. 28, pp. 1–174, 2019.

[3] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004.

[4] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Opt.*, no. 3, pp. 127–239, 2014.

[5] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics, 2011.

[6] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[7] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 9446–9454, 2018.

[8] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen, *Variational Methods in Imaging*. Springer, 2009.

[9] Y. E. Nesterov, "A method of solving a convex programming problem with convergence rate $\mathcal{O}(\frac{1}{k^2})$," in *Russ. Acad. Sci. Dokl. Akad. Nauk*, vol. 269, pp. 543–547, 1983.

[10] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci,*, vol. 2, no. 1, pp. 183–202, 2009.

[11] A. Chambolle and C. Dossal, "On the convergence of the iterates of the "fast iterative shrinkage/thresholding algorithm"'," *J. Opt. Theory Appl.*, vol. 166, pp. 968–982, 2015.

[12] E. Chouzenoux, J.-C. Pesquet, and A. Repetti, "Variable metric forward–backward algorithm for minimizing the sum of a differentiable function and a convex function," *J. Optim. Theory Appl.*, vol. 162, no. 1, pp. 107–132, 2014.

[13] M. Savanier, E. Chouzenoux, J.-C. Pesquet, and C. Riddell, "Unmatched preconditioning of the proximal gradient algorithm," *IEEE Signal Process. Lett.*, vol. 29, pp. 1122–1126, 2022.

[14] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *IEEE Glob. Conf. Signal Inf. Process. (GlobalSIP)*, pp. 945–948, 2013.

[15] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *SIAM J. Imag. Sci,*, vol. 10, no. 4, pp. 1804–1844, 2017.

[16] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *IEEE Comp. Vis. Patt. Recogn. (CVPR)*, vol. 2, pp. 60–65, 2005.

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

[18] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, 2017.

[19] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen, "On instabilities of deep learning in image reconstruction and the potential costs of AI," *Proc. Nat. Acad. Sci*, vol. 117, no. 48, pp. 30088–30095, 2020.

[20] P. del Aguila Pla, S. Neumayer, and M. Unser, "Stability of image-reconstruction algorithms," *IEEE Trans. Comp. Imag.*, vol. 9, pp. 1–12, 2023.

[21] R. G. Gavaskar, C. D. Athalye, and K. N. Chaudhury, "On plug-and-play regularization using linear denoisers," *IEEE Trans. Image Process.*, vol. 30, pp. 4802–4813, 2021.

[22] E. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, and W. Yin, "Plug-and-play methods provably converge with properly trained denoisers," in *Proc. Int. Conf. Mach. Learn.*, pp. 5546–5557, 2019.

[23] S. Mukherjee, A. Hauptmann, O. Öktem, M. Pereyra, and C.-B. Schönlieb, "Learned reconstruction methods with convergence guarantees: a survey of concepts and applications," *IEEE Signal Process. Mag.*, vol. 40, no. 1, pp. 164–182, 2023.

[24] J.-C. Pesquet, A. Repetti, M. Terris, and Y. Wiaux, "Learning maximally monotone operators for image recovery," *SIAM J. Imag. Sci.*, vol. 14, no. 3, pp. 1206–1237, 2021.

[25] T. Tirer and R. Giryes, "Image restoration by iterative denoising and backward projections," *IEEE Trans. Image Process.*, vol. 28, pp. 1220–1234, 2019.

[26] T. Tirer and R. Giryes, "Back-projection based fidelity term for ill-posed linear inverse problems," *IEEE Trans. Image Process.*, vol. 29, pp. 6164–6179, 2020.

[27] T. Tirer and R. Giryes, "On the convergence rate of projected gradient descent for a back-projection based objective," *SIAM J. Imaging Sci.*, vol. 14, no. 4, pp. 1504–1531, 2021.

[28] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *IEEE Conf. Comp. Vis. Patt. Recogn*, pp. 5197–5206, 2015.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

[31] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *IEEE Conf. Comp. Vis. Patt. Recogn. (CVPR) Workshops*, 2017.

[32] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, L. Zhang, B. Lim, *et al.*, "NTIRE 2017 challenge on single image super-resolution: Methods and results," in *IEEE Conf. Comp. Vis. Patt. Recogn. (CVPR) Workshops*, 2017.

[33] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comp. Vis. (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[34] V. Boominathan, J. T. Robinson, L. Waller, and A. Veeraraghavan, "Recent advances in lensless imaging," *Optica*, vol. 9, no. 1, pp. 1–16, 2022.

[35] N. Antipa, G. Kuo, R. Heckel, B. Mildenhall, E. Bostan, R. Ng, and L. Waller, "Diffusercam: lensless single-exposure 3d imaging," *Optica*, vol. 5, no. 1, pp. 1–9, 2018.

[36] M. J. Cieślak, K. A. Gamage, and R. Glover, "Coded-aperture imaging systems: Past, present and future development–A review," *Radiation Measurements*, vol. 92, pp. 59–71, 2016.