

...

Demetrio Labate* and Heng Zhao

Integration of Model- and Learning-based Methods in Image Restoration

Abstract: Following the spectacular success of deep learning algorithms in image restoration tasks, there is growing interest in exploring how to combine the practical advantages of learning-based methods with the theoretical understanding that comes from model-based approaches. This article reviews recent strategies proposed in the literature that combine model- and learning-driven methods for problems in image restoration including image inpainting, denoising and deblurring.

Keywords: Convolutional neural networks, deblurring, deconvolution, denoising, image processing, inpainting, neural networks

PACS: 07.05, 84.35

1 Image Restoration

Image restoration aims to recover a degraded image x in a Hilbert space \mathcal{H} (e.g., $\mathcal{H} = L^2(\mathbb{R}^2)$) from the observation

$$y = Ax + \epsilon, \quad (1)$$

where A is a linear operator between Hilbert spaces modeling the forward problem and ϵ represents a noise term; in many problems, the noise term is modeled as a zero-mean Gaussian component. Depending on the selection of the operator A in (1), the image restoration problem reduces to one of the following cases.

- Denoising: $A = I$, where I is identity matrix.
- Deconvolution or deblurring: $Ax = h * x$, where h is know or unknown blur kernel and $*$ denotes the operation of convolution. When h is unknown, the restoration problem is called *blind deconvolution*.
- Inpainting: $A = S$ where S is degradation operator that replace a set of pixels by some values. For instance, S can be such that $S = 0$ for pixels in a compact

*Corresponding author: Demetrio Labate, University of Houston
Heng Zhao, University of Houston

set $K \subset \mathbb{R}^n$ and $S = 1$ for pixels are not in K . If the set K associated the degradation operator is unknown, the restoration problem is called *blind inpainting*. This problem is clearly more difficult than the *non-blind* image inpainting task where the locations of the removed regions are known.

A variety of ideas and algorithms have been developed to tackle image restoration tasks, including to classical methods from traditional signal processing such as morphological operators, partial differential equations and Fourier analysis. Many of these methods are well established and covered in classical signal processing textbooks [18, 25]. Modern image processing techniques typically formulate the problem of recovering x from (1) as an ill-posed inverse problem, whose solution require some form of regularization to constraint the solution space. In the language of Bayesian estimation, a solution \hat{x} of (1) can be obtained by solving a Maximum A Posteriori (MAP) problem

$$\hat{x} = \arg \max_x (\log p(y|x) + \log p(x)) \quad (2)$$

where $\log p(y|x)$ is the log-likelihood of the observation y and $\log p(x)$ is the prior of x which is independent of y . More explicitly, the solution (2) can be formulated as the optimization problem

$$\hat{x} = \arg \min_x \left(\frac{1}{2} \|y - Ax\|^2 + \lambda \Phi(x) \right) \quad (3)$$

where the solution \hat{x} minimizes an energy functional composed of the *fidelity term* $\frac{1}{2} \|y - Ax\|^2$ and the *regularization term* $\Phi(x)$. The regularization parameter $\lambda > 0$ establishes a trade-off between the fidelity term that penalizes the distance between the predicted data Ax and the measured data y , and the regularization term that enforces a desired property of the output. In practice, it may be challenging to design a good regularizer Φ for use in (3); it must mimic the negative log signal-prior while at the same time providing tractable optimization.

Generally, the methods proposed in the literature to solve (3) can be divided into two main categories: *model-based* and *learning-based methods*, the latter ones sometimes called *model-free methods*. Model-based methods aim to solve (3) directly using some optimization scheme that often involves a computationally-intensive and time-consuming iterative algorithm. One of the main advantage of these methods is that they typically provide theoretical performance guarantees. Learning-based method, on the other hand, are designed to learn the prior parameters through the optimization of a loss function on a training set consisting of a list of degraded-clean image pairs. Learning-based algorithms have revolutionized the field of signal processing in recent years, producing state-of-the-art performance in terms of image quality and computational efficiency. However, they usually require an extensive

training procedure and a large number of training data to achieve competitive performance. This may be impractical or impossible for some image restoration tasks. For example, remote sensing and medical imaging applications may have limited access to data due to cost, physical constraints, or privacy issues. Another drawback of learning-based algorithms is that, unlike model-based methods, they usually lack a performance guarantee; that is, it is not possible to state under what conditions the algorithm is guaranteed to converge to a solution. This limitation can be a significant disadvantage for applications that require reliable and consistent results.

Considering these challenges, there is growing interest in exploring how to combine the practical advantages of learning-based methods with the theoretical understanding that comes from model-based approaches [55]. In this article, we focus specifically on the effort made in the literature for combining model- and drive-based methods for the task of image restoration.

Among such hybrid methods, perhaps the most straightforward approach is to transform the iterative algorithm used to solve the optimization problem (3) into a neural network (NN) by taking advantage of the *compositionality* property of NNs, that is, the ability to define functions by composition. As a result, there is relatively simple procedure to implement virtually any iterative algorithms using a NN through an operation of *unrolling* or *unfolding*. This leads to a class of methods called *Algorithm Unrolling* that we discuss in Sec. 2.

Another approach consists in exploiting the variable splitting approach that can be used to solve (3) in order to ‘plug-in’ a NN trained as a denoiser in place of the proximal map of the regularizer Φ . This idea leads to another class of powerful algorithms, called *Plug-and-Play* methods, where a deep learning denoising prior is combined with a model-based optimization methods to solve the image restoration problem. These methods, that we call Plug-and-Play networks, are described in Sec. 3

Yet another approach consists in taking advantage of the expressive power of NNs while at the same time imposing a structural constraint based on model-driven principles such as, for instance, a sparsity-based condition applied to the learned filter. This approach has the effect of reducing the parameter space of a NN, hence lowering the requirement for training data, while at the same improving interpretability. This class of methods is discussed in Sec. 4.

The investigation of algorithms of image restoration from the integration of model- and learning-based principles is currently an extremely active field of research. While this article reviews the key advancements in this exciting domain, it makes no claim of being exhaustive as novel ideas and methods continue to emerge in the literature.

2 Deep Unrolling

Several successful iterative algorithms were proposed in the literature to solve (3) including the popular Iterative Shrinkage-Thresholding Algorithm (ISTA) [12], its fast version called Fast Iterative Shrinkage/Thresholding Algorithm (FISTA) [3], the Alternating Direction Method of Multipliers (ADMM) [5], the Half Quadratic Splitting (HQS) [17] and K-SVD [14] algorithms. While they are generally well understood and provide theoretical performance guarantees, they typically require many iterations to converge and can be computationally intensive and time-consuming.

Algorithm unrolling (or unfolding) was introduced by a seminal work of Gregor and LeCun [19] to speed up the convergence of such iterative algorithms by connecting them to multilayered NN architectures. In brief, algorithm unrolling consists in representing each iteration of the algorithm step as one layer of a NN. Concatenating these layers forms a deep NN so that feed-forward through the network is equivalent to running the iterative algorithm a finite number of times. The algorithm parameters such as the model and regularization parameters are transferred to the network parameters where they are learned by training the NN from real world training data. Hence, the trained NN can be interpreted as a parameter optimized algorithm. Since unrolled networks inherit the structure of a prior algorithm, they are highly parameter efficient and usually require less training data than generic NNs.

In recent years, several unrolled networks were introduced to provide unrolled versions of the most popular iterative algorithms for image restoration, including ADMM-CSNet [68], Learned Iterative Shrinkage/Thresholding Algorithm (LISTA) [19] and Deep K-SVD [52] into deep learning architecture. Such methods are often described as deep unrolling or deep unfolding.

For illustration purpose, let us consider the classical sparse reconstruction problem

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{R}^m} \frac{1}{2} \|\mathbf{y} - \mathbf{W}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1 \quad (4)$$

where we want to recover a sparse vector $\mathbf{x} \in \mathbb{R}^m$ given an input vector $\mathbf{y} \in \mathbb{R}^n$ and an overcomplete dictionary $\mathbf{W} \in \mathbb{R}^{n \times m}$, with $m > n$. The ISTA algorithm solves (4) by performing the iterative computation:

$$\mathbf{x}^{k+1} = \mathcal{S}_\lambda \left\{ \left(\mathbf{I} - \frac{1}{\mu} \mathbf{W}^T \mathbf{W} \right) \mathbf{x}^k + \frac{1}{\mu} \mathbf{W}^T \mathbf{y} \right\}, \quad k = 0, 1, \dots, K. \quad (5)$$

where $\mathbf{I} \in \mathbb{R}^{m \times m}$ is the identity matrix, μ is a parameter controlling the iterative step size and $\mathcal{S}_\lambda(\cdot)$ is the soft-thresholding operator given by

$$\mathcal{S}_\lambda(\mathbf{x}) = \text{sign}(\mathbf{x}) \cdot \max\{|\mathbf{x}| - \lambda, 0\}, \quad \mathbf{x} \in \mathbb{R}^m$$

To derive the unrolled network, we substitute parameters in (5) by

$$\mathbf{W}_t = \mathbf{I} - \frac{1}{\mu} \mathbf{W}^T \mathbf{W}, \quad \mathbf{W}_e = \frac{1}{\mu} \mathbf{W}^T$$

so that we can re-write (5) as an affine transformation followed by soft-thresholding

$$\mathbf{x}^{k+1} = \mathcal{S}_{\lambda_k}(\mathbf{W}_t^k \mathbf{x}^k + \mathbf{W}_e^k \mathbf{y}), \quad k = 0, 1, \dots, K. \quad (6)$$

Note that, in the expression above, we allow the parameters $\theta_k = (\mathbf{W}_t^k, \mathbf{W}_e^k, \lambda_k)$ to change at each iteration k , in contrast to the original iterative algorithm (4) where the iterative operator is predefined and unchanged during iteration. The expression (6) shows that each step of iteration (5) can be regarded as a single layer of a neural network. This is illustrated in Figure 1 where the k -th layer carries out the transformation $\mathbf{x}^{k+1} = \mathbf{H}(\mathbf{x}^k; \theta^k)$ and $\mathbf{H}(\mathbf{x}^k; \theta^k)$ can be identified with the right hand side of (6).

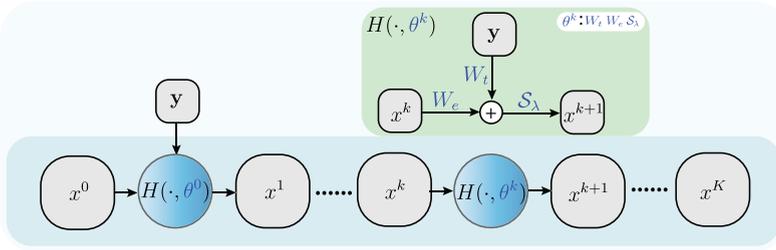


Fig. 1: General workflow of algorithm unrolling.

Hence, algorithm unrolling unrolls the iterations into a feed-forward NN where the network parameters are learned by an end-to-end approach. Namely, feeding an input $\mathbf{y}^i \in \mathbf{R}^n$, we generate an output $\tilde{\mathbf{x}}^K(\mathbf{y}^i; \cdot)$, which is the output of the K -th layer; we next compare it to the the ground truth data $\mathbf{x}^i \in \mathbf{R}^n$, $i = 1, 2, \dots, N$ so that we can learn the parameters $\theta^k = (\mathbf{W}_t^k, \mathbf{W}_e^k, \lambda_k)$, $k = 0, \dots, K$, by minimizing the loss function:

$$\ell(\theta^k, k = 0, \dots, K) = \frac{1}{N} \sum_{i=1}^N \left\| \tilde{\mathbf{x}}^K(\mathbf{y}^i; \theta^k, k = 0, \dots, K) - \mathbf{x}^i \right\|_2^2.$$

The NN can be trained through loss minimization using standard gradient-based learning techniques such as stochastic gradient descent. It has been shown that, using the algorithm unrolling strategy, the number of iterations required for the

algorithm to converge can be significantly reduced going from hundreds of iterations to tens of iterations [35].

Tab. 1: Summary of recent unrolling algorithms applied to image restoration.

Domain	Reference	Unrolled Algorithm	Year
Image denoising	Vu et al [61]	graph total variation (GTV)	2021
Image denoising	Scetbon et al [52]	ISTA for K-SVD	2021
Image denoising & demosaicking	Lecouat et al [31]	ISTA for CSC	2020
Image denoising & deblocking	Zhang et al [73]	restoration dynamics as RNN	2021
Image inpainting	Aberdam et al [1]	Fast ISTA	2021
Image inpainting	Sreter et al [57]	ISTA for CSC	2018
Image inpainting & denoising	Wei et al [65]	proximal gradient descent (PGD)	2022
Image deblurring	Schuler et [54]	alternating minimization	2015
Image deblurring	Li et al [32]	half quadratic splitting (HQS)	2019
Image deblurring	Bertocchi et al [4]	proximal interior point	2020
Image deblurring & denoising	Mou et al [43]	proximal gradient descent (PGD)	2022
Photon limited deblurring	Sanghvi et al [51]	plug-and-play	2022

2.1 Applications to image restoration

Due to its simplicity and computational power, algorithm unrolling has been very popular since its introduction and was applied to multiple problems including image restoration. The recent review by Monga et al. [42] provides an excellent review of such applications and Table 2 lists several unrolling algorithms for image restoration problems including problems of image denoising, inpainting and deblurring.

For instance, several authors proposed methods to combine sparse coding with the LISTA algorithm to perform image denoising, including Simon et al. [56] and Lecouat et al. [31] where self-similarity and sparsity principles were used as prior within a trainable framework. Using a similar idea, Zhang et al. [73] implemented a deep learning architecture imitating a nonlinear diffusion process for image denoising, while Scetbon et al. [52] unrolled the K-SVD algorithm to a sequence of differentiable layers for image denoising.

Several algorithm unrolling methods were also proposed for image deblurring. We recall that many classical image deblurring algorithms rely on sparsity constraints or total variation regularization which lead to computational intensive iterative numerical solutions [47, 46, 63]. Unrolled versions of such algorithms were shown to provide significantly faster implementations with often superior performance [54, 32, 43].

For image inpainting, Aberdam et al. [1] recently proposed a version of the LISTA algorithm, called Ada-LISTA, as an adaptive learned solver that combines as input both the signal and its corresponding dictionary and was successfully applied to the task of natural image inpainting.

We finally recall that algorithm unrolling was successfully applied to other image processing applications falling outside the scope of this paper, such as image super-resolution [64, 49, 11] and medical image reconstruction [75, 68].

3 Plug-and-Play Networks

The optimization problem (3) can be solved by *variable splitting*, where a new variable z is introduced to decouple the regularization term associated with the function Φ from the data fidelity term leading to the following constrained minimization problem

$$(\hat{x}, \hat{z}) = \arg \min_{x, z} \left(\frac{1}{2} \|y - Ax\|^2 + \lambda \Phi(z) \right) \quad \text{subject to } x = z. \quad (7)$$

Equation (7) suggests an algorithmic solution that alternates between separately estimating x and z . For instance, using the classical ADDM algorithm, the right hand side of equation (7) can be re-formulated as

$$\left(\frac{1}{2} \|y - Ax\|^2 + \lambda \Phi(z) + \frac{1}{2\eta} \|x - z + u\|^2 - \frac{1}{2\eta} \|u\|^2 \right), \quad (8)$$

where $\eta > 0$ is a penalty parameter, and then solved by alternating the optimization of x and z with gradient ascent of u . This leads to the iterative solution

$$\begin{aligned} x^{k+1} &= \text{prox}_g(z^k - u^k; \eta) \\ z^{k+1} &= \text{prox}_\Phi(x^{k+1} + u^k; \eta) \\ u^{k+1} &= u^k - (x^{k+1} - z^{k+1}) \end{aligned} \quad (9)$$

where $\text{prox}_\Phi(w; \eta) = \arg \min_x \left(\lambda \Phi(x) + \frac{1}{2\eta} \|x - w\|^2 \right)$ is the proximal map of Φ and $g(x) = \frac{1}{2} \|y - Ax\|^2$.

The key observation behind the Plug-and-Play (PnP) approach is to interpret the proximal mapping as a denoiser defined by a regularizer. Hence one can replace the proximal mapping prox_Φ in (9) by an off-the-shelf image denoising algorithm. The original approach by Venkatakrishnan et al. [60] replaced the proximal map with the popular Block-Matching and 3D Filtering (BM3D) denoising algorithm [10]. Several other denoising algorithms, including most notably deep NN denoisers

[40, 71], have been used as denoising prior and this approach was shown heuristically to produce better image restoration results than the regularization-based approach.

The same idea originally proposed by Venkatakrishnan et al. [60] applies to other variable splitting algorithms which deal with the fidelity term and regularization term separately and where, in particular, the regularization term only corresponds to a denoising subproblem. There is currently a whole family of PnP algorithms that, starting from an appropriate variable splitting algorithms, replace the proximal map with a denoising algorithm. In alternative to ADMM, other variable splitting algorithms commonly used within the PnP scheme include HQS, FISTA and primal-dual splitting (PDS) [7].

As observed in [13], the PnP approach “offers greater flexibility than the variational approach. First, a potential variational characterization of the a denoiser as proximal mapping might not be known. Second, and more importantly, any proximal mapping is in particular the gradient of some functional which excludes any denoiser that is not of gradient form. Hence PnP significantly extends the class of variational image reconstruction.” The same paper also demonstrated that PnP defines a stable and convergent regularization method.

3.1 Applications to image restoration

By exploiting the remarkable denoising power of deep learning architectures, PnP algorithms with deep NN denoisers have been shown to provide impressive performance in image restorations tasks including image denoising, inpainting and deblurring. Table 2 lists several recent PnP algorithms with a deep NN denoiser prior. The list includes the work of Zhang et al [71], the first method to propose a PnP algorithm with a deep NN. The application of this approach to denoising, deblurring and superresolution was shown to be very competitive against model-based methods like BM3D as well as simple deep learning schemes. The same authors later released an improved version of the PnP algorithm for image deblurring [72] and denoising [70]. Liu et al. [36] and Lai et al. [30] applied a similar idea to implement a PnP algorithms with a deep NN denoiser prior for denoising and inpainting in the context of hyperspectral images. Tirer et al. [59], Li et al. [33] and Nair et al. [44] developed PnPO algorithms with deep denoiser prior for problems of image inpainting, where they reported state-of-the art results.

Tab. 2: Summary of recent plug-and-play with DNN applied on image restoration.

Domain	Reference	Iterative algorithm	Year
Image denoising	Ryu et al [50]	ADMM & FBS	2019
Image denoising	Liu et al. [36]	ADMM	2021
Image denoising & inpainting	Lai et al. [30]	ADMM	2022
Image denoising & deblurring & SR	Zhang et al [71]	Half Quadratic Splitting (HQS)	2017
Image denoising & deblurring & SR	Zhang et al. [70]	Half Quadratic Splitting (HQS)	2020
Image inpainting & denoising	Tirer et al. [59]	ADMM	2018
Image inpainting	Li et al. [33]	Split Bregman Iteration (SBI)	2019
Image inpainting & deblurring & SR	Fermanian et al [16]	gradient of RED	2023
Image deblurring	Zhang et al [72]	HQS	2019
Image deblurring	Nan et al [45]	Variational EM	2020
Image deblurring	Kong et al [28]	regularization by denoing HQS	2022

We finally recall that, similar to unrolling algorithms, PnP algorithms with deep learning priors can be applied to a wide range of applications and were successfully applied to other image processing applications falling outside the scope of this paper, such as image super-resolution [20] and snapshot compressive imaging [69].

4 NNs with model-driven constraints

Another strategy for integrating model- and data-driven principles into algorithms for image restoration is to use model-driven constraints into the design of a deep NN.

The filters of a conventional NN are learned from data during training and this process may require a large number of training data depending on the complexity of the learning task. Clearly, using a clever initialization may be useful to facilitate the training process [23]. Nonetheless, learning the parameters of a deep NN typically requires a large number of training data, where this number is dependent on the number of learnable parameters of the NN.

Since images found in most applications include special structures which reduce the intrinsic dimensionality of the problem at hand, several methods have been proposed in the literature to bring physical or mathematical principles into the design of deep learning algorithms employed for image analysis or processing tasks. They include, for instance, methods to achieve invariance or covariance with respect to rotation and other groups [9, 67, 66], methods that incorporate a notion of multiscale to capture image features occurring at different scales [48, 15] as well as methods that try to enhance directional sensitivity of the convolutional filters of a CNN to improve their performance [24, 37]. Among such methods,

Structured Receptive Field Networks (SRFN) [24] provide an especially flexible approach for incorporating prior knowledge into the design of a CNN. The key idea in this approach is to represent convolutional filters as linear combinations from a pre-defined dictionary, so that only the coefficients of the representation are learned during training. By choosing a dictionary that is well adapted to the characteristics of natural images, it was shown that one can reduce the number of learnable parameters of a CNN so that it can be trained on fewer data or require fewer epochs.

While SRFNs were originally proposed to learn expressive feature representations in applications with limited training data, this idea was generalized and refined by the Geometric-Biased CNN (GBCNN) construction that was introduced by Schmallfuss et al. [53] for problems of blind image inpainting. The main novelties of the GBCNN approach include the design of a special class of convolutional filters targeted to problems of image analysis and the application of a sparsity constraint during training.

Below, we discuss in more detail the construction of GBCNNs and its application to problems of image restoration. In addition to reviewing the application to image inpainting problems already presented in [53], we also present some new applications to image denoising and deblurring.

4.1 Image restoration with GBCNNs

The GBCNN approach is motivated by the observation that sparsity-based methods such as curvelets and shearlets are especially effective in representing images due to their ability to capture edges and other critical features in images using representation elements that are very sensitive to orientations and are defined over multiple scales [29]. A Shearlet-based approach, in particular, was shown to provide state-of-the-art performance in problems of image inpainting by combining shearlets and ℓ_1 regularization [26, 22]. Hence the GBCNN approach adapts the SRFN main idea of expressing convolutional filters as a linear combinations from a pre-defined dictionary to combine the sparsity-inducing properties of the shearlet representation with the computational power of a CNN.

With respect SRFNs, the GBCNN approach introduces some novel ideas to better tackle image restoration problems. First, to ensure full expressive power at each layer of the NN, each convolutional filter W is expressed as a linear combination

$$W = \sum_{i=1}^L \alpha_i P_i,$$

where the basis elements P_i of the representation are taken from a *complete* dictionary D . Specifically, D is a frame or a Parseval frame (a frame being an overcomplete basis) consisting of $n \times n$ kernels, where n can be any prescribed size, e.g., $n = 3$ or $n = 5$. The frame condition was not enforced in the original SRFN approach, where the dictionary is a set of discrete Gaussian filters inspired by the theory of the scattering transform [6]. In contrast, the GBCNN design applies a rigorous mathematical theory, along with efficient implementation, to design a dictionary consisting of filters with compact support forming a frame or Parseval frame [2]. This construction also endows the dictionary with additional desirable geometric properties, such as directional sensitivity, that is very useful to capture relevant image features. The second main contribution is a sparsity constraint that reduces the number of learnable network parameters while at the same time imposing a form of geometric bias in the filter selection that does not reduce the overall expressive power of the network. We remark that, while pruning [41] - which reduces the number of kernels by eliminating nodes with small activation values - also achieves a sort of sparsification, the sparsity-inducing technique of the GBCNN is fundamentally different as it achieves reduction in the number of parameters before training, rather than afterwards.

4.1.1 Convolutional filter design

The filter dictionary used by the GBCNN is built using the theory of *compactly supported directional framelets* [2], a mathematical framework that provides an algorithmic procedure to build families of compactly supported filters with desirable mathematical properties, including the following:

- they form a frame or a Parseval frame (completeness);
- they produce discrete directional differentiation in all directions (directional sensitivity);
- most filters have relatively few non-zero entries (fast computation).

Figure 2 illustrates the 5×5 Parseval frame dictionary built from this approach that is used in [53] for applications in image inpainting and referred to as the Sparse Directional Parseval Frame (SDPF) dictionary. One hallmark of this dictionary is that it comprises many elements with high directional sensitivity; this property is critical for the efficient representation of edges in images, as shown by the theory of shearlets [21]. Combining this dictionary with the sparsity constraint, the filters of the GBCNN were shown to provide efficient representations of images dominated by objects with boundaries and this was a major motivating factor for the application

of this approach to image inpainting. We refer the interested reader to [53] for additional detail about the construction of this dictionary.

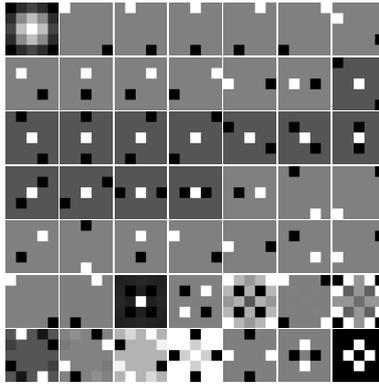


Fig. 2: 5×5 Sparse Directional Parseval Frame (SDPF) dictionary. Filters from top left to bottom right: one low-pass filter; twelve first-order finite difference filters; twelve second-order finite difference filters; 24 filters for the Parseval frame completion.

4.1.2 Network Architecture

While the GBCNN idea can be applied to virtually any network architecture, similarly to the original work [53], here we consider a simple multilayer CNN adapted from the Image Restoration CNN (IRCNN) from [8]. The IRCNN architecture is illustrated in Fig. 3 and consists of three blocks, i.e., feature extraction, transform, and image restoration, each one including convolutional layers. Namely, as shown in Fig 3, the feature extraction block consists of 2 convolutional layers of 64 channels each; the transform block is a single convolutional layer with 1×1 convolutional kernels; finally the image reconstruction block consists of 2 convolutional layers of 32 channels each.

To understand a fundamental difference of the GBCNN approach as compared with the conventional IRCNN training, it is useful to examine the convolutional filters learned by the two methods. Figure 4 displays side-by-side the convolutional filters in the first layer of the IRCNN (on the left) and GBCNN (on the right). As the figure shows, the filters learned by the GBCNN exhibit a more prominent geometric structure with sparse and highly directional filters, as compared to the IRCNN filters that are less sparse (fewer non-zero entries). The significance of this

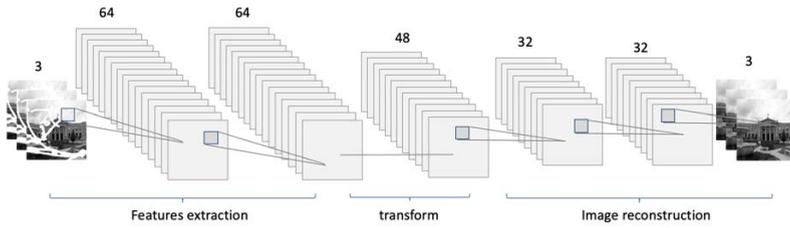


Fig. 3: Multilayered architecture used by the IRCNN [8] and by the GBCNN approach [53].

observation is that the GBCNN filters are potentially more efficient at capturing edges and highly directional image features.

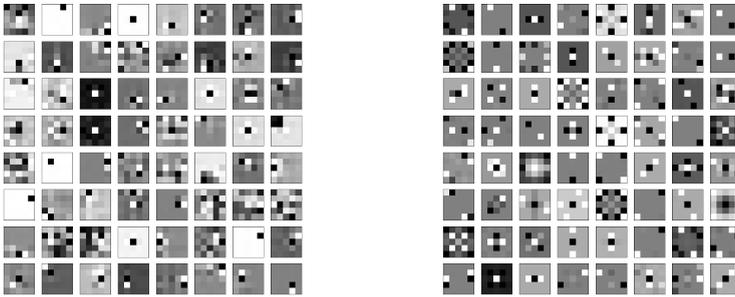


Fig. 4: The set of $64 \times 5 \times 5$ learned filters in the first layer of the CNN architecture in Fig. 2 resulting: on the left, from the conventional training strategy used by the IRCNN approach; on the right, from the GBCNN approach where filters are chosen from a predefined dictionary with a sparsity constraint.

4.2 Applications to problems of image restoration

The GBCNN approach was originally developed to solve the problem of *blind* image inpainting, where some portions of an image were removed and the locations of the removed regions are not known. Below, after reviewing the application of this approach to image inpainting, we apply the same algorithm also to problems of denoising and deblurring.

For our numerical applications, we used the publicly available dataset Places2 [74] consisting of 225,100 greyscale natural images of size 256×256 pixels. Places2 can be downloaded at <http://places2.csail.mit.edu/index.html>.

4.2.1 Image inpainting

We illustrate the performance of the GBCNN approach to blind image inpainting on the Places2 dataset. Following the original paper [53], corrupted images were generated by occluding a different fraction of the image area ranging from 5% to 25% of the total image area using handwritten characters. We considered three different versions of the inpainting experiment where the occluded area was replaced by solid white color or by a random solid grayscale color or by Gaussian noise. Image restoration performance was assessed using the standard Peak Signal to Noise Ratio (PSNR), defined by

$$PSNR = 10 \log_{10} \frac{\max_I^2}{MSE},$$

where \max_I is the maximum possible pixel value of the image and MSE is the mean squared error. The GBCNN was trained for 100 epochs using 221,000 images for training and 4,100 images for testing.

Table 3 compares the inpainting performance of the GBCNN approach against other state-of-the-art algorithms for blind image inpainting, including IRCNN [8], VCNet [62] and a shearlet approach [27]. VCNet is a deep learning algorithm that uses a significantly more complex network architecture than GBCNN, and is designed to first detect the occlusion location before inpainting. Unlike the other methods, the shearlet approach is a non-blind model-based inpainting method; it is included for baseline comparison since it relies on shearlet filters that are similar to those employed by the GBCNN. The table includes two different versions of the GBCNN methods, denoted as GBCNN₁ and GBCNN₂; in GBCNN₁, the first layer is a SDPF constrained receptive field layer and the remaining layers are conventional convolutional layers while in GBCNN₂, the two layers are SDPF constrained receptive field layer and the remaining layers are conventional convolutional layers.

In this tables as in all tables below, we adopt the convention that **bold font** indicates the best performance and underlined text indicates the second best performance.

Tab. 3: Image inpainting performance of different inpainting algorithms on images from the Places2 dataset with different occlusion sizes and different noise conditions. Reconstruction performance measured using PSNR (in dB) average over the image set.

Method	full test	00–05%	05–10%	10–15%	15–20%	20–25%
White mask						
Shearlet [27]	30.7005	34.5694	31.4638	29.5394	27.7532	25.4599
VCNet [62]	32.6120	37.8666	33.1650	30.9287	29.0408	27.0796
IRCNN [8]	32.5554	38.9941	33.0309	30.5410	28.2957	26.1550
GBCNN ₁ [53]	<u>32.9497</u>	<u>39.5763</u>	33.3030	<u>30.8849</u>	<u>28.6630</u>	<u>26.6679</u>
GBCNN ₂ [53]	33.0063	39.9796	<u>33.2698</u>	30.8267	28.5877	26.6221
Random grayscale mask						
VCNet	28.7741	33.9609	29.6332	27.9287	24.8558	22.6557
IRCNN	32.0852	32.0852	32.5079	30.2452	28.1787	26.2413
GBCNN ₁	32.4776	38.3913	32.9436	30.6258	28.5373	26.6047
GBCNN ₂	<u>32.2264</u>	<u>38.3673</u>	<u>32.5892</u>	<u>30.3000</u>	<u>28.2446</u>	<u>26.2721</u>
Random Gaussian mask						
VCNet	29.8666	35.7458	30.3789	27.8976	25.9770	24.0801
IRCNN	32.6010	<u>39.5739</u>	32.9097	30.3680	28.1927	26.1986
GBCNN ₁	<u>32.5193</u>	39.6120	<u>32.8116</u>	<u>30.2465</u>	<u>28.0567</u>	<u>26.0254</u>
GBCNN ₂	32.1891	39.2154	32.4960	29.9362	27.7528	25.7499

Table 3 shows that the GBCNN approach (including both GBCNN₁ and GBCNN₂) is the best performing approach overall when the occluded area is small. For larger occluded area, VCNet performs slightly better than GBCNN when the occluded area is replaced by white background. For larger occluded area replaced by random solid background GBCNN is still the best performing approach while; when the occluded area is replaced by Gaussian noise, IRCNN performs slightly better than GBCNN.

Table 4 reports the evaluation time per image for each inpainting method and the parameter count for each NN. The evaluation time was computed using a single NVIDIA Tesla V100 PCIe graphics card with 32 GB memory.

Tab. 4: Number of learnable parameters N_p and computing time per image t_I of several image inpainting methods. The computing time is the average value (in milliseconds) per image evaluated over images of size 256×256 pixels.

	Shearlets [27]	VCNet [62]	IRCNN [8]	GBCNN ₁ [53]	GBCNN ₂ [53]
N_p	–	3,789,892	172,113	<u>170,705</u>	80,593
t_I [ms]	29,147.001	17.460	2.539	2.453	<u>2.534</u>

The parameter count is the lowest for GBCNN₂ and the greatest for VCNet; this is expected since VCNet has a significantly more complex multilayer architecture and about 50 times more parameters than GBCNN₂. As a result, the computing time is also significantly higher for VCNet as compared to the GBCNN approach. The shearlet approach, implemented via the software ShearLab [27], is an iterative algorithm whose computing time is significantly higher (two orders of magnitude) than GBCNN and IRCNN.

Fig.5 illustrates the image inpainting performance of GBCNN and other methods on a representative image from the Places2 dataset using different types of background for the masked region.



Fig. 5: Image inpainting performance of multiple algorithms on representative image from the Places2 dataset using different types of background for the masked region.

4.2.2 Image Denoising and deblurring

We computed the performance of the GBCNN approach for image denoising on the Places2 dataset, where images were corrupted by adding white Gaussian noise with two different values of standard deviation, namely $\sigma = 25$ and $\sigma = 50$. To generate our GBCNN denoising model, the network was trained for 50 epochs, at which point the denoising performance was observed to stabilize. Similar to the inpainting experiment, we used 221,000 images from Places2 for training and 4,100 images for testing.

Results are reported in Table 5, where the performance of the GBCNN₁ architecture is compared against the deep learning algorithm IRCNN and the popular model based algorithm BM3D, that we have mentioned above. In addition to PSNR, we also include another widely used performance metric, the structural similarity index measure (SSIM), whose definition is based on a perception-based model that quantifies image degradation as perceived change in structural information.

The table shows that GBCNN₁ achieves the best denoising performance when $\sigma = 25$ and that IRCNN exhibits the best performance when $\sigma = 50$, even though, in the latter case, the GBCNN₁ performance is very similar to IRCNN.

Some additional numerical tests illustrating the denoising performance of the GBCNN approach are reported below in Sec. 5, where we present a broader comparison of denoising algorithms.

Tab. 5: Image denoising performance on the test images of the Places2 dataset corrupted by white Gaussian noise of standard deviation $\sigma = 25, 50$ using the algorithms IRCNN, GBCNN and BM3D.

	IRCNN [8]	BM3D [10]	GBCNN ₁ [53]
$\sigma = 25$			
PSNR	28.8813	<u>29.0030</u>	29.1927
SSIM	0.8389	<u>0.8449</u>	0.8492
$\sigma = 50$			
PSNR	26.0612	25.7576	<u>26.0038</u>
SSIM	0.7437	0.7320	<u>0.7415</u>

We also tested the GBCNN approach on a simple image deblurring task aiming at recovering an image degraded by a Gaussian blur kernel with a given standard deviation. As for the previous tasks, we trained and tested our algorithm on the Places2 dataset, using 221,000 images for training and 4,100 images for testing.

Table 6 reports the image deblurring performance on the test images of the Places2 dataset corrupted by a Gaussian blur kernel with $\sigma = 3$ and $\sigma = 5$. The performance of the GBCNN₁ algorithm is reported side-by-side the performance of the deep learning algorithm IRCNN and the model-based method BM3D. The BM3D algorithm exhibits the best performance in this experiment. We explain the poorer performance of IRCNN and GBCNN₁ with the fact that such methods are designed for denoising and inpainting tasks respectively.

Tab. 6: Image deblurring performance on the test images of the Places2 dataset corrupted by a Gaussian blur kernel with $\sigma = 3, 5$ using the algorithms IRCNN, RFCNN, BM3D.

	IRCNN [8]	BM3D [10]	GBCNN ₁ [53]
$\sigma = 3$			
PSNR	25.9243	26.4870	<u>25.9905</u>
SSIM	0.7676	0.8125	<u>0.7708</u>
$\sigma = 5$			
PSNR	23.1886	25.9923	<u>23.1918</u>
SSIM	0.6178	0.7565	<u>0.6179</u>

5 Numerical Experiments and Methods Comparison

It would be valuable and educational to systematically assess the performance of the various algorithms and techniques discussed in this article for image restoration tasks using publicly accessible datasets. Unfortunately, several practical challenges hinder this endeavor, primarily stemming from the limited availability of numerical codes that implement these methods. Moreover, these available codes are often tailored for specific conditions, necessitating major modifications to make them compatible with the same datasets in many cases.

Given these challenges, we will narrow our focus and set a more modest objective. Specifically, we will conduct a more limited comparison by applying some of the algorithms introduced above to a select set of image restoration experiments. In our comparisons, we will include at least one representative algorithm from each of the method categories we discussed in the sections above.

5.1 Image restoration experiments

To run our main denoising experiment, we considered as test images the publicly available BSD68 dataset [39], which is part of the Berkeley Segmentation Dataset (BSD) and consists of 68 natural images of various sizes; we also considered the smaller Set12 dataset, consisting of 12 widely used images of size 256×256 pixels, which is shown in Figure 6. The noisy inputs were obtained by adding white Gaussian noise with a constant standard deviation $\sigma = 25$ and $\sigma = 50$.



Fig. 6: The Set12 dataset includes 12 images of size 256×256 pixels.

We compared the performance of the following denoising algorithms on these two test sets: the model-based algorithm BM3D [10], the deep learning method IRCNN [8], the deep unrolling algorithms Learned K-SVD (LKSVD) [52], Group Sparse Coding (GroupSC) [31] and Convolutional Sparse Coding Network (CSCNet) [56], the HQS-based PnP method Deep Prior Image Restoration (DPIR) [70] and the NN with model-driven constraint GBCNN [53]. Results are reported in Table 7.

For these algorithms, we adopted the pre-trained denoising models that are available from the literature. We remark that these models were not trained on the same data. Specifically, the GroupSC model was computed using BSD400 (a subset of 400 images from the BSD) for training; the CSCNet model was computed using for training the Waterloo Exploration Dataset [38] and 432 images from BSD; the LKSVD model was computed using for training 432 images from BSD; the DPIR model [70] was computed by training on a large dataset including BSD400, 4,744 images from Waterloo Exploration Dataset, 900 images from DIV2K dataset [58], 2,750 images from Flickr2K dataset [34]; the IRCNN and GBCNN models were trained on the Places2 dataset, as described in Sec. 4.2. We also remark that, for all algorithms listed above, we adopted the default setting, as described in their original papers. For instance, the LKSVD algorithm requires to set as parameters

the patch size p , the number of dictionary elements m and the number of unfolding of network T ; we adopted $p = 64$, $m = 256$, $T = 7$ as in the default setting [52].

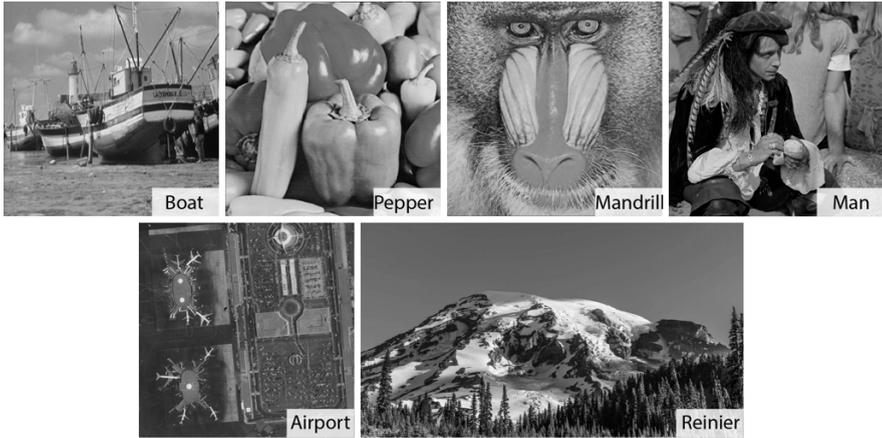


Fig. 7: The test images Boat, Pepper, Mandrill, Man, Airport and Reinier. The first four images have size 512×512 pixels; the last two images, have size 1024×1024 and 1920×1080 pixels, respectively.

Tab. 7: Image denoising performance (in PSNR) of the algorithms BM3D, LKSVD, GroupSC, CSCNet, DPIR, IRCNN, GBCNN₁ on the image sets Set12 and BSD68. Noisy images were generated by adding white Gaussian noise with standard deviation $\sigma = 25$ and $\sigma = 50$.

	BM3D [10]	LKSVD [52]	GroupSC [31]	CSCNet [56]	DPIR [70]	IRCNN [8]	GBCNN ₁ [53]
$\sigma = 25$							
Set12	29.89	30.22	<u>30.40</u>	30.25	30.94	29.94	29.99
BSD68	28.53	29.07	28.86	<u>29.10</u>	29.48	28.86	28.84
$\sigma = 50$							
Set12	26.66	27.04	27.09	<u>27.13</u>	27.90	26.80	26.72
BSD68	25.69	26.13	25.78	<u>26.22</u>	26.59	25.93	25.90

The results reported in Table 7 show that the DPIR approach exhibits the best denoising performance consistently over the noisy images we tested.

The same behavior is also observed in Table 8 which reports the denoising performance of the same algorithms on an additional set of 6 images displayed in Figure 7; these images have sizes ranging from 512×512 pixels to 1920×1080

pixels. The denoising performance of the different algorithms on four of the images considered in this table is further illustrated in Fig. 8.

Tab. 8: Image denoising performance (in PSNR) of the algorithms BM3D, GroupSC, CSCNet, DPIR, IRCNN, GBCNN₁ on 6 selected images with size 512×512 , 1024×1024 and 1920×1080 pixels. Noisy images were generated by adding white Gaussian noise with standard deviation $\sigma = 25$ and $\sigma = 50$.

	Boat _{512²}	Pepper _{512²}	Mandrill _{512²}	Man _{1024²}	Airport _{1024²}	Rainier _{1920×1080}
$\sigma = 25$						
BM3D [10]	29.7876	31.8127	25.3886	30.1486	28.2283	27.8278
GroupSC [31]	<u>30.1580</u>	32.1986	<u>25.8790</u>	<u>30.5631</u>	<u>28.5223</u>	<u>28.3846</u>
CSCNet [56]	30.1005	<u>32.2304</u>	25.7884	30.5106	28.4785	28.3414
DPIR [70]	30.5768	32.5809	26.1142	30.8268	28.8338	28.5624
IRCNN [8]	30.1065	32.0728	25.7508	30.2858	28.3953	28.1323
GBCNN ₁ [53]	30.0787	32.0430	25.6892	30.2336	28.3691	28.1530
$\sigma = 50$						
BM3D	26.7826	28.9796	22.3993	27.3421	25.8359	24.3002
GroupSC	27.0765	29.4308	<u>22.7979</u>	27.6699	26.1202	<u>24.8526</u>
CSCNet	<u>27.1820</u>	<u>29.6683</u>	22.7530	<u>27.7537</u>	<u>26.241</u>	24.8251
DPIR	27.6556	30.1341	23.1263	28.0471	26.6003	25.0643
IRCNN	26.9400	29.1987	22.4379	27.3973	25.9754	24.5860
GBCNN ₁	26.9709	29.1677	22.3890	27.3928	25.9723	24.6105

We attribute the very competitive performance of the DPIR approach in large part to the higher complexity of its model as compared with the other methods we considered. This observation is supported by Table 9 comparing the number of trainable parameters of the different algorithms. The table shows that DPIR has the highest number, i.e., 32,638,656 parameters, while GBCNN₁ has the lowest number, i.e., 170,705; BM3D has no learnable parameters being a model-based method. The data reported in the tables suggest that the DPIR model is able to take advantage of the complexity of its architecture; as indicated above, the DPIR denoising model was computed using the largest training set among all methods considered.

Table 9 also compares the computing time of the different methods. To have a fair comparison, computing time was evaluated for all methods using the Carya computing cluster from the University of Houston equipped with an Intel Xeon G 6252 CPU with 16 GB of RAM; no GPU acceleration was used for this computation.

Tab. 9: Number of learnable parameters N_p and computing time per image t_I of several image denoising methods. The computing time is the average value (in milliseconds) per image evaluated over images of size 256×256 pixels.

	BM3D [10]	LKSVD [52]	GroupSC [31]	CSCNet [56]	DPIR [70]	IRCNN [8]	GBCNN ₁ [53]
N_p	-	35,138	<u>68,434</u>	63,700	32,638,656	172,113	170,705
t_I [ms]	1,522.390	460.383	19,771.717	1,006.507	1,770.780	<u>223.929</u>	219.191

The table shows that, consistently with the parameter count, DPIR has by far the longest computing time; additionally, the PnP design involves an iterative implementation which increases computing time. The unrolled algorithms LKSVD, GroupSC and CSCNet are faster than DPIR. GBCNN₁ exhibits the lowest computing time being both non-iterative and light-weight; its computing time is about one order of magnitude less than DPIR.

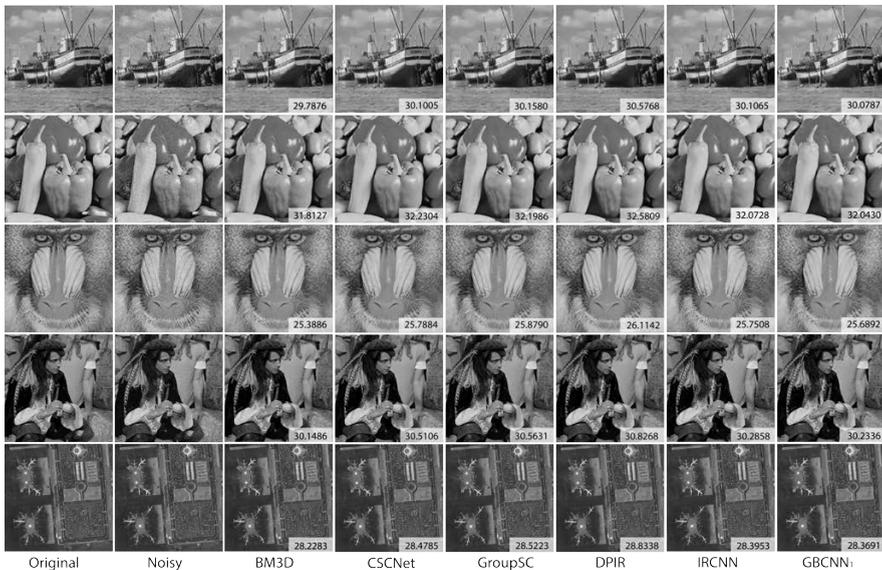


Fig. 8: Image denoising performance (in PSNR) on the Boat, Peppers, Mandrill, Man, Airport images corrupted by white Gaussian noise of standard deviation $\sigma = 25$ using the BM3D, CSCNet, GroupSC, DPIR, IRCNN and GBCNN₁ algorithms.

Tab. 10: Image inpainting performance (in PSNR) on 100 images from Places2 dataset using the algorithms ISTA, FISTA, ada-LISTA, IDBP, IRCNN, GBCNN₁ and VCNet. Images were corrupted either by occlusion with handwriting or by setting 50% of the randomly selected pixels to zero.

	ISTA [12]	FISTA [3]	ada-LISTA [1]	IDBP [59]	IRCNN [8]	GBCNN ₁ [53]	VCNet [62]
Handwriting	20.78	21.76	21.60	<u>26.86</u>	26.15	26.66	27.07
Pixel-wise	24.61	26.96	27.41	31.21	31.87	<u>31.74</u>	-

We also run two image inpainting experiments using as test set 100 images from Places2. In the first experiment, the corrupted images were generated by occluding 20% to 25% of the area using a white solid mask (consisting of handwriting), as in one of our experiments from Sec. 4.2. In our second experiment, the corrupted images were generated by randomly choosing 50% of the image pixels and setting those to 0. The results of these experiments are reported in Table 10 where we compared the inpainting performance of the following image inpainting algorithms: the model-based methods Iterative Shrinkage Thresholding Algorithm (ISTA) [12] and Fast-ISTA (FISTA) [3], the deep unrolling algorithm Ada-Lista [1], the PnP algorithm IDBP [59], the NN with model-driven constraints GBCNN₁ and deep learning algorithms IRCNN [8] and VCNet [62].

We remark that, out of the methods listed above, only IRCNN, GBCNN₁, VCNet are designed for blind image inpainting; ISTA, FISTA, Ada-Lista, and IDBP are non-blind methods, hence we also provided the location of the mask when the methods were run. We also note that Ada-Lista and IDBP are not designed for large area blind image inpainting and the training and testing are patch-based; they were trained on patches extracted from BSD400, according to their original formulation. By contrast, IRCNN, VCNet and GBCNN₁ are designed to inpaint possibly large image areas and are trained on whole images; they were trained on images from Places2, as described in Sec. 4.2.



Fig. 9: Image inpainting performance (in PSNR) on selected images from Places2 with white solid masks.

Table 10 shows that VCNet exhibits the best performance in the experiment with handwriting mask, with IDBP (a non-blind algorithm) exhibiting the second best performance, slightly better than GBCNN_1 ; in the experiment with pixel-wise mask, IRCNN exhibits the best performance and GBCNN_1 the second best one. We notice that GBCNN_1 has consistently good performance in both large areas blind inpainting and pixel-wise inpainting. Figures 9 and 10 show the application of these image inpainting algorithms on some representative images from the Places2 and BSD68 datasets.



Fig. 10: Image inpainting performance (in PSNR) on a selected image from BSD68 with pixel-wise random mask.

5.1.1 Data reproducibility

Python code and pre-trained models for the inpainting and denoising applications of the GBCNN and IRCNN approaches that we reported above are found at <https://github.com/zhaoheng001/GBCNN>.

For completeness, we list below the links to the numerical codes and, if available, the pre-trained models that we used to run the other algorithms we reported:

BM3D (Python): <https://pypi.org/project/bm3d/>

CSCNet (Python): <https://github.com/drorsimon/CSCNet>

DPIR (Python): <https://github.com/cszn/DPIR>

GroupSC (Python): <https://github.com/bruno-31/groupsc>

IDBP (Matlab): <https://github.com/tomtirer/IDBP>

ISTA, FISTA, Ada-Lista (Python): <https://github.com/aaberdam/AdaLISTA>

LKSVD (Python): <https://github.com/meyerscetbon/Deep-K-SVD>

VCNet (Python): https://github.com/shepnerd/blindinpainting_vcnet

5.2 Conclusion

The development of algorithms that integrate learning- and model-driven principles is currently a topic of intense research activity due to the promise of combining high-computational efficiency and theoretical performance guarantees. As we illustrated in the pages above, several image restoration strategies integrating learning and models have been proposed in the literature including most notably algorithm unrolling, plug-and-play networks and NNs with model-driven constraints. Such algorithms have been shown to provide state-of-the-art performance in problems of image denoising, deblurring and inpainting, often outperforming more conventional deep learning strategies. In fact, the advantages of such emerging hybrid algorithmic strategies go beyond image restoration performance and computational cost, as such methods can also provide performance guarantees, improved interpretability and reduced computational complexity, as compared to deep learning algorithms.

To better illustrate the properties of these algorithms, we presented several numerical tests on problems of image denoising and inpainting. Our numerical experiments confirm that these methods not only provide state-of-the-art performance but are very competitive against other state-of-the-art methods in terms of computational complexity and computational cost.

Acknowledgment: The authors acknowledge the use of the Carya Cluster and the advanced support from the Research Computing Data Core at the University of Houston to carry out part of the research presented here.

References

- [1] ABERDAM, A., GOLTS, A., AND ELAD, M. Ada-lista: Learned solvers adaptive to varying models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 12 (2021), 9222–9235.
- [2] ATREAS, N., KARANTZAS, N., PAPADAKIS, M., AND STAVROPOULOS, T. On the design of multi-dimensional compactly supported parseval framelets with directional characteristics. *Linear Algebra and its Applications* 582 (2019), 1–36.
- [3] BECK, A., AND TEOULLE, M. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE transactions on image processing* 18, 11 (2009), 2419–2434.

- [4] BERTOCCHI, C., CHOUZENOUX, E., CORBINEAU, M.-C., PESQUET, J.-C., AND PRATO, M. Deep unfolding of a proximal interior point method for image restoration. *Inverse Problems* 36, 3 (2020), 034005.
- [5] BOYD, S., PARIKH, N., CHU, E., PELEATO, B., ECKSTEIN, J., ET AL. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* 3, 1 (2011), 1–122.
- [6] BRUNA, J., AND MALLAT, S. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1872–1886.
- [7] CHAMBOLLE, A., AND POCK, T. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision* 40 (2011), 120–145.
- [8] CHAUDHURY, S., AND ROY, H. Can fully convolutional networks perform well for general image restoration problems? In *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA) (2017)*, IEEE, pp. 254–257.
- [9] COHEN, T., AND WELLING, M. Group equivariant convolutional networks. In *International conference on machine learning (2016)*, PMLR, pp. 2990–2999.
- [10] DABOV, K., FOI, A., KATKOVNIK, V., AND EGIAZARIAN, K. Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Transactions on image processing* 16, 8 (2007), 2080–2095.
- [11] DARDIKMAN-YOFFE, G., AND ELДАР, Y. C. Learned SPARCOM: unfolded deep super-resolution microscopy. *Optics express* 28, 19 (2020), 27736–27763.
- [12] DAUBECHIES, I., DEFRISE, M., AND DE MOL, C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 57, 11 (2004), 1413–1457.
- [13] EBNER, A., AND HALTMEIER, M. Plug-and-play image reconstruction is a convergent regularization method. *arXiv preprint arXiv:2212.06881* (2022).
- [14] ELAD, M., AND AHARON, M. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing* 15, 12 (2006), 3736–3745.
- [15] FAN, Y., LIN, L., YING, L., AND ZEPEDA-NÚÑEZ, L. A multiscale neural network based on hierarchical matrices. *Multiscale Modeling & Simulation* 17, 4 (2019), 1189–1213.
- [16] FERMANIAN, R., LE PENDU, M., AND GUILLEMOT, C. Pnp-reg: Learned regularizing gradient for plug-and-play gradient descent. *SIAM Journal on Imaging Sciences* 16, 2 (2023), 585–613.
- [17] GEMAN, D., AND YANG, C. Nonlinear image recovery with half-quadratic regularization. *IEEE transactions on Image Processing* 4, 7 (1995), 932–946.

- [18] GONZALES, R. C., AND WINTZ, P. *Digital image processing*. Addison-Wesley Longman Publishing Co., Inc., 1987.
- [19] GREGOR, K., AND LECUN, Y. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning* (2010), pp. 399–406.
- [20] GU, S., TIMOFTE, R., AND VAN GOOL, L. Integrating local and non-local denoiser priors for image restoration. In *2018 24th International Conference on Pattern Recognition (ICPR)* (2018), IEEE, pp. 2923–2928.
- [21] GUO, K., AND LABATE, D. Optimally sparse multidimensional representation using shearlets. *SIAM journal on mathematical analysis* 39, 1 (2007), 298–318.
- [22] GUO, K., LABATE, D., AND AYLLON, J. P. R. Image inpainting using sparse multiscale representations: Image recovery performance guarantees. *Applied and Computational Harmonic Analysis* 49, 2 (2020), 343–380.
- [23] HE, K., ZHANG, X., REN, S., AND SUN, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (2015), pp. 1026–1034.
- [24] JACOBSEN, J.-H., VAN GEMERT, J., LOU, Z., AND SMEULDERS, A. W. Structured receptive fields in cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 2610–2619.
- [25] JAIN, A. K. *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989.
- [26] KING, E. J., KUTYNIOK, G., AND ZHUANG, X. Analysis of inpainting via clustered sparsity and microlocal analysis. *Journal of mathematical imaging and vision* 48, 2 (2014), 205–234.
- [27] KING, E. J., KUTYNIOK, G., AND ZHUANG, X. Analysis of inpainting via clustered sparsity and microlocal analysis. *Journal of mathematical imaging and vision* 48, 2 (2014), 205–234.
- [28] KONG, S., WANG, W., FENG, X., AND JIA, X. Deep red unfolding network for image restoration. *IEEE Transactions on Image Processing* 31 (2021), 852–867.
- [29] KUTYNIOK, G., AND LABATE, D. *Shearlets: Multiscale analysis for multivariate data*. Springer, 2012.
- [30] LAI, Z., WEI, K., AND FU, Y. Deep plug-and-play prior for hyperspectral image restoration. *Neurocomputing* 481 (2022), 281–293.
- [31] LECOAT, B., PONCE, J., AND MAIRAL, J. Fully trainable and interpretable non-local sparse models for image restoration. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16* (2020), Springer, pp. 238–254.

- [32] LI, Y., TOFIGHI, M., GENG, J., MONGA, V., AND ELDAR, Y. C. Efficient and interpretable deep blind image deblurring via algorithm unrolling. *IEEE Transactions on Computational Imaging* 6 (2020), 666–681.
- [33] LI, Z., AND WU, J. Learning deep CNN denoiser priors for depth image inpainting. *Applied Sciences* 9, 6 (2019), 1103.
- [34] LIM, B., SON, S., KIM, H., NAH, S., AND LEE, K. M. Enhanced deep residual networks for single image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (July 2017).
- [35] LIU, J., AND CHEN, X. ALISTA: Analytic weights are as good as learned weights in LISTA. In *International Conference on Learning Representations (ICLR)* (2019).
- [36] LIU, Y.-Y., ZHAO, X.-L., ZHENG, Y.-B., MA, T.-H., AND ZHANG, H. Hyperspectral image restoration by tensor fibered rank constrained optimization and plug-and-play regularization. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), 1–17.
- [37] LUAN, S., CHEN, C., ZHANG, B., HAN, J., AND LIU, J. Gabor convolutional networks. *IEEE Transactions on Image Processing* 27, 9 (2018), 4357–4366.
- [38] MA, K., DUANMU, Z., WU, Q., WANG, Z., YONG, H., LI, H., AND ZHANG, L. Waterloo exploration database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing* 26, 2 (2016), 1004–1016.
- [39] MARTIN, D., FOWLKES, C., TAL, D., AND MALIK, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001* (2001), vol. 2, IEEE, pp. 416–423.
- [40] MEINHARDT, T., MOLLER, M., HAZIRBAS, C., AND CREMERS, D. 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.
- [41] MOLCHANOV, P., TYREE, S., KARRAS, T., AILA, T., AND KAUTZ, J. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations* (2016).
- [42] MONGA, V., LI, Y., AND ELDAR, Y. C. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine* 38, 2 (2021), 18–44.
- [43] MOU, C., WANG, Q., AND ZHANG, J. Deep generalized unfolding networks for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 17399–17410.
- [44] NAIR, P., AND CHAUDHURY, K. N. Plug-and-play regularization using linear solvers. *IEEE Transactions on Image Processing* 31 (2022), 6344–6355.

- [45] NAN, Y., QUAN, Y., AND JI, H. Variational-EM-based deep learning for noise-blind image deblurring. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 3626–3635.
- [46] PAN, J., HU, Z., SU, Z., AND YANG, M.-H. l_0 -regularized intensity and gradient prior for deblurring text images and beyond. *IEEE transactions on pattern analysis and machine intelligence* 39, 2 (2016), 342–355.
- [47] PAN, J., SUN, D., PFISTER, H., AND YANG, M.-H. Deblurring images via dark channel prior. *IEEE transactions on pattern analysis and machine intelligence* 40, 10 (2017), 2315–2328.
- [48] PELT, D. M., AND SETHIAN, J. A. A mixed-scale dense convolutional neural network for image analysis. *Proceedings of the National Academy of Sciences* 115, 2 (2018), 254–259.
- [49] RIEGLER, G., RÜTHER, M., AND BISCHOF, H. Atgv-net: Accurate depth super-resolution. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part III 14* (2016), Springer, pp. 268–284.
- [50] RYU, E., LIU, J., WANG, S., CHEN, X., WANG, Z., AND YIN, W. Plug-and-play methods provably converge with properly trained denoisers. In *International Conference on Machine Learning* (2019), PMLR, pp. 5546–5557.
- [51] SANGHVI, Y., GNANASAMBANDAM, A., AND CHAN, S. H. Photon limited non-blind deblurring using algorithm unrolling. *IEEE Transactions on Computational Imaging* 8 (2022), 851–864.
- [52] SCETBON, M., ELAD, M., AND MILANFAR, P. Deep k-SVD denoising. *IEEE Transactions on Image Processing* 30 (2021), 5944–5955.
- [53] SCHMALFUSS, J., SCHEURER, E., ZHAO, H., KARANTZAS, N., BRUHN, A., AND LABATE, D. Blind image inpainting with sparse directional filter dictionaries for lightweight cnns. *Journal of Mathematical Imaging and Vision* 65, 2 (2023), 323–339.
- [54] SCHULER, C. J., HIRSCH, M., HARMELING, S., AND SCHÖLKOPF, B. Learning to deblur. *IEEE transactions on pattern analysis and machine intelligence* 38, 7 (2015), 1439–1451.
- [55] SHLEZINGER, N., WHANG, J., ELДАР, Y. C., AND DIMAKIS, A. G. Model-based deep learning. *Proceedings of the IEEE* (2023).
- [56] SIMON, D., AND ELAD, M. Rethinking the CSC model for natural images. *Advances in Neural Information Processing Systems* 32 (2019).
- [57] SRETER, H., AND GIRYES, R. Learned convolutional sparse coding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), IEEE, pp. 2191–2195.
- [58] TIMOFTE, R., GU, S., WU, J., VAN GOOL, L., ZHANG, L., YANG, M.-H., HARIS, M., ET AL. Ntire 2018 challenge on single image super-resolution:

- Methods and results. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (June 2018).
- [59] TIRER, T., AND GIRYES, R. Image restoration by iterative denoising and backward projections. *IEEE Transactions on Image Processing* 28, 3 (2018), 1220–1234.
- [60] VENKATAKRISHNAN, S. V., BOUMAN, C. A., AND WOHLBERG, B. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing* (2013), IEEE, pp. 945–948.
- [61] VU, H., CHEUNG, G., AND ELDAR, Y. C. Unrolling of deep graph total variation for image denoising. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), IEEE, pp. 2050–2054.
- [62] WANG, Y., CHEN, Y.-C., TAO, X., AND JIA, J. Vcnet: A robust approach to blind image inpainting. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16* (2020), Springer, pp. 752–768.
- [63] WANG, Y., YANG, J., YIN, W., AND ZHANG, Y. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences* 1, 3 (2008), 248–272.
- [64] WANG, Z., LIU, D., YANG, J., HAN, W., AND HUANG, T. Deep networks for image super-resolution with sparse prior. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (December 2015).
- [65] WEI, X., VAN GORP, H., GONZALEZ-CARABARIN, L., FREEDMAN, D., ELDAR, Y. C., AND VAN SLOUN, R. J. Deep unfolding with normalizing flow priors for inverse problems. *IEEE Transactions on Signal Processing* 70 (2022), 2962–2971.
- [66] WEILER, M., AND CESA, G. General $e(2)$ -equivariant steerable CNNs. *Advances in neural information processing systems* 32 (2019).
- [67] WEILER, M., GEIGER, M., WELLING, M., BOOMSMA, W., AND COHEN, T. S. 3d steerable CNNs: Learning rotationally equivariant features in volumetric data. *Advances in Neural Information Processing Systems* 31 (2018).
- [68] YANG, Y., SUN, J., LI, H., AND XU, Z. ADMM-CSNet: A deep learning approach for image compressive sensing. *IEEE transactions on pattern analysis and machine intelligence* 42, 3 (2018), 521–538.
- [69] YUAN, X., LIU, Y., SUO, J., AND DAI, Q. Plug-and-play algorithms for large-scale snapshot compressive imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 1447–1457.

- [70] ZHANG, K., LI, Y., ZUO, W., ZHANG, L., VAN GOOL, L., AND TIMOFTE, R. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 10 (2021), 6360–6376.
- [71] ZHANG, K., ZUO, W., GU, S., AND ZHANG, L. Learning deep CNN denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 3929–3938.
- [72] ZHANG, K., ZUO, W., AND ZHANG, L. Deep plug-and-play super-resolution for arbitrary blur kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 1671–1681.
- [73] ZHANG, X., LU, Y., LIU, J., AND DONG, B. Dynamically unfolding recurrent restorer: A moving endpoint control method for image restoration. *arXiv preprint arXiv:1805.07709* (2018).
- [74] ZHOU, B., LAPEDRIZA, A., KHOSLA, A., OLIVA, A., AND TORRALBA, A. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence* 40, 6 (2017), 1452–1464.
- [75] ZHU, B., LIU, J. Z., CAULEY, S. F., ROSEN, B. R., AND ROSEN, M. S. Image reconstruction by domain-transform manifold learning. *Nature* 555, 7697 (2018), 487–492.