A Deep Network for Reconstructing Images from Undersampled Poisson data

Antonio Lorente Mur, F Peyrin, Nicolas Ducros

A Deep Network for Reconstructing Images from Undersampled Poisson data

A Lorente Mur, F Peyrin, *Member, IEEE,* N. Ducros, *Member, IEEE,*

***Abstract*—We consider the reconstruction of an image from a sequence of a few linear measurements corrupted by Poisson noise. This generic problem has many biomedical applications, such as computerized tomography, positron emission tomography, and optical microscopy. Here, we focus on a computational optics problem where the set-up acquires some coefficients of the Hadamard transform of the image of the scene.**

We formalize this problem in a Bayesian setting where we estimate the missing Hadamard coefficients from those acquired. Then, we propose a deep-learning network that consists of two fully connected layers (FCLs) that map data from the measurement domain to the image domain, followed by convolutional layers that act in the image domain. On the one hand, we set the FCLs so that they compute the best linear solution of the problem. While the first FCL denoises the raw measurements, the second FCL completes the missing measurements from the denoised measurements. The convolutional layers undergo learning through a training phase.

We also describe a framework for training the network in the presence of Poisson noise. In particular, our approach includes an estimation of the image intensity, together with a normalization scheme that allows varying noise levels to be handled during training. We compare our network to linear reconstructors and to network variants that do not address the noise issue at all, or that address it implicitly.

Finally, we present results from simulated and experimental acquisitions, considering varying noise levels. Our network yields higher reconstruction peak signal-to-noise ratios in scenarios where the actual noise level is higher than that expected and used during the training phase.

***Index Terms*—Image reconstruction, Poisson noise, undersampled data, deep learning, convolutional neural network, computational optics, single-pixel imaging.**

I. Introduction

IMAGE reconstruction from noisy measurements where the number of unknowns is larger than the number of measurements is a generic problem that has several applications in computational imaging. While such inverse problems have long benefited from the compressed sensing theory, which exploits sparsity priors, recent advances in deep learning have been revolutionizing the field [1], [2], [3]. In particular, convolutional neural networks have shown great success at solving computed tomography problems, either by learning a direct inverse mapping [4], or through the use of adversarial neural networks [5]. Much effort is currently devoted to bridging the gap between more traditional approaches for image reconstruction and the deep-learning-based approach to solving inverse problems. Convolutional neural networks have been used to serve as a prior to solve ill-posed inverse problems. As such they have been used as a prior for manifolds of natural images [6], as a projector for sets of natural images [7], or as sparsifying transforms [8]. These priors are then plugged into an unrolled optimisation algorithm [9] or used as a series expansion, like the Neumann series [10]. This trend is very strong in computational optics [11], where deep learning is also used to optimize the design of experiments [12].

Single-pixel imaging is an extreme configuration of computational optics, where a single point detector is used to recover an image [13]. Since the seminal work by Duarte and coworkers [14], single-pixel imaging has been successfully applied to fluorescence microscopy [15], hyperspectral imaging [16], [17], diffuse optical tomography [18], image-guided surgery [19], short-wave infrared imaging [20], and imaging through scattering media [21]. Single-pixel measurements can be modeled as dot products between an image and some two-dimensional functions that are implemented through a spatial light modulator [13]. To limit acquisition times, it is highly desirable to reduce the number of light patterns, which leads to an undetermined inverse problem.

In the field of single-pixel imaging, deep learning has been used to unmix the fluorescence intensity and lifetime from time-resolved measurements [22], [23]. In [24], they proposed a convolutional auto-encoder for single-pixel image reconstruction imaging that outperformed compressed sensing approaches. This network directly maps the measurement vector to the desired image, using a fully connected layer (FCL) followed by convolutional layers. In [21], a similar network was used for measurements with very low signal-to-noise ratio, while in [25], a network was introduced where the first FCL computes the conditional expectation of the image for a given set of noiseless measurements.

Focusing on single-pixel imaging, we consider that our data to be corrupted by a Poisson noise model. Traditionally, inverse problems with data corrupted by Poisson noise are tackled using variance stabilizing transforms, such as the Anscombe transform [26], followed by a Wiener filter [27]. However, the resulting image can be blurred, and in particular for undersampled data. More recent alternatives solve this issue by exploiting statistical or handmade image priors [28], [29]. Solving the resulting optimization problem is usually prohibitive for real-time applications. Alternatively, the Poisson-Gaussian unbiased risk estimator-linear expansion

All authors are with the University of Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM Saint-Etienne, CREATIS CNRS UMR 5220, Inserm U1206, F-69621, Lyon, France

We would like to thank J.F.P.J Abascal for comments on the manuscript and fruitful discussions on deep learning. This work was supported by the French National Research Agency (ANR), under Grant ANR-17-CE19-0003 (ARMONI Project), and performed within the framework of the LABEX PRIMES (ANR-11-LABX-0063) of Université de Lyon.

of thresholds [30], [31] seeks a solution as an optimal linear parameterization of Wiener filters, followed by a domain-transform thresholding. Although this approach is much faster, it still takes several seconds to reconstruct an image. In this regard, deep networks are ideal candidates, due to their rapid evaluation. As indicated by [32], however, the robustness of a neural network to various noise conditions is still not fully understood. For data corrupted by signal-dependent noise, such as Poisson noise, this can be a severe limitation.

A. Contribution

We interpret the reconstruction from undersampled data as a completion problem, where the missing data needs to be estimated before reconstruction. Preliminary results were shown for this idea in [25], and these are extended here.

1) Completion in the presence of noise: Inspired by [33], we adopt a Bayesian framework where we use deep learning to compute a conditional expectation. Under Gaussian assumptions, we derive an analytical solution that is the best linear solution to our problem. We interpret this as denoising of the raw measurements, followed by completion of the missing measurements. We propose to implement this as the FCL of a convolutional neural network, where only the convolution layers are trained to mitigate the nonGaussian distribution of the data in the image domain .

2) Training a network with Poisson noise: We describe a machinery that allows for a network to be trained in the presence of Poisson noise. The contribution here is two-fold. First, we propose an approximation for the covariance matrix of the measurement. This approximation is required when the image intensity is not known. Secondly, we provide a normalization scheme that allows raw data with different orders of magnitudes to be considered, which is mandatory for realistic scenarios where the image intensity is not known.

3) Experimental data: We validate our approach by reconstruction of an experimental dataset that we acquired with varying integration time and light flux. Upon acceptation of this paper, the datasets will be made available.

4) Open source software: Upon acceptation of this paper, implementations of our reconstruction methods in Matlab (SPIRiT [34]; only for the linear methods) and Python (SPYRiT [35] toolbox) will be made available.

B. Organization of the paper

In Section II, we model an undersampled computational (compressive) acquisition device and describe the associated reconstruction problem. In Section III, we introduce a deep network that includes a FCL that implements an interpretable raw solution for a Bayesian inversion. In Section IV, we propose an approach to address data corrupted by Poisson noise. In Section V, we describe how we implement and train the network. In Section VI, we describe the simulations, experimental acquisition, and variants for the reconstructions. In Section VII, we report and analyze our reconstruction results, before concluding in Section VIII.

C. Notations

Throughout this paper, deterministic variables are indicated by italic letters. In particular, we use normal font letters to denote scalars (e.g., $x \in \mathbb{R}$), lowercase bold letters for vectors (e.g., $\boldsymbol{x} \in \mathbb{R}^N$), and capital bold letters for matrices (e.g., $\boldsymbol{X} \in \mathbb{R}^{N\times M}$). The i-th element of the vector $\boldsymbol{x} \in \mathbb{R}^N$ is denoted by $(\boldsymbol{x})_i$. $\mathrm{Diag}\,(\boldsymbol{x}) \in \mathbb{R}^{N\times N}$ is a diagonal matrix where the diagonal elements are given by $\boldsymbol{x} \in \mathbb{R}^N$, while $\mathrm{diag}\,(\boldsymbol{A}) \in \mathbb{R}^N$ is a vector where the elements are taken from the main diagonal of the matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$. We define the weighted squared norm as $\|\boldsymbol{x}\|^2_{\boldsymbol{A}} = \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{A}$ is a positive definite matrix.

We use bold, non-italic symbols to denote random vectors. An N-dimensional random vector $\mathbf{x}$ that follows a multivariate normal distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^N$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{N\times N}$ is written as $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. An N-dimensional random vector $\mathbf{x}$ is said to follow a Poisson distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^N$ in notations $\mathbf{x} \sim \mathcal{P}(\boldsymbol{\mu})$, if it contains N independent Poisson-distributed random variables. We denote the expected value of a random variable $\mathbf{x}$ as $\mathbb{E}\,(\mathbf{x})$.

II. Compressive imaging

A. Compressive image acquisition

Let $\boldsymbol{f} \in [0,1]^N$ be the image to be acquired. The main idea of compressive optics is to measure a compressed version of $\boldsymbol{f}$ using hardware, and to recover this using software. The acquisition can be modeled by

$$\boldsymbol{m} = \boldsymbol{H}_1 \boldsymbol{f}, \tag{1}$$

where $\boldsymbol{m} \in \mathbb{R}^M$ is the measurement vector and $\boldsymbol{H}_1 \in \mathbb{R}^{M\times N}$, with $M < N$, is the acquisition matrix. The matrix $\boldsymbol{H}_1$ collects the patterns that are sequentially uploaded onto a spatial light modulator, to get $\boldsymbol{m}$. The patterns in $\boldsymbol{H}_1$ are traditionally chosen from among a basis $\boldsymbol{H} \in \mathbb{R}^{N\times N}$; i.e., $\boldsymbol{H}_1 = \boldsymbol{S}\boldsymbol{H}$ with $\boldsymbol{S} = [\boldsymbol{I}_M, \boldsymbol{0}]$. Classical choices include Fourier, discrete cosine, wavelets, and Hadamard bases (for a comparison, see [36]). The patterns $\boldsymbol{H}_1$ are often chosen before acquisition (e.g., retaining low frequency patterns; [37]), but they can also be sampled adaptively during acquisition [38]. An example of measurements that are available is illustrated in Fig. 1, where a Hadamard basis with the sampling strategy introduced in [37], [24] is used.

In practice, optical data are subject to Poisson noise [39]. Furthermore, implementation of the negative values of $\boldsymbol{H}_1$ implies the performing of extra measurements that are combined (more details can be found in [40]). Denoting $\boldsymbol{H}_1^+$ as the matrix such that $(\boldsymbol{H}_1^+)_{i,n} = \max((\boldsymbol{H}_1)_{i,n}, 0)$, and $\boldsymbol{H}_1^-$ as the matrix such that $(\boldsymbol{H}_1^-)_{i,n} = \max(-(\boldsymbol{H}_1)_{i,n}, 0)$, the noisy measurement vector $\hat{\mathbf{m}}^\alpha$ is obtained by subtracting the Poisson corrupted raw measurements $\hat{\mathbf{m}}^\alpha_+$ and $\hat{\mathbf{m}}^\alpha_-$; i.e.,

$$\hat{\mathbf{m}}^\alpha = \hat{\mathbf{m}}^\alpha_+ - \hat{\mathbf{m}}^\alpha_-, \tag{2a}$$

$$\hat{\mathbf{m}}^\alpha_+ \sim \mathcal{P}(\alpha \boldsymbol{H}_1^+ \boldsymbol{f}) \quad \text{and} \quad \hat{\mathbf{m}}^\alpha_- \sim \mathcal{P}(\alpha \boldsymbol{H}_1^- \boldsymbol{f}), \tag{2b}$$

where α is the intensity (in photons) of the image. This parameter is a scaling factor that controls the noise level. The larger the α, the larger the signal-to-noise ratio of the

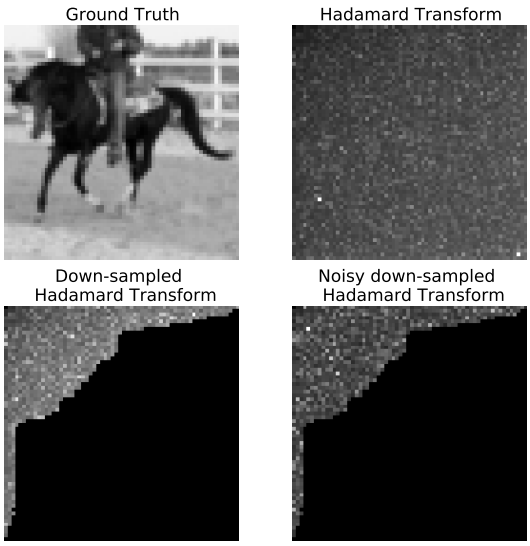

Fig. 1. An image from the stl-10 dataset, alongside its Hadamard transform, the Hadamard transform with missing coefficients (down-sampled Hadamard transform), and the Hadamard transform with missing coefficients (noisy down-sampled Hadamard transform). The noisy coefficients are subject to Poison noise, and we chose $\alpha = 10$ ph.

measurements. The effect of noise on the raw data is illustrated in Fig. 1. The scaling factor depends on the image photon flux (in photons/s), the quantum efficiency of the detector, and the the acquisition time (in s).

For different reasons that will be detailed in Section III and Section IV-A, we also introduce the normalized measurements here

$$\mathbf{m}^\alpha = \frac{1}{\alpha}\hat{\mathbf{m}}^\alpha \sim \frac{1}{\alpha}\mathcal{P}(\alpha \boldsymbol{H}_1^+ \boldsymbol{f}) - \frac{1}{\alpha}\mathcal{P}(\alpha \boldsymbol{H}_1^- \boldsymbol{f}). \tag{3}$$

Note that the expected value of the normalized measurements is consistent with the noiseless model of Equation (1), as $\mathbb{E}(\mathbf{m}^\alpha) = \boldsymbol{H}_1^+\boldsymbol{f} - \boldsymbol{H}_1^-\boldsymbol{f} = \boldsymbol{H}_1\boldsymbol{f} = \boldsymbol{m}$, and it is independent of α.

B. Compressive image reconstruction

Classical image reconstruction approaches consist of solving a minimization problem of the form

$$\min_{\boldsymbol{f}} \mathcal{R}(\boldsymbol{f}) \quad \text{such that} \quad \|\boldsymbol{m}^\alpha - \boldsymbol{H}_1\boldsymbol{f}\| \leq \epsilon. \tag{4}$$

where $\epsilon > 0$ is a parameter related to the noise level. A popular choice for $\mathcal{R}$ is the ℓ_2-norm $\mathcal{R} = \|\cdot\|_2$, which leads to a closed form solution; in the case of $\epsilon = 0$ for instance, the least squares solution is $\boldsymbol{f}^* = \boldsymbol{H}_1^\top[\boldsymbol{H}_1\boldsymbol{H}_1^\top]^{-1}\boldsymbol{m}$. An alternative choice is total variation $\mathcal{R} = \sum_n \|\boldsymbol{\nabla}_n \cdot\|_2$, where $\boldsymbol{\nabla}_n$ computes the gradient at pixel n, or the variants that promote piecewise constant solutions. In this case, Equation (4) has to be solved using time-consuming iterative optimisation algorithms.

More recently, a lot of effort has been put into learning to reconstruct $\boldsymbol{f}$ using non-linear models

$$\boldsymbol{f}^* = \mathcal{H}(\boldsymbol{m}^\alpha; \boldsymbol{\theta}), \tag{5}$$

where $\mathcal{H}$ is typically a neural network parameterized by $\boldsymbol{\theta}$. Given an image database, the parameters $\boldsymbol{\theta}$ are determined during a training stage by minimizing the reconstruction error. Many different neural network architectures have been proposed for this, which usually rely on convolutional layers. A very simple convolution architecture that includes a FCL and three convolutional layers is shown in Fig. 2. In particular, the network needs to adapt to the level of noise - implicitly or explicitly.

III. Denoised completion

A. Bayesian completion

Let $\boldsymbol{y} \in \mathbb{R}^N$ be the full data vector, such that

$$\boldsymbol{y} = \begin{bmatrix}\boldsymbol{y}_1 \\ \boldsymbol{y}_2\end{bmatrix} = \boldsymbol{H}\boldsymbol{f} = \begin{bmatrix}\boldsymbol{H}_1 \\ \boldsymbol{H}_2\end{bmatrix}\boldsymbol{f} \tag{6}$$

where $\boldsymbol{y}_1 = \boldsymbol{m} \in \mathbb{R}^M$ are the coefficients that are actually acquired in Equation (1), and $\boldsymbol{y}_2 \in \mathbb{R}^J$, $J = N - M$ are the missing coefficients. Assuming $\boldsymbol{H}$ is an orthogonal matrix, the least-squares solution simplifies to

$$\boldsymbol{f}^* = \boldsymbol{H}^\top \boldsymbol{y}^*, \quad \text{with } \boldsymbol{y}^* = \begin{bmatrix}\boldsymbol{m}^\alpha \\ \boldsymbol{0}\end{bmatrix}, \tag{7}$$

where $\boldsymbol{y}^*$ is obtained by zero-padding the acquired coefficients. However, neglecting the missing coefficients leads to reduced image quality, as will be shown in Section VII.

As in [25], we propose to complete the data vector with non-zero coefficients before reconstruction; i.e.,

$$\boldsymbol{f}^* = \boldsymbol{H}^\top \boldsymbol{y}^*, \quad \text{with } \boldsymbol{y}^*(\boldsymbol{m}^\alpha) = \begin{bmatrix}\boldsymbol{y}_1^* \\ \boldsymbol{y}_2^*\end{bmatrix}, \tag{8}$$

Our idea is to estimate the missing coefficients $\boldsymbol{y}_2^*$ through their correlation with the acquired coefficients. In the presence of noise, however, the acquired measurements $\boldsymbol{m}^\alpha$ are replaced by $\boldsymbol{y}_1^*$, to make the reconstruction more robust.

To formalize this idea, we adopt a Bayesian point-of-view and assume that the measurement $\boldsymbol{m}^\alpha$ is a sample of a random vector $\mathbf{m}^\alpha$. In this framework, the full data vector $\boldsymbol{y}^*$ is given by the conditional expectation

$$\boldsymbol{y}^*(\boldsymbol{m}^\alpha) = \mathbb{E}(\mathbf{y}_1, \mathbf{y}_2 \,|\, \mathbf{m}^\alpha = \boldsymbol{m}^\alpha) \tag{9}$$

where $\mathbf{y}_1$ and $\mathbf{y}_2$ are the random vectors associated to the acquired coefficients and the missing coefficients, respectively.

B. Gaussian completion in the presence of noise measurements

We first assume that the data random vector follows a multivariate normal distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^N$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{N\times N}$. In notations

$$\mathbf{y} = \begin{bmatrix}\mathbf{y}_1 \\ \mathbf{y}_2\end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{10}$$

where the covariance matrix takes the following form

$$\boldsymbol{\Sigma} = \begin{bmatrix}\boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_{21}^\top \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_2\end{bmatrix} \tag{11}$$

with $\boldsymbol{\Sigma}_1 \in \mathbb{R}^{M\times M}$ and $\boldsymbol{\Sigma}_{21} \in \mathbb{R}^{J\times M}$. This assumption is consistent with [41], where the statistical distributions of the coefficients of discrete Hadamard transforms of natural images

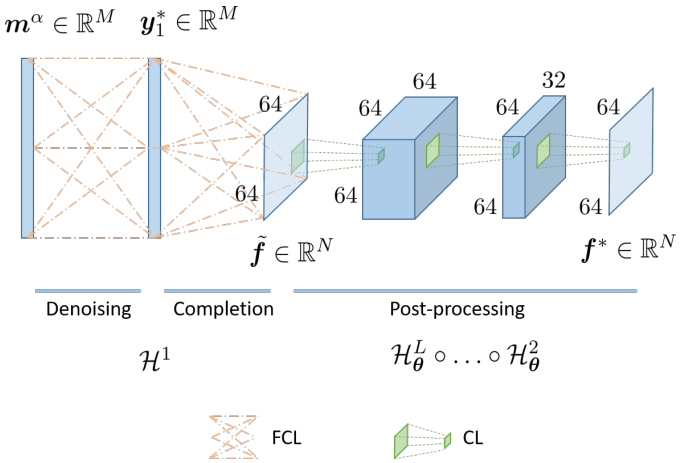

Fig. 2. Proposed denoised completion network (DC-Net). The first layer is a fully connected layer (FCL) that implements the best linear mean squared estimator of Equation (9) according to the analytical solution given by Equation (13). The raw image $\tilde{\boldsymbol{f}}$ is corrected by a cascade of convolution layers (CL) followed by rectified linear units (ReLU), which correct the remaining artefacts.

are best modeled by generalized Gaussian distributions. For our problem, we illustrate this assumption with the marginal distribution of Fig. 4.

We also consider that the measurement vector is corrupted by additive Gaussian noise. In the Bayesian framework, we have

$$\mathbf{m}^\alpha \,|\, \mathbf{y} \sim \mathcal{N}(\mathbf{y}_1, \boldsymbol{\Sigma}_\alpha), \tag{12}$$

where $\boldsymbol{\Sigma}_\alpha \in \mathbb{R}^{M\times M}$ is the noise covariance. Under the assumptions of Equations (10) and (12), we can exploit the properties of multivariate Gaussian variables (see Appendix A.1 of [42]), to obtain the full vector as

$$\boldsymbol{y}_1^*(\boldsymbol{m}^\alpha) = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_1[\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_\alpha]^{-1}(\boldsymbol{m}^\alpha - \boldsymbol{\mu}_1), \tag{13a}$$

$$\boldsymbol{y}_2^*(\boldsymbol{m}^\alpha) = \boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_1^{-1}[\boldsymbol{y}_1^*(\boldsymbol{m}^\alpha) - \boldsymbol{\mu}_1]. \tag{13b}$$

where Equation (13a) corresponds to denoising the acquired image $\boldsymbol{m}^\alpha$, and Equation (13b) to completing the missing coefficient from the denoised image. When the Gaussian assumptions of Equations (10) and (12) are not met, the denoised completion scheme of Equation (13) is optimal, in the sense that it provides the best linear estimator of our problem, as Equation (14). This solution thus solves (see Chapter 5 of [43])

$$\boldsymbol{y}^* = \arg\min_{\boldsymbol{y}} \|\boldsymbol{S}\boldsymbol{y} - \boldsymbol{m}^\alpha\|^2_{\boldsymbol{\Sigma}_\alpha^{-1}} + \|\boldsymbol{y} - \boldsymbol{\mu}\|^2_{\boldsymbol{\Sigma}^{-1}}. \tag{14}$$

C. Proposed denoised completion network

Assuming $\mathbf{y}$ has finite mean and variance, the conditional expectation of Equation (9) is given by the minimum mean squared error estimator

$$\boldsymbol{y}^* = \mathcal{G}^*(\boldsymbol{m}^\alpha), \tag{15}$$

where the mapping of $\mathcal{G}^*$ is given by

$$\mathcal{G}^* \in \arg\min_{\mathcal{G}} \mathbb{E}(\|\mathcal{G}(\mathbf{m}^\alpha) - \mathbf{y}\|^2). \tag{16}$$

Under the Gaussian assumption of Equation (10), $\mathcal{G}^*$ simplifies to the linear mapping provided by Equation (13). However, for general probability distributions of $\mathbf{y}$, the mapping of $\mathcal{G}^*$ is nonlinear, and there are no closed-form solutions.

As the problem of Equation (16) is computationally intractable in general, we propose to use learning of a mapping among a family of mappings parameterized by $\boldsymbol{\theta}$. Therefore, we replace the expectation by the empirical mean over a database, which leads to

$$\mathcal{G}^* \in \arg\min_{\boldsymbol{\theta}} \frac{1}{K}\sum_{k=1}^{K} \|\mathcal{H}_{\boldsymbol{\theta}}(\boldsymbol{m}^{\alpha,(k)}) - \boldsymbol{f}^{(k)}\|^2 \tag{17}$$

where $\{\boldsymbol{f}^{(k)}\}_{k=1}^K$ represents a database of K images, and $\{\boldsymbol{m}^{\alpha,(k)}\}_{k=1}^K$ represent the measurements associated to the image database through Equation (3). Note that the learning of Equation (17) is to map the raw data into the image domain, whereas the learning of Equation (16) is to complete the raw data in the measurement domain.

Much attention has been devoted to neural network models with cascaded layers

$$\mathcal{H}_{\boldsymbol{\theta}} = \mathcal{H}_{\boldsymbol{\theta}}^L \circ \ldots \circ \mathcal{H}_{\boldsymbol{\theta}}^1 \tag{18}$$

where $\mathcal{H}^\ell$, $1 \le \ell \le L$ is the ℓ-th (nonlinear) layer of the network, and $\circ$ is the function composition. Classically, the first layer is a FCL that maps the measurement $\boldsymbol{m}^\alpha \in \mathbb{R}^M$ to a raw solution in $\tilde{\boldsymbol{f}} \in \mathbb{R}^N$. To establish a link between measurement domain and image domain learning, we propose to parameterize the FCL $\mathcal{H}^1$ such that it provides the best linear mean squared error estimator; i.e.,

$$\tilde{\boldsymbol{f}} = \mathcal{H}^1(\boldsymbol{m}^\alpha) = \boldsymbol{H}^\top(\boldsymbol{W}\boldsymbol{m}^\alpha + \boldsymbol{b}) \tag{19}$$

where the weight $\boldsymbol{W}$ and bias $\boldsymbol{b}$ are computed using Equation (13). We illustrate the resulting network in Fig. 2. As the parameters of the FCL are fixed, the dimension of the variable $\boldsymbol{\theta}$, which is optimized during the training phase, is much reduced compared to more common neural networks where the FCL is free. Note that this architecture is similar to the automated transform by manifold approximation (AUTOMAP) network [44], which has been successfully applied to magnetic resonance imaging.

IV. Handling Poisson noise

A. Denoised completion in the presence of Poisson noise

Using the normal approximation to the Poisson distribution and assuming that the raw measurements are independent, we can approximate the noise model of Equation (3) as

$$\mathbf{m}^\alpha \sim \frac{1}{\alpha}\left[\mathcal{N}(\alpha\boldsymbol{H}_1^+\boldsymbol{f}, \hat{\boldsymbol{\Sigma}}_\alpha^+) - \mathcal{N}(\alpha\boldsymbol{H}_1^-\boldsymbol{f}, \hat{\boldsymbol{\Sigma}}_\alpha^-)\right] \tag{20}$$

where $\hat{\boldsymbol{\Sigma}}_\alpha^+ = \mathsf{Diag}\left(\alpha\boldsymbol{H}_1^+\boldsymbol{f}\right)$ and $\hat{\boldsymbol{\Sigma}}_\alpha^- = \mathsf{Diag}\left(\alpha\boldsymbol{H}_1^-\boldsymbol{f}\right)$. Rearranging the different terms leads to an expression that is in agreement with Equation (12)

$$\boldsymbol{\Sigma}_\alpha = \frac{1}{\alpha^2}\left(\hat{\boldsymbol{\Sigma}}_\alpha^+ + \hat{\boldsymbol{\Sigma}}_\alpha^-\right), \tag{21}$$

which can also be expressed as

$$\boldsymbol{\Sigma}_\alpha = \mathsf{Diag}\left(\boldsymbol{\sigma}_\alpha^2\right) = \mathsf{Diag}\left(\frac{1}{\alpha}\boldsymbol{H}_1^+\boldsymbol{f} + \frac{1}{\alpha}\boldsymbol{H}_1^-\boldsymbol{f}\right). \tag{22}$$

Therefore, we can investigate two extreme cases for the denoised completion scheme given by Equation (13). In high signal-to-noise situations for which α is large, Equation (13a) tends to $\boldsymbol{y}_1^* = \boldsymbol{m}^\alpha$ and $\boldsymbol{y}_2^* = \boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{m}^\alpha - \boldsymbol{\mu}_1)$. In other words, no denoising is required, and the completion approach of [25] applies directly. However, in low signal-to-noise situations for which α is small, $\boldsymbol{y}_1^*$ of Equation (13a) behaves as $\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_\alpha^{-1}\boldsymbol{m}^\alpha$. The lower α, the more the raw measurements are filtered out.

B. Noise covariance estimation

It is important to note that $\boldsymbol{\sigma}_\alpha^2$ depends on the unknown image $\boldsymbol{f}$ as well as on its unknown intensity α, as underlined in Equation (22). To address this issue, we propose to exploit the raw data that also depend on $\boldsymbol{f}$ and α. Recalling that the variance of the Poisson variable is equal to its expected value, we can write $\hat{\boldsymbol{\Sigma}}_\alpha^+ = \mathbb{E}\left(\hat{\mathbf{m}}_+^\alpha\right)$ and $\hat{\boldsymbol{\Sigma}}_\alpha^- = \mathbb{E}\left(\hat{\mathbf{m}}_-^\alpha\right)$, and choose to approximate the expected value by the noisy sample; i.e.,

$$\boldsymbol{\sigma}_\alpha^2 \approx \frac{1}{\tilde{\alpha}^2}(\hat{\boldsymbol{m}}_+^\alpha + \hat{\boldsymbol{m}}_-^\alpha) \tag{23}$$

Note that we replace the unknown intensity α in the denominator of Equation (21) by an estimation $\tilde{\alpha}$. The derivation of $\tilde{\alpha}$ from the raw measurement is detailed in Section IV-C.

C. Raw estimation of α

Data normalization as given by Equation (3) is fundamental in the presence of Poisson noise, as different noise levels are obtained by scaling the intensity of the light source. Scaling the measurement is not an issue for linear reconstructors; the reconstruction is scaled in the same way. However, this is a major concern for nonlinear reconstructors that behave quite differently for measurements with different amplitudes.

For image denoising, the estimation of α can be achieved through fitting methods (e.g., see [39]). These methods usually exploit homogeneous regions of the image, and are therefore not suited to raw measurements. Instead, we propose a simple empirical method, which consists of estimating α from a rough estimate of the nonnormalized image $\alpha\boldsymbol{f}$. Considering the pseudo inverse, which is a linear operator that scales with α, we consider

$$\tilde{\alpha} = \max_{i\in\{1,\dots,N\}} (\boldsymbol{H}^\top \boldsymbol{y}^*)_i, \quad \text{with } \boldsymbol{y}^* = \begin{bmatrix} \hat{\boldsymbol{m}}^\alpha \\ \mathbf{0} \end{bmatrix} \tag{24}$$

In practice, this simple estimator can provide the order of magnitude of α with good approximation, but it obviously leads to some inaccuracies. This is illustrated in Fig. 3, where we show the distribution of the error of our estimation over the STL-10 dataset [45] (i.e., 113,000 images obtained by merging the unlabeled, training and testing sets) for different values of M (512 or 1024) and α (5, 50, 2500 photons). As expected, we obtain the poorest estimations under high noise; i.e., when α is small (e.g, 5 photons). Overall, the relative error is usually below 50%. More advanced methods, such as that of [46], might improve the estimation of α; however, their iterative nature limits their integration within deep learning training strategies.

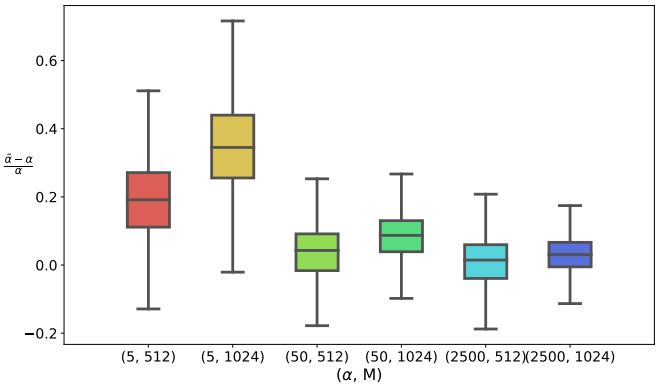

Fig. 3. Box plot of the distribution of the relative error on the estimation of α using Equation (24) with six combinations of values of (α, M).

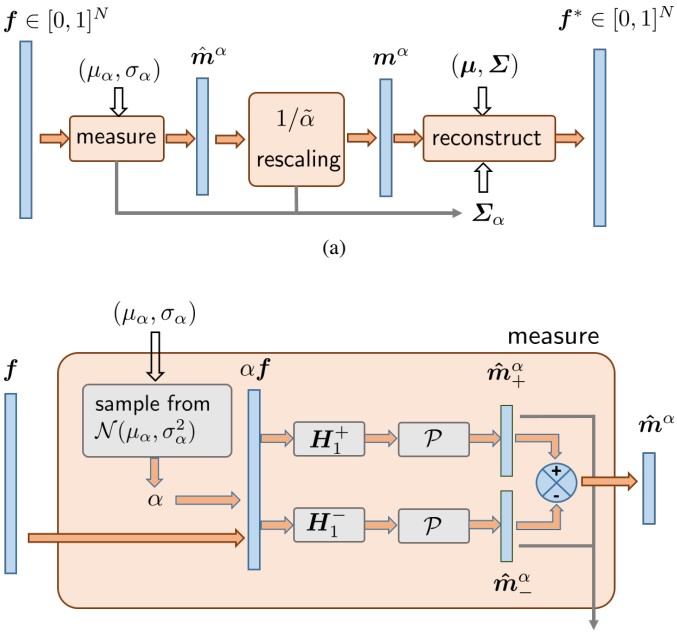

Fig. 4. Training of the network. (a) Overview of the full network involved during the training phase. The first module of the network simulates the raw measurements, considering an image intensity of $\mu_\alpha \pm \sigma_\alpha$ photons. Secondly, the raw measurement are normalized by an estimation of α. Thirdly, the network depicted in Fig. 2 reconstructs the image. While $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are precomputed according to Equations (25) and (26), $\boldsymbol{\Sigma}_\alpha = \text{Diag}\left(\boldsymbol{\sigma}_\alpha^2\right)$ is computed according to Equation (23). (b) Detailed description of the simulation of the measurements. Note that the image intensity is sampled from a normal distribution with mean μ_α and variance σ_α^2, and that the raw measurements $\hat{\boldsymbol{m}}_+^\alpha$ and $\hat{\boldsymbol{m}}_-^\alpha$ are passed to the reconstruction layers to estimate $\boldsymbol{\Sigma}_\alpha$.

V. Network Implementation and Training

A. Full network overview

Our full network is not limited to the reconstruction stage shown in Fig. 2. As shown in Fig. 4a, it also includes a module that simulates the raw measurements and a normalization stage. This implementation, where the forward model is computed by the network, allows the raw data to be simulated on graphics processor units directly, which is usually fast. This solution is also more flexible (e.g., easy access to a large variety of image databases).

The `measure` module of our network essentially implements Equation (2), where the image intensity α is sampled from a normal distribution with mean μ_α and variance σ_α^2. Note that the raw measurement $\hat{\boldsymbol{m}}_+^\alpha$ and $\hat{\boldsymbol{m}}_-^\alpha$ are passed to the reconstruction layers to estimate $\boldsymbol{\Sigma}_\alpha$.

The `rescaling` module essentially normalizes the raw measurements according to Equation (3). However, as α is unknown in practice, i.e., when only the raw measurement are available, the division by α is replaced by division by an estimation $\tilde{\alpha}$ that is obtained according to Equation (24). $\tilde{\alpha}$ is also passed to the reconstruction layers, to estimate $\boldsymbol{\Sigma}_\alpha$.

The `reconstruction` module corresponds to the denoised completion network to reconstruct the image, as shown in Fig. 2. While $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are precomputed according to Equations (25) and (26), $\boldsymbol{\Sigma}_\alpha = \operatorname{Diag}\left(\boldsymbol{\sigma}_\alpha^2\right)$ is computed according to Equation (23).

We implement the full networks using Pytorch [47] (version 1.5.1; cuda V10.2.89). We train the network by solving Equation (17) using the ADAM optimizer [48], with an initial learning rate of 10^{-3}, which is halved every 10 epochs, for a maximum of 100 epochs. In our experiments, the validation loss traditionally stops decreasing after 70 epochs. The training phase takes 3 h and 45 min on a NVIDIA GP107GLM [Quadro P1000 Mobile]. Our implementation of the denoised statistical completion network will be made publicly available through the initial release of the open-source Python SPYRiT toolbox [35]). We will also provide the Matlab implementation of the denoised completion method (as part of a new release of the SPIRiT Toolbox [34]).

B. Dataset

We train our network using the STL10 database [49], with $K = 105,000$ images that correspond to the 'unlabeled' and 'train' subsets. We consider 8,000 images for the testing (i.e., the 'test' subset). The original 96×96 images are resized to 64×64 using bicubic transform, and are normalized between -1 and 1.

C. A-priori mean and covariance

We compute the mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ as introduced in Equation (10) from the same 105,000 STL10 images that are considered during the training. We consider the classical estimators

$$\boldsymbol{\mu} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{H}\boldsymbol{f}^{(k)}, \tag{25}$$

$$\boldsymbol{\Sigma} = \frac{1}{K-1}\sum_{k=1}^{K} (\boldsymbol{H}\boldsymbol{f}^{(k)} - \boldsymbol{\mu})(\boldsymbol{H}\boldsymbol{f}^{(k)} - \boldsymbol{\mu})^\top \tag{26}$$

Note that both quantities are computed once and for all. We load them into the graphical processor unit when the network is initialized, with no need to re-compute them later (e.g., during training or evaluation).

D. Training with different source intensities

As seen in Section IV-C, it is not realistic to consider the intensity α as a known constant. Therefore, we make α vary during the training, with mean μ_α and standard deviation σ_α. Therefore, the network has to learn how to be robust to vary α about μ_α. To fit with the observations made in Fig. 3, we choose $\sigma_\alpha = 0.5\mu_\alpha$ to account for the worst-case scenarios. As shown in Section VII-C, varying alpha during the training phase, i.e., considering different noise levels, is crucial to get a robust network.

However, such an approach does not guarantee that a network trained around a given intensity μ_α can perform well for another intensity. A neural network trained for a certain probability distribution cannot generalize 'for free' to another probability distribution. We look into this question in Section VII-D.

E. Data normalization

Data normalization is a key to obtaining a robust network. By bringing the values of the measurements into the same range, the resulting network generalizes better to different noise levels, in particular when different noise levels are used during the training phase (see Section VII-C).

Here, we chose to train our network using zero-mean images $\boldsymbol{f}_{\text{net}}$ with values in the range $[-1, 1]$, to accelerate the optimisation [50]. By plugging $\boldsymbol{f}_{\text{net}} = 2\boldsymbol{f} - \mathbf{1}$ into Equation (1), where $\mathbf{1} = [1, \dots, 1]^\top \in \mathbb{R}^N$, we obtain the following input for the convolutional layers $\boldsymbol{m}_{\text{net}}^\alpha = \frac{2}{\alpha}\hat{\boldsymbol{m}}^\alpha - \boldsymbol{H}_1\mathbf{1}$.

The output of the neural network $\boldsymbol{f}_{\text{net}}$ can be normalized back to $[0, 1]$ by computing $\frac{1}{2}(\boldsymbol{f}_{\text{net}} + \mathbf{1})$, which allows fair comparison of the reconstruction and the ground truth. For simplicity, the diagram of Fig. 4 does not include the zero-mean normalization, and works with $\boldsymbol{f}$ directly.

F. Diagonal covariance approximation for rapid denoising

Implementing the denoised completion scheme of Equation (13) as a layer of a neural network is challenging, due to the matrix inversion required in Equation (13a). As the matrix to be inverted is signal dependent, the inverses cannot be precomputed; the inversion has to be computed on the fly, i.e., during the training phase. A matrix inversion is typically much more time consuming than the operations usually used during training (e.g., convolutions). Repeating matrix inversions many times during the optimization of Equation (17) might be prohibitive.

To lighten the computational burden, we choose to neglect the nondiagonal terms of $\boldsymbol{\Sigma}_1$. Denoting $\boldsymbol{\sigma}_1^2 = \operatorname{diag}(\boldsymbol{\Sigma}_1)$, we consider

$$\boldsymbol{y}_1^*(\boldsymbol{m}^\alpha) = \boldsymbol{\mu}_1 + \boldsymbol{\sigma}_1^2/(\boldsymbol{\sigma}_1^2 + \boldsymbol{\sigma}_\alpha^2)(\boldsymbol{m}^\alpha - \boldsymbol{\mu}_1), \tag{27}$$

where division and multiplication apply element-wise. As illustrated in Fig. 5, this approximation impacts on the quality of the reconstruction. The diagonal approximation of the covariance leads to reduced quality compared to the vanilla approach of Equation (13). However, the quality loss appears to be limited. Compared to no denoising at all (see Fig. 5b, c), diagonal denoising leads to clear improvement.

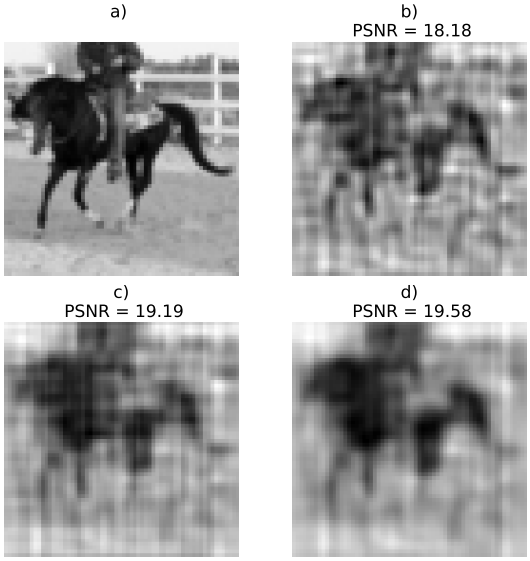

Fig. 5. (a) An image from the stl-10 dataset. (b) Reconstruction using statistical completion (SC); i.e., using Equation (13b) with $\boldsymbol{y}_1^* = \boldsymbol{m}^\alpha$. (c) Reconstruction using denoised statistical completion (diagonal DSC) with the diagonal approximations; i.e., Equations (13a) and (27). (d) Reconstruction using the vanilla denoised statistical completion (DSC) of Equation (13). The acquired coefficients are subjected to Poison noise according to Equation (2), where we set $\alpha = 7$ photons. The noise covariance was computed using Equation (23).

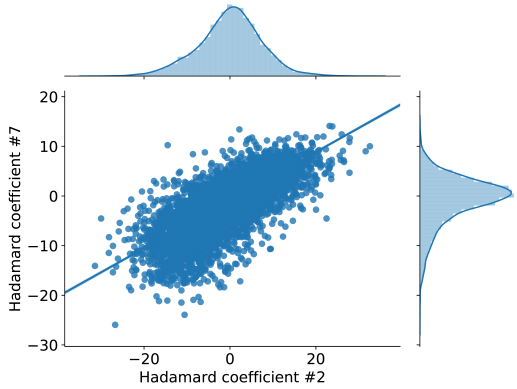

Fig. 6. Measurement scatter plot for two coefficients of the measurement vector in the absence of noise. Each dot represents an image from the STL10 test set. The two histograms represent the marginal distribution of each variable, while the full line on top of the scatter plot indicates the correlation between the two variables.

VI. Experiments

A. Reconstruction methods

We consider three linear reconstruction methods and three nonlinear methods.

The linear reconstruction methods are :

1) Pseudo inverse (PI): This solution is given by Equation (7) and corresponds to no completion and no noise handling.

2) Statistical completion (SC): This solution is given by Equation (13) (where $\boldsymbol{\Sigma}\alpha$ is replaced by $\mathbf{0}$). Here, the noise is not taken into account.

3) Denoised statistical completion (DSC): This solution is given by Equation (13). It corresponds to the best linear solution of Equation (16).

The nonlinear reconstructors are some of the variants of the reconstruction network depicted on Fig. 2. These networks have different FCL $\mathcal{H}^1$, and differ in the way they are trained.

4) Completion network (C-Net): The weights of the FCL are frozen such that no denoising is performed; i.e., Equation (13) is used with $\boldsymbol{y}_1^* = \boldsymbol{m}^\alpha$. This is the network that was introduced in [25]

5) Noisy completion network (NC-Net): The architecture of the NC-Net and the C-Net are identical; however, the NC-Net is trained in the presence of noise, with benefit from the training machinery described in Section V. Here, the post-processing layers $\mathcal{H}_{\boldsymbol{\theta}}^L \circ \ldots \circ \mathcal{H}_{\boldsymbol{\theta}}^2$ implicitly learn in the image domain how to cope with noise.

6) Denoised completions network (DC-Net): The weights of the FCL are chosen so as to implement Equation (13), which includes the denoising step of Equation (13a). Here, we cope with noise explicitly in the measurement domain.

Note that all of these approaches are suited to real-time imaging, as the slowest method, which is the DC-Net, can reconstruct an image in only 30 ms with a NVIDIA GP107GLM GPU [Quadro P1000 Mobile]; the other approaches can take up to 4 ms less than the DC-Net.

B. Simulated data

We evaluate the reconstruction error for each of the 5,000 unseen images (i.e., the'test' subset of STL-10) using the `measure` module described in Fig. 4. We recall that this includes two sources of uncertainties; namely, the measurement noise and the variability of the image intensity.

Figure 6 illustrates the distribution of two coefficients of the measurement vector in the absence of noise. It can be seen that the marginal distributions for both of the coefficients are nearly Gaussian, which is representative of all of the coefficients and motivates the hypothesis of Equation (10). The full line represents the correlation between the two coefficients, which motivate the use of the completion scheme of Equation (13b).

C. Experimental data

To validate our reconstruction methods, we consider the single-pixel camera experimental set-up depicted in Fig. 7, as first described in [51]. The telecentric lens (Edmund Optics 62901) is positioned such that its image side projects the image of the scene onto the digital micro-mirror device (DMD; vialux V-7001), which is positioned at the object side of the lens. The object is transparent and is illuminated by a LED lamp (Thorlabs LIUCWHA/M00441662). The DMD can implement different light patterns (denoted as $\boldsymbol{H}_1$ in Section II) by reflection of the incident light onto a relay lens, which projects the light into an optical fiber (Thorlabs FT1500UMT 0.39NA). This optical fiber is connected to a compact spectrometer (BWTek examplar BRC115P-V-ST1). For every object, we sequentially upload onto the DMD all of the $M = 4096$ Hadamard patterns of dimension $N = 64 \times 64$ pixels. We

can down-sample the full measurement vector a posteriori to achieve any sampling ratio. To consider different noise levels, we acquire the same object with varying integration times and neutral optical densities.

As shown in the first column of Fig. 8, we acquire three different objects: the LED lamp directly (first row); a cat from the STL-10 test set printed on a transparent sheet (middle row); and the Siemens Star resolution target (bottom row) [52]. The ground-truth image is computed (see Section VI-D) from a fully sampled measurement vector that is acquired with the highest signal-to-noise ratio (i.e., high flux illumination, long acquisition time). Specifically, we consider no neutral density, and the integration time to 1 ms per pattern for the lamp, to 4 ms per pattern for the STL-10 cat, and to 8 ms per pattern for the Siemens target,

For each object, we also acquire a high-noise dataset by placing neutral optical density behind the lamp to reduce the light flux. We consider optical densities of 1.3 for the LED lamp, 0.6 for the STL-10 cat, and 0.3 for the Siemens target. We set the integration time to 4 ms for both the LED lamp and the STL-10 cat, and to 8 ms for the Siemens target, we retain only $M = 512$ measurements for both the LED lamp and the STL-10 cat, which accelerates the acquisition by a factor of 8. For the Siemens target, which has a richer spatial frequency content, we keep more measurements ($M = 2048$; acceleration factor of 2).

D. Evaluation metrics

Given the ground-truth image $\boldsymbol{f}$, we compute the peak signal-to-noise ratio (PSNR) of a reconstructed image $\boldsymbol{f}^*$ as

$$\mathsf{PSNR}(\boldsymbol{f}^*, \boldsymbol{f}) = 10 \log_{10} \frac{2^2}{\|\boldsymbol{f}^* - \boldsymbol{f}\|^2} \tag{28}$$

For the experimental data, we have no direct access to the ground truth image. This image is computed as $\boldsymbol{f} = \boldsymbol{H}^\top \boldsymbol{y}_{\mathrm{gt}}$, where $\boldsymbol{y}_{\mathrm{gt}}$ is a fully sampled measurement with high signal-to-noise ratio. Before computing the PSNR according to Equation (28), we normalize the ground truth image in the range $[0, 1]$ with $\boldsymbol{f} \leftarrow (\boldsymbol{f} - \min_i(\boldsymbol{f})_i)/(\max_i(\boldsymbol{f})_i - \min_i(\boldsymbol{f})_i)$.

VII. Results and Discussion

A. Performance of linear reconstructors

In Table I, we report the performance of the SC and the DSC methods for five image intensities/noise levels (from $\alpha = 2 \pm 1$ photons, to $\alpha = 2500 \pm 1250$ photons). We observe that DSC outperforms SC for all noise levels. On average, the gain is 7.89 dB, 5.84 dB, 4.68 dB, 2.06 dB, 0.06 dB for the mean intensity of 2, 5, 10, 50, and 2500 photons, respectively. As expected, the superiority of DSC vanishes for increasing intensity of α (i.e., decreasing noise).

Note that the PSNRs for the two image intensities are not directly comparable. This is mainly due to the rescaling described in Fig. 4. The rescaling is carried out with an estimated image intensity $\tilde{\alpha}$, which deviates from the true intensity $\tilde{\alpha}$. This has a high impact on the PSNR when α is high. However, reconstruction methods from the same level of noise can be compared without any issues, as they apply to the same dataset.

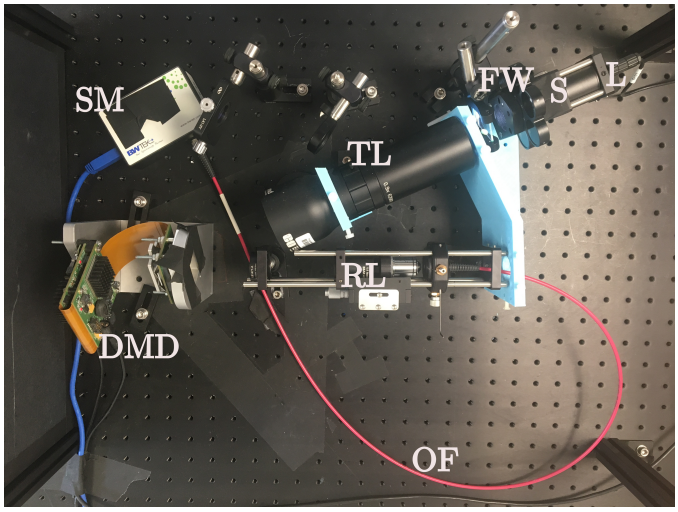

Fig. 7. Optical set-up of the single-pixel camera [14]. This set-up is composed of a sample (S) illuminated by a lamp (L) in front of a filter wheel (FW), a telecentric lens (TL), a digital micro-mirror device (DMD), some relay lenses (RL), an optical fiber (OF), and a spectrometer (SM).

We also observe the superiority of DSC for the experimental data in Fig. 8. For the LED lamp, we get a PSNR of 15.98 dB, as opposed to 15.56 dB and 13.96 dB for SC and pseudo inverse, respectively. For the STL-10 cat, the PSNR is significantly improved using DSC (18.73 dB) than using SC (16.92 dB) or pseudo inverse (15.96 dB). For the Siemens Star, we get a PSNR of 16.09 dB, compared to 15.95 dB and 15.88 dB for SC and pseudo inverse, respectively. Visually it appears that the denoising approach tends to remove some of the artifacts introduced by the Poisson noise; the LED lamp and the STL-10 cat in particular are much smoother and easier to recognize. This confirms that the completion approach performs well in the presence of Poisson noise.

B. Training a network for noisy data

In Table II, we report the reconstruction PSNR using networks that are trained for different noise levels. In particular, the first column ($\alpha = \infty$) corresponds to networks that are trained with no noise. Here, we observe that a network that is trained with no noise performs very poorly when it is evaluated on noisy data. The reconstruction PSNR drops to as low as 9.48 ± 1.78 dB for $\alpha = 2 \pm 1$ ph and the reconstruction quality is significantly degraded, even under very low noise levels (e.g., $\alpha = 2500 \pm 1250$ ph.), as the PSNR reaches only 15.88 ± 2.11 dB, while the NC-Net trained with $\alpha = 2500 \pm 1250$ ph. performs significantly better, with PSNR of 22.19 ± 1.56 dB.

This can be further confirmed with the experimental data of Fig. 8. We recall that the C-Net and the NC-Net share the same architecture but the NC-Net is trained with noise, contrary to the C-Net. Training with no noise leads to significant drop in the reconstruction PSNR (15.65-13.79 = 1.86 dB for the LED; 2.35 dB for the cat; 0.15 dB for the Siemens Star).

	Testing				
α	2 ± 1	5 ± 2.5	10 ± 5	50 ± 25	2500 ± 1250
SC	9.72 ± 1.76	12.43 ± 1.80	13.91 ± 1.85	15.63 ± 1.96	16.19 ± 2.02
DSC	17.61 ± 1.23	18.37 ± 1.22	18.59 ± 1.26	17.70 ± 1.58	16.25 ± 2.01

TABLE I

AVERAGE PEAK SIGNAL-TO-NOISE RATIO OF STATISTICAL COMPLETION (SC) [25] AND DENOISED STATISTICAL COMPLETION (DSC) OF EQUATION (27) - RECONSTRUCTIONS UNDER VARIOUS NOISE LEVELS: $\alpha = 2 \pm 1$ PH.; $\alpha = 5 \pm 2.5$ PH.; $\alpha = 10 \pm 5$ PH.; $\alpha = 50 \pm 25$ PH.; AND $\alpha = 2500 \pm 1250$ PH.. FROM LEFT TO RIGHT, IMAGES ARE RECONSTRUCTED FROM DATA SIMULATED WITH DECREASING NOISE LEVELS.

	Testing		Training				
	α (in ph.)	∞ / C-Net	2 ± 1	5 ± 2.5	10 ± 5	50 ± 25	2500 ± 1250
NC-Net	2 ± 1	9.48 ± 1.78	18.79 ± 1.33	18.12 ± 1.41	17.04 ± 1.50	15.19 ± 1.43	14.55 ± 1.38
	5 ± 2.5	12.11 ± 1.87	19.80 ± 1.42	20.11 ± 1.39	19.84 ± 1.40	18.70 ± 1.34	18.12 ± 1.28
	10 ± 5	13.58 ± 1.94	**20.11 ± 1.47**	**20.75 ± 1.46**	20.88 ± 1.44	20.39 ± 1.38	19.98 ± 1.31
	50 ± 25	15.32 ± 2.06	**20.32 ± 1.51**	**21.16 ± 1.53**	**21.59 ± 1.53**	21.83 ± 1.52	21.76 ± 1.49
	2500 ± 1250	15.88 ± 2.11	**20.37 ± 1.52**	**21.24 ± 1.55**	**21.73 ± 1.56**	**22.14 ± 1.57**	22.19 ± 1.56
DC-Net	2 ± 1		18.79 ± 1.47	**18.74 ± 1.53**	**18.60 ± 1.55**	**18.31 ± 1.55**	**18.14 ± 1.55**
	5 ± 2.5		19.76 ± 1.39	20.05 ± 1.50	**19.99 ± 1.55**	**19.61 ± 1.60**	**19.34 ± 1.59**
	10 ± 5		19.88 ± 1.33	20.64 ± 1.42	20.82 ± 1.51	**20.55 ± 1.61**	**20.21 ± 1.62**
	50 ± 25		18.68 ± 1.43	20.15 ± 1.36	21.04 ± 1.39	21.86 ± 1.54	21.69 ± 1.61
	2500 ± 1250		17.40 ± 1.71	18.89 ± 1.63	19.94 ± 1.59	21.69 ± 1.51	22.17 ± 1.56

TABLE II

RECONSTRUCTION OF PEAK SIGNAL-TO-NOISE RATIOS (PSNRS) FOR DIFFERENT NETWORKS. NOTE THAT THE NC-NET TRAINED WITH NO NOISE ($\alpha = \infty$) CORRESPONDS TO THE C-NET. FROM TOP TO BOTTOM, IMAGE ACQUISITION IS SIMULATED ASSUMING INCREASING LIGHT INTENSITY α (I.E., DECREASING NOISE LEVELS). FROM LEFT TO RIGHT, IMAGES ARE RECONSTRUCTED BY NETWORKS THAT ARE TRAINED USING DECREASING NOISE LEVELS. TO FACILITATE THE COMPARISON BETWEEN THE TWO NETWORKS, WE UNDERLINE SIMILAR PSNRS (I.E., DIFFERENCE ¡0.1 DB) AND USE BOLD FONT TO INDICATE THE BEST PERFORMING NETWORKS (I.E., DIFFERENCE ¿0.1 DB). BLUE FONT INDICATES THAT THE TESTING AND TRAINING NOISE ARE THE SAME LEVEL; GREEN FONT, THAT THE TESTING NOISE IS LOWER THAN THE TRAINING NOISE; AND RED FONT, THAT THE TESTING NOISE IS HIGHER THAN THE TRAINING NOISE.

C. Training with varying levels of noise

In Table III, we evaluate the effect of training a network under a varying noise levels. We also simulate the acquisition of the test images with all with the same source intensity (α = 50 photons) and for varying intensities (mean values μ_α = 50 photons and standard deviation $\sigma_\alpha = 25$ photons).

Training a network under different noise levels is detrimental when the image intensity is known; on average, it results in a PSNR drop of $23.19 - 22.13 = 1.06$ dB. On the contrary, noise-varying training improves the reconstruction when the image intensity is unknown; on average, we obtain an enhancement of $21.07 - 19.33 = 1.74$ dB (second row of Table III).

This observation is crucial, as the exact image intensity is not available beforehand in real-life experiments. Although we can estimate the image intensity from the raw data (e.g., by the method introduced in Section IV-C), this inevitably leads to inaccuracies. In the following, we only consider realistic scenarios where the image intensity is not known, which requires training with varying noise levels.

Testing	Training	
α (in ph.)	50	50 ± 25
50	23.19 ± 1.98	22.13 ± 1.88
50 ± 25	19.33 ± 1.67	21.07 ± 1.63

TABLE III

TRAINING AND TESTING UNDER VARYING NOISE LEVELS. TOP ROW: DATA SIMULATED FOR A GIVEN THE SOURCE INTENSITY (α = 50 PHOTONS), WHICH IS THE SAME FOR ALL TEST IMAGES. BOTTOM ROW: DATA SIMILATED FOR TEST IMAGES WITH VARYING INTENSITIES (MEAN VALUES μ_α = 50 PHOTONS AND STANDARD DEVIATION $\sigma_\alpha = 25$ PHOTONS). RECONSTRUCTION PSNRS ARE REPORTED FOR A NETWORK TRAINED USING CONSTANT INTENSITY (MIDDLE COLUMN) AND VARYING INTENSITY (RIGHT COLUMN).

D. Performance of nonlinear methods in simulations

In Table II, we report the reconstruction PSNRs obtained using NC-Net, DC-Net, and also C-Net, which is given by the first column of NC-Net ($\alpha = \infty$). We train both the NC-Net and the DC-Net with varying noise levels, where the standard deviation of the image intensity is set to 50%, in agreement with the findings of Section IV-C and in Section VII-C.

As expected, the C-Net, which is trained with no noise, performs very poorly in the presence of noise. Therefore, we focus our analysis on the NC-Net and the DC-Net, and divide our analysis into three cases, which depend on the relative noise levels used during training and testing.

1) Training noise and testing noise have the same levels: In Table II, the PSNRs of these experiments (i.e., equal noise levels) are shown in blue (i.e., diagonals). The NC-Net and DN-Net have very similar performances. For decreasing noise levels, for NC-Net and DC-Net we obtain (respectively): 18.79 ± 1.78 dB against 18.79 ± 1.44 dB; 20.11 ± 1.39 dB against 20.05 ± 1.50 dB; 20.88 ± 1.44 dB against 20.82 ± 1.51 dB; 21.83 ± 1.52 dB against 21.86 ± 1.54 dB; and 22.19 ± 1.56 dB against 22.17 ± 1.56 dB. For all of the noise levels, the differences are below 0.1 dB on average. This means that the post processing layers of the NC-Net can compensate for the lack of the denoising layer, due to a training scenario that perfectly mimics the test one.

2) Training noise is lower than testing noise: In Table II, the PSNRs of these experiments (i.e., training lower than testing) are shown in red (below lower diagonal). For all of the experiments except one, the NC-Net produces better average reconstructed PSNRs than the DC-Net. For example, a DC-Net trained using $\alpha = 2 \pm 1$ photons, gives a reconstruction PSNR of 17.40 ± 1.71 dB, while an NC-Net trained at the same noise level gives a PSNR of 20.37 ± 1.52 dB, which is a large improvement of 2.97 dB, on average. Here, the presence of the denoising layer is detrimental; it kills the fine information that is useful in low-noise experiments.

3) Training noise is higher than testing noise: In Table II, the PSNRs of these experiments (i.e., training higher than testing) are shown in green (above diagonals). Contrary to the previous scenario, the DC-Net outperforms the NC-Net in all experiments but one. For example, a DC-Net trained using $\alpha = 2500 \pm 1250$ photons, gives a reconstruction PSNR of 18.14 ± 1.55 dB while a NC-Net trained the same noise level yields to a PSNR of 14.55 ± 1.38 dB, which is a large decrease of 3.59 dB, on average. Here, the presence of the denoising layer has a great advantage; the DC-Net generalizes better to high noise experiments.

In summary, the noise levels of the images under acquisition are a key feature. In real-life experiments, this implies that this parameter can be estimated first, and then the network that fits the actual noise level can be evaluated. In this case, both the NC-Net and the DC-Net are equivalent. However, this approach requires the loading of several networks, which might be a severe limitation for real-time applications, and in particular if the models are too large to be stored directly on the graphics processor unit. If only one network has to be retained, the DC-Net outperforms the NC-Net in high-noise situations. For $\alpha = 2 \pm 1$ photons, the worst PSNR across all of the noise levels used for training is 18.14 ± 1.55 dB for the DC-Net, while it collapses to 14.55 ± 1.38 dB for the NC-Net. Also, for a target noise level, it might be safe the choose the DC-Net, as it is more robust to noise levels that are higher than expected, compared to the NC-Net.

E. Performance of nonlinear methods considering experimental data

Fig. 8 illustrates the performance of our networks on experimental data acquired with the experimental set-up of a single-pixel camera presented in Section VI. We choose a training intensity of α = 50 photons, which corresponds to an intermediate level of noise (neither as noisy as α = 2 photons, nor as uncorrupted as α = 2500 photons). We find that the DC-Net outperforms the other networks (as well as the linear methods). For the LED lamp, the DC-Net yields a reconstruction PSNR of 16.14 dB, which is a $16.14 - 15.65 = 0.49$ dB ($16.14 - 13.79 = 2.35$ dB, resp.) improvement compared to the NC-Net (C-Net, resp.). For the STL-10 cat, the DC-Net outperforms the DC-Net (C-Net, resp.) by 0.73 dB (3.09 dB, resp.). For the Siemens Star, the the DC-Net outperforms the DC-Net (C-Net, resp.) by 0.46 dB (0.61 dB, resp.).

The gain provided by the DC-Net is more apparent in cases where the noise is high ($\tilde{\alpha} = 10$ photons for the LED lamp, and $\tilde{\alpha} = 9$ photons for the STL-10 cat), whereas it is more modest when the noise level is lower ($\tilde{\alpha} = 25$ photons for the Siemens star). Visually, the C-Net images are subject to multiple artefacts that share their pattern with the artefact observed in the pseudo inverse images. While the NC-Net has a smoothing effect, removing most noise artefacts, the DC-Net provides the smoothest images, while preserving most of the details (see the Siemens star image, for instance).

F. Limitations

One limitation of our deep network compared to more classic approaches such as [28], [30], [31] is that there is has no theoretical guarantee that it will work for any image; in particular, the image under acquisition significantly differs from those of in our training set. This is a common concern for deep-learning approaches that are, however, seen to work well in practise. Also, this deep network that is trained using simulations only generalizes relatively well considering the experimental data. A limiting aspect of our work might be the choice of our architecture, which is shallower than popular architectures, such as the U-Net used in [53] and more recent variants. However, we are keen to keep the number of network parameters as low as possible, to keep both the training and evaluation times as short as possible. Another limitation of this work concerns the analysis of the PSNRs of the images reconstructed from the experimental data, where the ground truth is not known. We limit this common issue by acquiring fully sampled low-noise images. Finally, we only test our algorithms on $N \times N$ pixel images, with $N = 64$. By considering a database with high-resolution images (e.g., ImageNet), our network can be generalized to handle the case where $N > 64$, directly or by implementing a patch-based strategy to limit the memory requirements.

G. Generalization to other problems

Although this work focuses on single-pixel imaging, it can provide some insight for any linear reconstruction problem where the measured data is scarcely sampled (e.g., limited angle computerized tomography, accelerated magnetic resonance imaging). For orthogonal transform, the generalization is straightforward, as it only consists of setting $\boldsymbol{H}$ to the corresponding basis (e.g., discrete Fourier for magnetic resonance imaging). For nonorthogonal transforms (e.g., computerized tomography), Equations (8) and (24) need to be updated by replacing $\boldsymbol{H}^\top$ with $\boldsymbol{H}^\top(\boldsymbol{H}\boldsymbol{H}^\top)^{-1}$.

This network can easily be adapted to other noise models, as we only need an estimation of the mean and covariance of the noisy measurements. For instance, we can handle the Poissonian-Gaussian noise models advocated in [39] simply by adding a constant term to $\boldsymbol{\Sigma}_\alpha$ in Equation (22).

Compared to most studies dedicated to deep learning for inverse problems, we mainly focus on the (fully connected) layers of the network, which acts in the measurement domain. The post-processing layers that act in the image domain can be replaced by any variants (e.g., U-Net, resNet, others). Moreover, our approach is compatible with architectures inspired by conventional variational methods (e.g., [6]). This will be the object of future work.

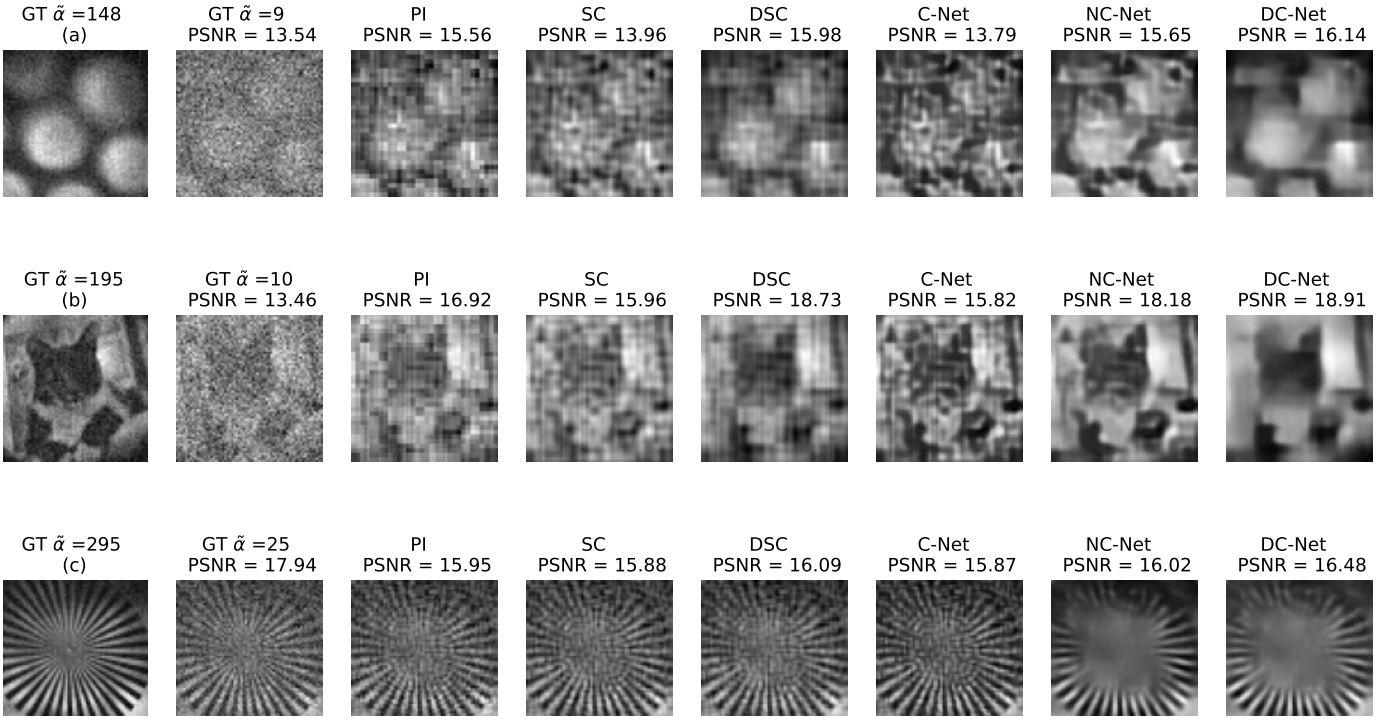

Fig. 8. Reconstructions of three experimental datasets by the different methods (top row: LED lamp with $M = 512$, middle row: STL-10 cat with $M = 512$; bottom row: Siemens star resolution target with $M = 2048$). We display the images reconstructed from a fully sampled dataset (ground-truth; GT) acquired with high image intensity (first column, $\alpha = 148$ photons) and lower image intensity (second column, $\alpha = 9$ photons). The following columns shown reconstructions using the pseudo inverse (PI) of Equation (7), the statistical completion (SC) of Equation (13), the denoised statistical completion (DSC) of Equation (13), the C-Net Section VI-A5, the NC-Net Section VI-A5 (trained with $\mu_\alpha = 50$ photons and $\sigma_\alpha = 0.5\mu_\alpha$), and the DC-Net Section VI-A6 (trained with $\mu_\alpha = 50$ photons and $\sigma_\alpha = 0.5\mu_\alpha$). All of the PSNRs are computed as described in Section VI-D, with the first column as the ground-truth.

VIII. Conclusion

We proposed a deep network that can reconstruct images from a small number of noisy single-pixel measurements. Our method is generic and can be applied to any linear inverse problem where the raw data is corrupted by Poisson noise. The proposed network includes a FCL that maps data from the measurement domain to the image domain. This FCL takes into account the signal-dependent covariance of the noise, and computes the best linear solution of a completion problem. This approach yields better reconstructions in scenarios where the network is tested under more noisy conditions when compared to their training values of noise. This is verified here on experimental data from a single-pixel camera. These results are particularly interesting under real-time constraints, or when the use of several neural networks to deal with different noise levels is not compatible with the amount of memory available. Under experimental conditions, we cannot foretell what levels of noise we will be faced with, therefore our approach is suited for dealing with real-time experimental data. In future work, we will use deep learning models that ensures that the output of the model is consistent with the inputs of said model. In order to do so, we will work on unrolled recursive networks that extend the ideas developed in this paper.

References

[1] S. Ravishankar et al., "Image reconstruction: From sparsity to data-adaptive methods and machine learning," Proceedings of the IEEE, pp. 1–24, 2019.

[2] G. Wang et al., "Image reconstruction is a new frontier of machine learning," IEEE Transactions on Medical Imaging, vol. 37, no. 6, pp. 1289–1296, 2018.

[3] S. Arridge et al., "Solving inverse problems using data-driven models," Acta Numerica, vol. 28, p. 1–174, 2019.

[4] M. T. McCann et al., "Convolutional neural networks for inverse problems in imaging: A review," IEEE Signal Processing Magazine, vol. 34, no. 6, pp. 85–95, 2017.

[5] E. Kang et al., "Cycle-consistent adversarial denoising network for multiphase coronary CT angiography," Medical Physics, vol. 46, no. 2, pp. 550–562, Dec. 2018.

[6] H. K. Aggarwal et al., "Modl: Model-based deep learning architecture for inverse problems," IEEE Transactions on Medical Imaging, vol. 38, no. 2, pp. 394–405, 2019.

[7] H. Gupta et al., "Cnn-based projected gradient descent for consistent ct image reconstruction," IEEE Transactions on Medical Imaging, vol. 37, no. 6, pp. 1440–1453, June 2018.

[8] I. Y. Chun et al., "Convolutional analysis operator learning: Acceleration and convergence," IEEE Transactions on Image Processing, vol. 29, pp. 2108–2122, 2020.

[9] K. Monakhova et al., "Learned reconstructions for practical mask-based lensless imaging," Opt. Express, vol. 27, no. 20, pp. 28 075–28 090, Sep 2019.

[10] D. Gilton et al., "Neumann networks for linear inverse problems in imaging," IEEE Transactions on Computational Imaging, vol. 6, pp. 328–343, 2020.

[11] G. Barbastathis et al., "On the use of deep learning for computational imaging," Optica, vol. 6, no. 8, pp. 921–943, Aug 2019.

[12] M. R. Kellman et al., "Physics-based learned design: Optimized coded-illumination for quantitative phase imaging," IEEE Transactions on Computational Imaging, vol. 5, no. 3, pp. 344–353, 2019.

[13] M. P. Edgar et al., "Principles and prospects for single-pixel imaging," Nature Photonics, vol. 13, no. 1, pp. 13–20, Jan. 2019.

[14] M. Duarte et al., "Single-pixel imaging via compressive sampling," Signal Processing Magazine, IEEE, vol. 25, no. 2, pp. 83–91, March 2008.

[15] V. Studer et al., "Compressive fluorescence microscopy for biological and hyperspectral imaging," Proceedings of the National Academy of Sciences, vol. 109, no. 26, pp. E1679–E1687, 2012.

[16] F. Rousset et al., "Time-resolved multispectral imaging based on an adaptive single-pixel camera," Opt. Express, vol. 26, no. 8, pp. 10 550–10 558, Apr 2018.

[17] G. R. Arce et al., "Compressive coded aperture spectral imaging: An introduction," IEEE Signal Processing Magazine, vol. 31, no. 1, pp. 105–115, Jan 2014.

[18] Q. Pian et al., "Compressive hyperspectral time-resolved wide-field fluorescence lifetime imaging." Nature photonics, vol. 11, pp. 411–414, 2017.

[19] E. Aguénounon et al., "Single snapshot imaging of optical properties using a single-pixel camera: a simulation study," Journal of Biomedical Optics, vol. 24, no. 7, pp. 1 – 6, 2019.

[20] Y. Zhang et al., "Dual-band single-pixel telescope," Opt. Express, vol. 28, no. 12, pp. 18 180–18 188, Jun 2020.

[21] F. Li et al., "Compressive ghost imaging through scattering media with deep learning," Opt. Express, vol. 28, no. 12, pp. 17 395–17 408, Jun 2020.

[22] R. Yao et al., "Net-flics: fast quantitative wide-field fluorescence lifetime imaging with compressed sensing - a deep learning approach," Light: Science & Applications, vol. 8, no. 1, p. 26, Mar. 2019.

[23] J. T. Smith et al., "Unmix-me: spectral and lifetime fluorescence unmixing via deep learning," Biomed. Opt. Express, vol. 11, no. 7, pp. 3857–3874, Jul 2020.

[24] C. Higham et al., "Deep learning for real-time single-pixel video," Scientific Reports., no. 8, Feb 2018.

[25] N. Ducros et al., "A completion network for reconstruction from compressed acquisition," in 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), 2020, pp. 619–623.

[26] F. J. Anscombe, "The transformation of Poisson, binomial and negative-binomial data," Biometrika, vol. 35, no. 3-4, pp. 246–254, 12 1948.

[27] N. Wiener, Extrapolation, interpolation, and smoothing of stationary time series. New York: John Wiley & Sons, Inc, 1949.

[28] S. Lefkimmiatis et al., "Poisson image reconstruction with hessian schatten-norm regularization," IEEE Transactions on Image Processing, vol. 22, no. 11, pp. 4314–4327, 2013.

[29] Q. Ding et al., "Statistical image reconstruction using mixed poisson-gaussian noise model for x-ray ct," arXiv: Medical Physics, 2018.

[30] F. Luisier et al., "Image denoising in mixed poisson–gaussian noise," IEEE Transactions on Image Processing, vol. 20, no. 3, pp. 696–708, 2011.

[31] J. Li et al., "Deconvolution of poissonian images with the pure-let approach," in 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 2708–2712.

[32] K. Kim et al., "Penalized pet reconstruction using deep learning prior and local linear fitting," IEEE Transactions on Medical Imaging, vol. 37, no. 6, pp. 1478–1487, 2018.

[33] J. Adler et al., "Deep bayesian inversion," 2018.

[34] N. Ducros et al., "Single-Pixel Image Reconstruction Toolbox (SPIRiT) Version 2.1," https://github.com/nducros/SPIRIT/, 2020.

[35] A. Lorente Mur et al., "Single-Pixel pYthon Image Reconstruction Toolbox (spyrit) Version 1.0," https://github.com/todo, 2020.

[36] M. Ochoa et al., "Assessing patterns for compressive fluorescence lifetime imaging," Opt. Lett., vol. 43, no. 18, pp. 4370–4373, Sep 2018.

[37] L. Baldassarre et al., "Learning-based compressive subsampling," IEEE Journal of Selected Topics in Signal Processing, vol. 10, no. 4, pp. 809–822, June 2016.

[38] F. Rousset et al., "Adaptive basis scan by wavelet prediction for single-pixel imaging," IEEE Transactions on Computational Imaging, vol. 3, no. 1, pp. 36–46, March 2017.

[39] A. Foi et al., "Practical poissonian-gaussian noise modeling and fitting for single-image raw-data," IEEE Transactions on Image Processing, vol. 17, no. 10, pp. 1737–1754, Oct. 2008.

[40] A. Lorente Mur et al., "Handling negative patterns for fast single-pixel lifetime imaging," in Molecular-Guided Surgery: Molecules, Devices, and Applications V, B. W. Pogue et al., Eds., vol. 10862, International Society for Optics and Photonics. SPIE, 2019, pp. 16 – 25.

[41] V. K. Nath et al., "Statistical distributions of discrete walsh hadamard transform coefficients of natural images," Signal & Image Processing, vol. 3, no. 1, p. 75, 2012.

[42] S. Särkkä, Bayesian Filtering and Smoothing, ser. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2013.

[43] A. Tarantola, Inverse Problem Theory and Methods for Model Parameter Estimation. Society for Industrial and Applied Mathematics, 2005.

[44] B. Zhu et al., "Image reconstruction by domain-transform manifold learning," Nature, vol. 555, p. 487, Mar. 2018.

[45] A. Coates et al., "An analysis of single-layer networks in unsupervised feature learning," in Proceedings of the fourteenth international conference on artificial intelligence and statistics, 2011, pp. 215–223.

[46] L. Azzari et al., "Variance stabilization for noisy+estimate combination in iterative poisson denoising," IEEE Signal Processing Letters, vol. 23, no. 8, pp. 1086–1090, 2016.

[47] A. Paszke et al., "Automatic differentiation in pytorch," 2017.

[48] D. P. Kingma et al., "Adam: A method for stochastic optimization," CoRR, vol. abs/1412.6980, 2014.

[49] A. Coates et al., "An analysis of single-layer networks in unsupervised feature learning," in Proceedings of the fourteenth international conference on artificial intelligence and statistics, 2011, pp. 215–223.

[50] Y. Cun et al., "Eigenvalues of covariance matrices: Application to neural-network learning," Physical Review Letters, vol. 66, no. 18, pp. 2396–2399, 1991.

[51] A. L. Mur et al., "Deep neural networks for single-pixel compressive video reconstruction," in Unconventional Optical Imaging II, C. Fournier et al., Eds., vol. 11351, International Society for Optics and Photonics. SPIE, 2020, pp. 71 – 80.

[52] Photography — Electronic still picture imaging — Resolution and spatial frequency responses, International Organization for Standardization, Geneva, CH, 2019, https://www.iso.org/standard/71696.html.

[53] K. H. Jin et al., "Deep convolutional neural network for inverse problems in imaging," IEEE Transactions on Image Processing, vol. 26, no. 9, pp. 4509–4522, 2017.