Single-Pixel Image Reconstruction from Experimental Data Using Neural Networks

Antonio Lorente Mur, Pierre Leclerc, Françoise Peyrin, Nicolas Ducros

Single-Pixel Image Reconstruction from Experimental Data Using Neural Networks

Antonio Lorente Mur,[1,†] Pierre Leclerc,[1] Françoise Peyrin [1], and Nicolas Ducros,[1,*]

[1] *Univ Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM Saint-Etienne, CNRS, Inserm, CREATIS UMR 5220, U1206, F-69621, Lyon, France*

[†] *Antonio.Lorente-Mur@creatis.insa-lyon.fr,* [*]*Nicolas.Ducros@insa-lyon.fr*

Abstract: Single-pixel cameras that measure image coefficients have various promising applications, in particular for hyper-spectral imaging. Here, we investigate deep neural networks that when fed with experimental data, can output high-quality images in real time. Assuming that the measurements are corrupted by mixed Poisson-Gaussian noise, we propose to map the raw data from the measurement domain to the image domain based on a Tikhonov regularization. This step can be implemented as the first layer of a deep neural network, followed by any architecture of layers that acts in the image domain. We also describe a framework for training the network in the presence of noise. In particular, our approach includes an estimation of the image intensity and experimental parameters, together with a normalization scheme that allows varying noise levels to be handled during training and testing. Finally, we present results from simulations and experimental acquisitions with varying noise levels. Our approach yields images with improved peak signal-to-noise ratios, even for noise levels that were foreseen during the training of the networks, which makes the approach particularly suitable to deal with experimental data. Furthermore, while this approach focuses on single-pixel imaging, it can be adapted for other computational optics problems.

1. Introduction

Single-pixel imaging is an extreme configuration of computational optics, where a single point detector is used to recover an image [1]. Since the seminal work by Duarte and coworkers [2], single-pixel imaging has been successfully applied to fluorescence microscopy [3], hyperspectral imaging [4, 5], diffuse optical tomography [6], image-guided surgery [7], short-wave infrared imaging [8], and imaging through scattering media [9]. 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 [1]. To limit acquisition times, it is highly desirable to reduce the number of light patterns, which leads to an undetermined inverse problem to recover the image.

In the field of single-pixel imaging, deep learning has been used to unmix the fluorescence intensity and lifetime from time-resolved measurements [10, 11]. Higham and coworkers [12] 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 followed by convolutional layers. Several Deep Learning architectures have been proposed since then, to solve single-pixel reconstruction problems. For instance, Li et al. in [9] used a similar network to that introduced by [12]. For measurements with very low signal-to-noise ratios, in [13] the authors proposed a U-net for highly compressed single-pixel Fourier imaging, while in [14], a recurrent neural network was used. Further, Li and coworkers [15] investigated a conditional generative adversarial network to reconstruct highly compressed data from measurements with random binary patterns.

In the present work, we propose a new deep learning methodology for image reconstruction from

single-pixel measurements corrupted by a mixed Poisson-Gaussian noise model. Traditionally, inverse problems with data corrupted by Poisson noise are tackled using variance stabilizing transforms, such as the Anscombe transform [16], followed by a Wiener filter [17]. However, the resulting image can be blurred, and in particular for undersampled data. More recent alternatives have been used to solve this issue by exploiting statistical or handmade image priors [18–20]. Solving the resulting problems is usually prohibitive for real-time applications, as a single reconstructed image can take several seconds. In this regard, deep networks are ideal candidates due to their rapid evaluation. As such, deep learning has been used for single-pixel imaging in the presence of Poisson noise [21], although no special modifications were made to the neural network to allow for the Poisson noise.

Many studies have explored many different neural network architectures; however, they usually consider convolutional layers that act in the image domain, while the raw data are in the measurement domain. Moreover, the interpretation of the output of a neural network is still an open question, specifically in the presence of noise where robustness issues can occur, as indicated by [22]. In particular, in the domain of single-pixel imaging, a neural network that is not finely tuned to the system can be outperformed by linear reconstructors [23].

1.1. Contribution

First, we explore the best way to map the raw data to the image domain, before applying a cascade of convolutional layers. This mapping is traditionally learned or computed through the Moore-Penrose pseudo-inverse. We describe a linear mapping that can be interpreted in terms of traditional image reconstruction. This linear mapping is implemented as the first layer of a (nonlinear) deep neural network. We introduced this idea in [24] for noiseless data, and here we extend it to noisy data. In particular, we propose to estimate the image intensity, which is unknown in practice, and to use this estimation to approximate the covariance matrix of the measurements.

Next, we describe a machinery that allows a network to be trained in the presence of Poisson noise. 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.

Finally, we validate our approach by reconstruction of an experimental dataset that we acquired with varying integration times and light fluxes. Upon acceptance, the datasets will be made available alongside implementations of our reconstruction methods in the Python toolbox (SPyRiT [25]).

2. Problem, assumptions and limitations

2.1. Single-pixel imaging

Let $\boldsymbol{f} \in [0,1]^N$ be the image to acquire. The main idea of compressive optics is to measure $\boldsymbol{m} = \boldsymbol{H}_1 \boldsymbol{f}$ using hardware, and to recover $\boldsymbol{f}$ using software. The system matrix $\boldsymbol{H}_1 \in \mathbb{R}^{M \times N}$, with $M < N$, collects the patterns that are sequentially uploaded on a spatial light modulator, to obtain the measurement vector. The patterns are traditionally chosen from within a basis $\boldsymbol{H} \in \mathbb{R}^{N \times N}$; i.e., $\boldsymbol{H}_1 = \boldsymbol{S}\boldsymbol{H}$ where $\boldsymbol{S} = [\boldsymbol{I}_M, \boldsymbol{0}]$ describes the subsampling strategy. Classical choices for the basis $\boldsymbol{H}$ include Fourier, discrete cosines, wavelets, and Hadamard basis, as discussed in [26]. In the present study, we consider the Hadamard basis. The subsampling strategies can be divided into adaptive and nonadaptive. Adaptive strategies progressively determine $\boldsymbol{S}$ during the acquisition [27], while nonadaptive chose $\boldsymbol{S}$ beforehand. Here, for simplicity, we consider the nonadaptive strategy introduced in [12, 28].

2.2. Acquisition model

As Hadamard patterns contain negative values, we adopt a differential strategy [29]. We denote by $\hat{\mathbf{m}}_{+}^{\alpha}$ the measurements of the positive parts of the patterns, and by $\hat{\mathbf{m}}_{-}^{\alpha}$ the measurements of the negative parts. We model noise as a mixture of Poisson and Gaussian distributions [30, 31]. The Poisson noise is signal dependant and originates from the discrete nature of the electronic charge, while the signal-independent Gaussian noise accounts for readout and circuit fluctuations. For both the positive and negative measurements, we have

$$\hat{\mathbf{m}}_{+,-}^{\alpha} \sim K\,\mathcal{P}(\alpha \boldsymbol{H}_1^{+,-}\boldsymbol{f}) + \mathcal{N}(\mu_{\text{dark}}, \sigma_{\text{dark}}^2) \tag{1}$$

where $\mathcal{P}$ and $\mathcal{N}$ are the Poisson and Gaussian distributions, K is a constant that represents the overall system gain (in counts/electron), α is the intensity (in photons) of the image (which is proportional to the integration time), μ_{dark} is the dark current (in counts), and σ_{dark} is the dark noise (in counts). We further hypothesize that μ_{dark} and σ_{dark} are independent of the image intensity α.

The normalized measurements $\mathbf{m}^{\alpha}$ are finally defined as

$$\mathbf{m}^{\alpha} = (\hat{\mathbf{m}}_{+}^{\alpha} - \hat{\mathbf{m}}_{-}^{\alpha})/(\alpha K), \tag{2}$$

2.3. Deep learning for image reconstruction.

Deep-learning-based methods reconstruct an image $\tilde{\boldsymbol{f}}$ using nonlinear mapping $\tilde{\boldsymbol{f}} = \mathcal{G}_{\omega}(\boldsymbol{m}^{\alpha})$ where the weights of the network $\boldsymbol{\omega}$ are optimized with respect to a loss functions; e.g., to minimize the quadratic error over an image database

$$\mathcal{G}_{\omega} = \underset{\boldsymbol{\Omega}}{\operatorname{argmin}} \frac{1}{s}\sum_{i=0}^{s-1} \|\mathcal{G}_{\boldsymbol{\Omega}}(\boldsymbol{m}_i^{\alpha}) - \boldsymbol{f}_i\|^2. \tag{3}$$

where $(\boldsymbol{f}_i)_{0\le i\le s-1}$ are the image samples of the image database, and $(\boldsymbol{m}_i^{\alpha})_{0\le i\le s-1}$ are the corresponding normalized noisy measurements given by Equation (2). Traditionally, neural networks are made of cascading layers, and can be written as

$$\mathcal{G}_{\omega} = \mathcal{G}_{\omega}^{L} \circ \ldots \circ \mathcal{G}_{\omega}^{1} \tag{4}$$

where $\mathcal{G}^{\ell}$, $1 \le \ell \le L$ is the ℓ-th (nonlinear) layer of the network, and $\circ$ is the function composition. The first layer is generally a fully connected layer that maps the normalized measurements $\boldsymbol{m}^{\alpha} \in \mathbb{R}^M$ to a raw solution in $\boldsymbol{f}^* \in \mathbb{R}^N$. In Section 3.2, we explore different strategies for the design of this layer.

3. Theory

3.1. Experimental set-up

Our measurements are obtained from the single-pixel camera experimental set-up depicted in Fig. 1, as first described by [32]. 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 2) 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 exemplar 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.

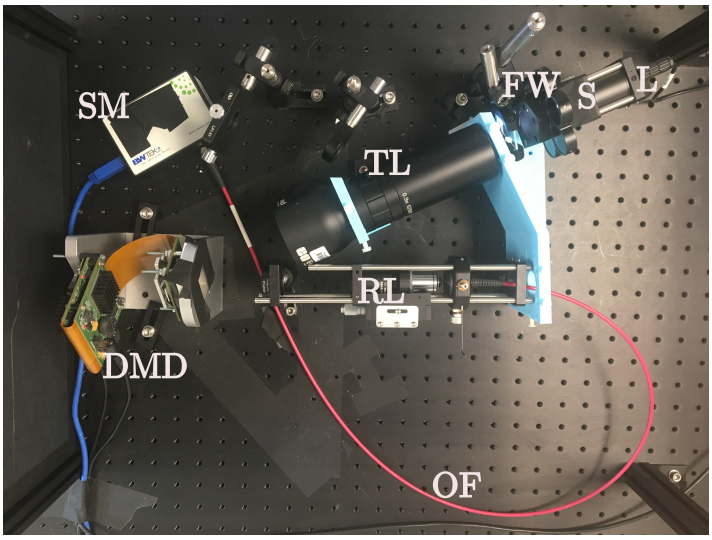

Fig. 1. Optical set-up of the single-pixel camera [2]. This set-up is composed of a sample (S) illuminated by a lamp (L) in front of a filter wheel (FW), a telecentric lens (TL), a digital micro-mirror device (DMD), some relay lenses (RL), an optical fiber (OF), and a spectrometer (SM).

3.2. *Mapping of the raw data to the image domain*

We propose to use a Tikhonov-regularized interpretable solution [33] to map the raw data into the image domain. In particular, we choose

$$\tilde{\boldsymbol{f}} = \mathcal{G}^{\dagger}(\hat{\boldsymbol{m}}^{\alpha}) = \boldsymbol{H}^{\top}\boldsymbol{y}^{*}, \tag{5}$$

where the regularized data $\boldsymbol{y}^{*}$ is such that $\boldsymbol{y}^{*} = [\boldsymbol{y}_1^{*}, \boldsymbol{y}_2^{*}]^{\top}$, where $\boldsymbol{y}_1^{*} \in \mathbb{R}^{M}$ and $\boldsymbol{y}_2^{*} \in \mathbb{R}^{(N-M)}$ are regularized versions of the acquired and missing coefficients, respectively.

Let $\boldsymbol{\Sigma}_{\alpha}$ be the variance of our noisy measurements, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are respectively the mean and variance of our prior model in the Hadamard domain. The computation of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ is described in Section 3.3, and that of $\boldsymbol{\Sigma}_{\alpha}$ in Section 3.4. We derived the analytical Tikhonov regularized solution given by

$$\boldsymbol{y}_1^{*}(\boldsymbol{m}^{\alpha}) = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_1[\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_{\alpha}]^{-1}(\boldsymbol{m}^{\alpha} - \boldsymbol{\mu}_1), \tag{6a}$$

$$\boldsymbol{y}_2^{*}(\boldsymbol{m}^{\alpha}) = \boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_1^{-1}[\boldsymbol{y}_1^{*}(\boldsymbol{m}^{\alpha}) - \boldsymbol{\mu}_1]. \tag{6b}$$

where $\boldsymbol{\Sigma}_1 \in \mathbb{R}^{M\times M}$, $\boldsymbol{\Sigma}_{21} \in \mathbb{R}^{(N-M)\times M}$ and $\boldsymbol{\Sigma}_2 \in \mathbb{R}^{(N-M)\times(N-M)}$ are the blocks of the covariance $\boldsymbol{\Sigma}$ in the measurement domain, $\boldsymbol{\mu}_1 \in \mathbb{R}^{M}$ and $\boldsymbol{\mu}_2 \in \mathbb{R}^{(N-M)}$ are the blocks of $\boldsymbol{\mu}$ (see Section 3.3). Interestingly, Equation (6a) can be interpreted as the denoising of the raw data in the measurement domain, while Equation (6b) is the estimation of the missing coefficients from the denoised acquired coefficients.

To circumvent the difficulty of inverting the signal-dependant matrix in Equation (6a), we choose to neglect the nondiagonal terms of $\boldsymbol{\Sigma}_1$;Denoting $\boldsymbol{\sigma}_1^2 = \mathsf{diag}\,(\boldsymbol{\Sigma}_1)$, we get

$$\boldsymbol{y}_1^{*}(\boldsymbol{m}^{\alpha}) = \boldsymbol{\mu}_1 + \boldsymbol{\sigma}_1^2/(\boldsymbol{\sigma}_1^2 + \boldsymbol{\sigma}_{\alpha}^2)(\boldsymbol{m}^{\alpha} - \boldsymbol{\mu}_1), \tag{7}$$

where division and multiplication apply element-wise.

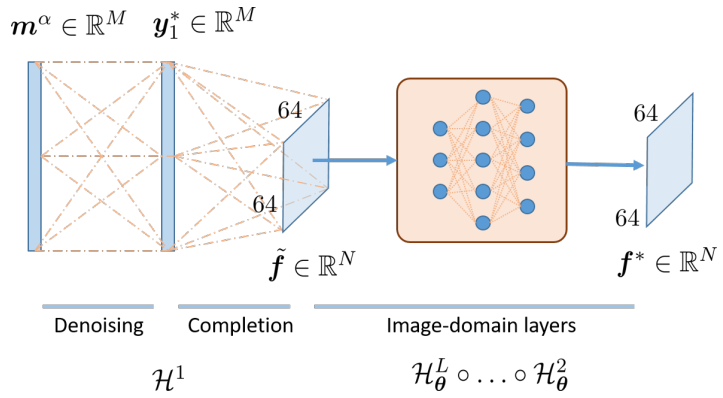

Fig. 2. Proposed network. The first two layers map the measurements into the image domain according to the analytical solution given by Equation (6). These two layers can be interpreted as a layer that denoises Equation (6a) of the raw measurements, followed by a layer that estimates the missing coefficients from the denoised measurements of Equation (6b). The raw image $\tilde{\boldsymbol{f}}$ is corrected by a cascade of Image-domain convolution layers (CL) and non-linear layers, such as the ReLU layers.

3.3. *Prior mean and covariance*

We compute the mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ as introduced in Equation (6) from the the samples of an image database. We consider the classical estimators of the mean and covariance

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix} = \frac{1}{S} \sum_{i=1}^{S} \boldsymbol{H} \boldsymbol{f}_i, \tag{8}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_{21}^\top \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_2 \end{bmatrix} = \frac{1}{S-1} \sum_{i=1}^{S} (\boldsymbol{H} \boldsymbol{f}_i - \boldsymbol{\mu})(\boldsymbol{H} \boldsymbol{f}_i - \boldsymbol{\mu})^\top. \tag{9}$$

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).

3.4. *Noise covariance estimation*

To use the mapping proposed in Section 3.2, we need an estimation of the covariance matrix of our measurements. Assuming that the raw measurements are independent, from Equation (2) we can determine the values of the covariance as

$$\boldsymbol{\Sigma}_\alpha = \mathsf{Diag}\left(\boldsymbol{\sigma}_\alpha^2\right) = \mathsf{Diag}\left(\frac{1}{\alpha} \boldsymbol{H}_1^+ \boldsymbol{f} + \frac{1}{\alpha} \boldsymbol{H}_1^- \boldsymbol{f}\right) + \frac{2\sigma_{\text{dark}}^2}{K^2 \alpha^2} \boldsymbol{I}, \tag{10}$$

where $\mathsf{Diag}\,(\boldsymbol{x})$ refers to the diagonal matrix where the diagonal coefficients are the elements of $\boldsymbol{x}$, and $\boldsymbol{I}$ refers to the identity matrix.

As $\boldsymbol{\sigma}_\alpha^2$ depends on the unknown image $\boldsymbol{f}$ as well as on the intensity α, we exploit the raw data that also depends on $\boldsymbol{f}$ and α. Recalling that the variance of a Poisson variable is the same as its expected value, we approximate the expected value by the noisy sample; i.e.,

$$\boldsymbol{\sigma}_\alpha^2 \approx \frac{1}{\alpha^2 K} (\hat{\boldsymbol{m}}_+^\alpha + \hat{\boldsymbol{m}}_-^\alpha - 2\mu_{\text{dark}}) + \frac{2\sigma_{\text{dark}}^2}{K^2 \alpha^2}. \tag{11}$$

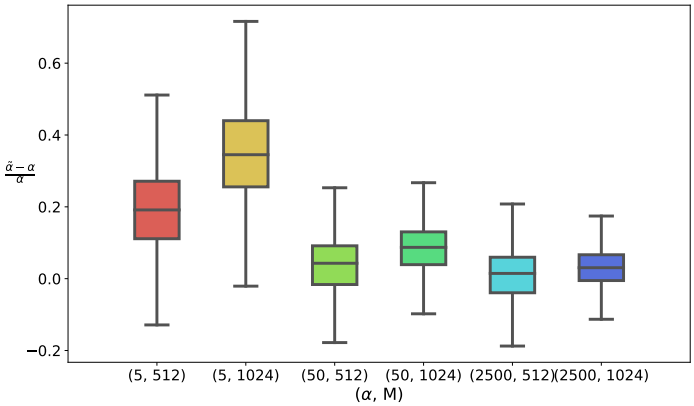

Fig. 3. Box plot of the distribution of the relative error on the estimation of α using Equation (12) with six combinations of values of (α, M).

The experimental parameters μ_{dark}, K and σ_{dark} can be estimated as described in [31]. We describe how to estimate α in Section 3.5.

3.5. Estimation of the image intensity

For image denoising, estimation of α can be achieved through fitting methods (e.g., see [30]). 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

$$\tilde{\alpha} = \max_{i \in \{1,\dots,N\}} (\boldsymbol{H}^\top \boldsymbol{y}^*)_i, \quad \text{with } \boldsymbol{y}^* = \begin{bmatrix} \hat{\boldsymbol{m}}^\alpha \\ \boldsymbol{0} \end{bmatrix} \tag{12}$$

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 [34] (i.e., 113,000 images obtained by merging the unlabeled, training and testing sets) for different values of M (512, 1024) and α (5, 50, 2500 photons). Overall, the relative error is usually below 50%.

4. Experiments

4.1. Training neural networks using simulated data

We train our network using the STL10 database [34], with $S = 105,000$ 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.

We implement the full networks using Pytorch [35] (version 1.5.1; cuda V10.2.89). We train the network by solving Equation (3) using the ADAM optimizer [36], 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]. As seen in Section 3.5, 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 5, 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 will further develop this point in Section 5.

Our experiments were done with the same Image domain layer structure as in [12]. This means that our network has three convolutional layers, with each layer separated by a ReLU layer. The first has a kernel size of 9 and a depth of 64, the second has a kernel size of 1 and a depth of 32, and the final one has a kernel size of 5 and a depth of 1. This structure can, however, be changed.

4.2. Experimental data

As shown in the first column of Fig. 4, 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) [37]. The ground-truth image is computed (see Section 4.3) 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).

Our estimated instrumental parameters were : $\mu_{\text{dark}} = 1070$ counts, $K = 1.54$ count/electron, and $\sigma_{\text{dark}} = 53$ counts.

4.3. 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

$$\mathsf{PSNR}(\boldsymbol{f}^*, \boldsymbol{f}) = 10 \log_{10} \frac{2^2}{\|\boldsymbol{f}^* - \boldsymbol{f}\|^2} \tag{13}$$

For the experimental data, we have no direct access to the ground truth image. This image is computed as $\boldsymbol{f} = \boldsymbol{H}^\top \boldsymbol{y}_{\text{gt}}$, where $\boldsymbol{y}_{\text{gt}}$ is a fully sampled measurement with high signal-to-noise ratio. Before computing the PSNR according to Equation (13), we normalize the ground-truth image in the range $[-1, 1]$.

5. Results

5.1. Simulated results

Training for different levels of noise In Table 1, we evaluate the effects of training a network under varying noise levels. We also simulate the acquisition of the test images with the same source intensity ($\alpha = 50$ photons) and for varying intensities (mean values $\mu_\alpha = 50$ 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 1).

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 3.5), 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 1. Training and testing under varying noise levels. Top row: Data simulated for a given source intensity (α = 50 photons), which is the same for all of the test images. Bottom row: Data simulated for test images with varying intensities (mean μ_α = 50 photons, standard deviation σ_α = 25 photons). Reconstruction PSNRs are reported for a network trained using constant intensity (middle column) and varying intensity (right column).

In Table 2, we report the reconstruction PSNRs obtained using the proposed network, and for a network with the same architecture where the mapping is learned ('Free Layer') for different levels of noise dictated by different image intensities α ($\alpha = \infty$ corresponds to the noiseless version of the proposed network). The standard deviation of the image intensity is set to 50%, in agreement with the findings of Section 3.5.

As expected, the network trained with no noise (i.e., $\alpha = \infty$) performs very poorly in the presence of noise. Therefore, we focus our analysis on the cases where our network is trained with noise. We divide our analysis into three cases, which depend on the relative noise levels used during training and testing.

Training noise and testing noise have the same levels In Table 2, the PSNRs of these experiments are shown in blue (i.e., diagonals). In all of our simulations, the proposed method outperforms the 'Free layer' network proposed in [12].

Training noise is higher than testing noise In Table 2, the PSNRs of these experiments are shown in red (below lower diagonal). We can observe that networks trained on high levels of noise (e.g., α = 2 or 5 photons) with our proposed method perform poorly when they are tested on data with low levels of noise (e.g., α = 50 or 2500 photons). For instance, a neural network trained with α = 2 photons yields an average PSNR of 17.40 dB when it is tested for α = 2500 photons. This is a PSNR drop of 4.77 dB compared to the optimal training conditions ($\alpha_{\text{train}} = \alpha_{\text{test}}$ = 2500 photons). We can also observe that in these cases the 'Free Layer' [12] outperforms the proposed method. We can see that in the most extreme cases where the training was with very high levels of noise, and the testing was with very low levels of noise, the 'Free Layer' network can increase the PSNR by 20.09-17.40 = 2.69 dB on average.

Training noise is lower than testing noise In Table 2, the PSNRs of these experiments are shown in green (above diagonals). We can see that the proposed network behaves similarly to the optimal training conditions. The worst drop in PSNR is relatively low, as 0.71 dB. This shows that the proposed network has high reconstruction quality provided that it is trained with relatively low levels of noise (e.g., $\alpha = 50$ or 2500 photons). Contrary to the previous scenario, the proposed method outperforms the 'Free Layer' network in all of these experiments. Here, the presence of the denoising layer has a great advantage; the proposed method generalizes better to high noise experiments.

Testing		Training				
	α (in ph.)	∞	2 ± 1	10 ± 5	50 ± 25	2500 ± 1250
Proposed	2 ± 1	9.48 ± 1.78	18.79 ± 1.47	**18.60 ± 1.55**	**18.31 ± 1.55**	**18.14 ± 1.55**
	10 ± 5	13.58 ± 1.94	19.88 ± 1.33	**20.82 ± 1.51**	**20.55 ± 1.61**	**20.21 ± 1.62**
	50 ± 25	15.32 ± 2.06	18.68 ± 1.43	**21.04 ± 1.39**	**21.86 ± 1.54**	**21.69 ± 1.61**
	2500 ± 1250	15.88 ± 2.11	17.40 ± 1.71	19.94 ± 1.59	**21.69 ± 1.51**	**22.17 ± 1.56**
Free Layer	2 ± 1		18.72 ± 1.44	17.93 ± 1.35	17.35 ± 1.37	16.99 ± 1.37
	10 ± 5		19.80 ± 1.64	20.10 ± 1.48	20.04 ± 1.45	19.91 ± 1.42
	50 ± 25		**20.03 ± 1.71**	20.63 ± 1.58	20.81 ± 1.56	20.75 ± 1.53
	2500 ± 1250		**20.09 ± 1.72**	**20.76 ± 1.61**	21.02 ± 1.60	20.97 ± 1.58

Table 2. Reconstruction of peak signal-to-noise ratios (PSNRs) for different training strategies. 'Proposed' refers to the neural network method with the proposed mapping in Section 3. 'Free Layer' refers to the neural network method where the mapping of the raw measurements is learnt jointly with the postprocessing layers, as in [12]. Note that the network trained with no noise ($\alpha = \infty$) corresponds to the case where we choose $\boldsymbol{\Sigma}_\alpha = \mathbf{0}$. 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. Blue font, the testing and training noise levels are the same; green font, the testing noise is lower than the training noise; and red font, the testing noise is higher than the training noise. 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).

5.2. Experimental Data

Fig. 4 illustrates the performance of our networks on experimental data acquired with the experimental set-up of a single-pixel camera presented in Section 3.1. We compare the performance of our proposed method with the result of the total variation regularized solution [38], the Tikhonov-regularized solution of Equation (6), the proposed method trained without noise ('Noiseless Net'), and with noise ('Proposed'), as well as the Tikhonov-regularized solution of Equation (6) to which we applied a state-of-the-art denoising method BM3D [39] ('Tikhonov+bm3d'), and a neural network reconstructor with the same architecture as the proposed method but where the mapping is learned jointly with the postprocessing layer ('Free

Layer'), as in [12].

First, we observe that the network trained with no noise performs very poorly on noisy data, and it is outperformed by the pseudo-inverse in the case of the LED lamp and the STL-10 cat. Visually, the reconstructed image retains many of the artifacts introduced by Poisson noise, which emphasizes the need for accounting for noise when training a neural network. Furthermore, we observe that learning the fully connected layer leads to many artifacts and a loss of detail (see the star sector target with the 'Free Layer'). This can be attributed to the learning of the mapping for specific noise values, instead of adapting to the noise, as in our proposed method. We also observe that the Tikhonov-regularized reconstruction is itself quite powerful. Even if it keeps some of the reconstruction artifacts, it compares very well to the results obtained using total variation regularization. In terms of PSNRs, Tikhonov-regularized reconstruction performs very similarly to the 'Free Layer' neural network, which is strong motivation to use it for mapping to the image domain.

Finally, we observe that the proposed network and 'Tikhonov+bm3d' offer the smoothest reconstructed images that are almost free of the artifacts introduced by down-sampling or Poisson noise. In terms of PSNRs, we observe that our method outperforms the others on experimental data, while being similar to 'Tikhonov+bm3d'. However, the proposed method is 1,000 times faster than the latter (0.2 seconds for BM3D versus 0.2 milliseconds for the proposed network). Moreover, 'Tikhonov+bm3d' is a two-step method where denoising is agnostic to reconstruction, while our method solves both problems at once, adapting denoising to the noise level. Although the PSNR gain is relatively low compared to the Tikhonov-regularized reconstruction, the proposed network leads to images that are well enhanced visually.

6. Discussion

The proposed mapping generalises much better to higher levels of noise than the learned mapping proposed in [12]. This is an advantage when dealing with experimental data in real time, as it can be time consuming to dynamically load different neural networks for different lighting conditions. In experimental scenarios, we do not know α, and therefore our proposed mapping with a training value of $\alpha = 2500 \pm 1250$ is the most suited for processing experimental data. Indeed, for all of the testing conditions, that neural network offers a very similar quality of reconstruction to the optimally trained neural network for that noise value.

Our method allows visually convincing reconstructions and good performance with respect to PSNR against several levels of noise, when trained for low levels of noise (high values of α). It therefore shows good proprieties for dealing with experimental data. Unlike previous single-pixel deep-learning studies, such as [12–14, 38], we have showcased the robustness of our method to different lighting conditions, and given an interpretable meaning to the first layer of our neural network.

The noise level of the image under acquisition is a key feature in real-life experiments. This parameter can be estimated first, and then the network that fits the actual noise level can be evaluated. However, as the noise level is unknown, this 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 all stored on the graphics processor unit. Our proposed method is robust to different levels of noise as long as it was trained with low levels of noise (high values of α). Therefore, the same network can be used almost optimally in a wide variety of situations. The Tikhonov-regularized mapping that we propose helps to denoise the raw data, to provide an approximate reconstruction to the convolutional layers that can focus on learning spatial features.

One limitation of our deep network compared to more classic approaches such as [18, 20, 40] is that there is no theoretical guarantee that it will work for any image; in particular, if the image under acquisition significantly differs from those of our training set. This is a common concern for deep-learning approaches that are, however, seen to work well in practice. While this tends to

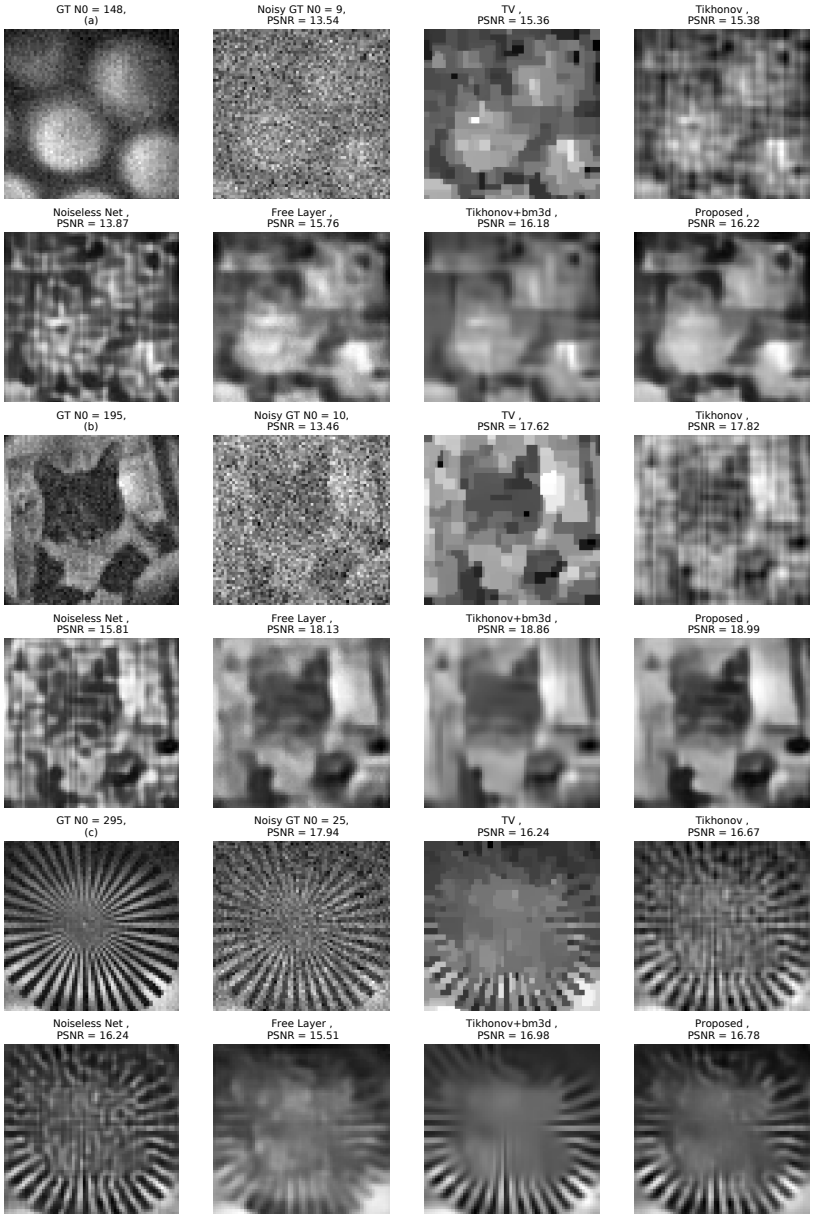

Fig. 4. Reconstructions of three experimental datasets by the different methods (top row: LED lamp with $M = 512$; middle row: STL-10 cat with $M = 512$; bottom row: Siemens Star resolution target with $M = 1024$). We display the images reconstructed from a fully sampled dataset (ground-truth; GT) acquired with high image intensity (first column, $\alpha = 148$ photons, $\alpha = 195$ photons and $\alpha = 295$ photons for the LED lamp, STL-10 cat and Siemens Star resolution target, respectively) and lower image intensity ('Noisy GT' second column, $\alpha = 9$ photons, $\alpha = 10$ photons and $\alpha = 25$ photons for the LED lamp, STL-10 cat and Siemens Star resolution target, respectively). The following columns show reconstructions using the total variation regularized solution [38], the Tikhonov-regularized solution of Equation (7), the noiseless proposed network (Noiseless Net) Section 3.2 (network trained with no noise and where we assume $\boldsymbol{\Sigma}_\alpha = \mathbf{0}$), a Deep neural network with the same architecture as our proposed method, where the mapping is learned as in [12] (Free Layer), the Tikhonov-regularized method combined with BM3D [39] denoising and the proposed network Section 3.2 (trained with $\mu_\alpha = 50$ photons and $\sigma_\alpha = 0.5\mu_\alpha$). All of the PSNRs are computed as described in Section 4.3, with the first column as the ground-truth.

be confirmed by our experimental results where our approach worked on the Siemens Star sector and on a LED light (both of which are very different from the images of the stl-10 database), there are no theoretical guarantees that this will always be the case.

Another limiting aspect of our study might be the choice of our architecture, which is shallower than popular architectures, such as the U-Net used in [41], 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 study 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 $N > 64$, either directly or by implementing a patch-based strategy to limit the memory requirements.

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., [42]). This will be the object of future work. Although this work focuses on single-pixel imaging, it can be used for any linear problem where the measured data is scarcely sampled in an orthogonal basis, just by setting $\boldsymbol{H}$ to the corresponding basis. Finally, 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.

7. Conclusion

We present a deep learning method to recover an image in single-pixel imaging by introducing mapping of the raw data to the image domain. This mapping is implemented as the first layer of a neural network, which provides an approximate reconstruction to a traditional cascade of convolutional layers. We also describe a framework for training a network using simulated data only. We describe how to exploit all of the experimental parameters to deal with experimental data, and also how to normalize the measurement, which is fundamental when a nonlinear reconstructor (e.g., neural network) is considered.

Although our network is trained using simulations only, it performs very well on experimental data, even when the noise level is unknown. The estimation of the noise covariance coupled with an appropriate training process appears to be efficient in a wide variety of scenarios. This is particularly interesting in real-time configurations where it is not possible to evaluate many different reconstructors. In future work, we will explore more complex interpretable network architectures.

Funding. This study 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.

Acknowledgments. This material is based upon work done on the PILoT facility (PILoT, INSA LYON, Bât. Blaise Pascal, 7 Av. Jean Capelle 69621 Villeurbanne).

Disclosures. The authors declare that they have no conflicts of interest.

Data availability. Data underlying the results presented in this paper are available in Ref. [43].

References

1. M. P. Edgar, G. M. Gibson, and M. J. Padgett, "Principles and prospects for single-pixel imaging," Nat. Photonics **13**, 13–20 (2019).
2. M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," Signal Process. Mag. IEEE **25**, 83–91 (2008).

3. V. Studer, J. Bobin, M. Chahid, H. S. Mousavi, E. Candes, and M. Dahan, "Compressive fluorescence microscopy for biological and hyperspectral imaging," Proc. Natl. Acad. Sci. **109**, E1679–E1687 (2012).
4. F. Rousset, N. Ducros, F. Peyrin, G. Valentini, C. D'Andrea, and A. Farina, "Time-resolved multispectral imaging based on an adaptive single-pixel camera," Opt. Express **26**, 10550–10558 (2018).
5. G. R. Arce, D. J. Brady, L. Carin, H. Arguello, and D. S. Kittle, "Compressive coded aperture spectral imaging: An introduction," IEEE Signal Process. Mag. **31**, 105–115 (2014).
6. Q. Pian, R. Yao, N. Sinsuebphon, and X. Intes, "Compressive hyperspectral time-resolved wide-field fluorescence lifetime imaging." Nat. photonics **11**, 411–414 (2017).
7. E. Aguénounon, F. Dadouche, W. Uhring, N. Ducros, and S. Gioux, "Single snapshot imaging of optical properties using a single-pixel camera: a simulation study," J. Biomed. Opt. **24**, 1 – 6 (2019).
8. Y. Zhang, G. M. Gibson, M. P. Edgar, G. Hammond, and M. J. Padgett, "Dual-band single-pixel telescope," Opt. Express **28**, 18180–18188 (2020).
9. F. Li, M. Zhao, Z. Tian, F. Willomitzer, and O. Cossairt, "Compressive ghost imaging through scattering media with deep learning," Opt. Express **28**, 17395–17408 (2020).
10. R. Yao, M. Ochoa, P. Yan, and X. Intes, "Net-flics: fast quantitative wide-field fluorescence lifetime imaging with compressed sensing - a deep learning approach," Light. Sci. & Appl. **8**, 26 (2019).
11. J. T. Smith, M. Ochoa, and X. Intes, "Unmix-me: spectral and lifetime fluorescence unmixing via deep learning," Biomed. Opt. Express **11**, 3857–3874 (2020).
12. C. F. Higham, R. Murray-Smith, M. J. Padgett, and M. P. Edgar, "Deep learning for real-time single-pixel video," Sci. Reports **8** (2018).
13. S. Rizvi, J. Cao, K. Zhang, and Q. Hao, "Deringing and denoising in extremely under-sampled fourier single pixel imaging," Opt. Express **28**, 7360–7374 (2020).
14. I. Hoshi, T. Shimobaba, T. Kakue, and T. Ito, "Single-pixel imaging using a recurrent neural network combined with convolutional layers," Opt. Express **28**, 34069–34078 (2020).
15. J. Li, Y. Li, J. Li, Q. Zhang, and J. Li, "Single-pixel compressive optical image hiding based on conditional generative adversarial network," Opt. Express **28**, 22992–23002 (2020).
16. F. J. Anscombe, "The transformation of Poisson, binomial and negative-binomial data," Biometrika **35**, 246–254 (1948).
17. N. Wiener, Extrapolation, interpolation, and smoothing of stationary time series (John Wiley & Sons, Inc, New York, 1949).
18. S. Lefkimmiatis and M. Unser, "Poisson image reconstruction with hessian schatten-norm regularization," IEEE Transactions on Image Process. **22**, 4314–4327 (2013).
19. Q. Ding, Y. Long, X. Zhang, and J. A. Fessler, "Statistical image reconstruction using mixed poisson-gaussian noise model for x-ray ct," arXiv: Med. Phys. (2018).
20. J. Li, F. Luisier, and T. Blu, "Deconvolution of poissonian images with the pure-let approach," in 2016 IEEE International Conference on Image Processing (ICIP), (2016), pp. 2708–2712.
21. X. Liu, J. Shi, L. Sun, Y. Li, J. Fan, and G. Zeng, "Photon-limited single-pixel imaging," Opt. Express **28**, 8132–8144 (2020).
22. S. Y. Kim, J. Lim, T. Na, and M. Kim, "3dsrnet: Video super-resolution using 3d convolutional neural networks," CoRR **abs/1812.09079** (2018).
23. S. Jiao, Y. Gao, J. Feng, T. Lei, and X. Yuan, "Does deep learning always outperform simple linear regression in optical imaging?" Opt. Express **28**, 3717–3731 (2020).
24. N. Ducros, A. Lorente Mur, and F. Peyrin, "A completion network for reconstruction from compressed acquisition," in 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), (2020), pp. 619–623.
25. A. Lorente Mur, F. Peyrin, and N. Ducros, "Single-Pixel pYthon Image Reconstruction Toolbox (spyrit) Version 1.0," https://github.com/openspyrit/spyrit (2020).
26. M. Ochoa, Q. Pian, R. Yao, N. Ducros, and X. Intes, "Assessing patterns for compressive fluorescence lifetime imaging," Opt. Lett. **43**, 4370–4373 (2018).
27. F. Rousset, N. Ducros, A. Farina, G. Valentini, C. D'Andrea, and F. Peyrin, "Adaptive basis scan by wavelet prediction for single-pixel imaging," IEEE Transactions on Comput. Imaging **3**, 36–46 (2017).
28. L. Baldassarre, Y. Li, J. Scarlett, B. Gözcü, I. Bogunovic, and V. Cevher, "Learning-based compressive subsampling," IEEE J. Sel. Top. Signal Process. **10**, 809–822 (2016).
29. A. Lorente Mur, M. Ochoa, J. E. Cohen, X. Intes, and N. Ducros, "Handling negative patterns for fast single-pixel lifetime imaging," in SPIE Photonics : Molecular-Guided Surgery: Molecules, Devices, and Applications V, vol. 10862 (2019).
30. A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical poissonian-gaussian noise modeling and fitting for single-image raw-data," IEEE Transactions on Image Process. **17**, 1737–1754 (2008).
31. M. Rosenberger, C. Zhang, P. Votyakov, M. Preißler, R. Celestre, and G. Notni, "Emva 1288 camera characterisation and the influences of radiometric camera characteristics on geometric measurements," ACTA IMEKO **5**, 81 (2016).
32. A. Lorente Mur, B. Montcel, F. Peyrin, and N. Ducros, "Deep neural networks for single-pixel compressive video reconstruction," in Unconventional Optical Imaging II, vol. 11351 C. Fournier, M. P. Georges, and G. Popescu, eds., International Society for Optics and Photonics (SPIE, 2020), pp. 71 – 80.
33. A. Tarantola, Inverse Problem Theory and Methods for Model Parameter Estimation (Society for Industrial and

Applied Mathematics, 2005).
34. A. Coates, A. Ng, and H. Lee, "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.
35. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in Advances in Neural Information Processing Systems 32, (Curran Associates, Inc., 2019), pp. 8024–8035.
36. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," CoRR **abs/1412.6980** (2014).
37. International Organization for Standardization, Geneva, CH, Photography — Electronic still picture imaging — Resolution and spatial frequency responses (2019). https://www.iso.org/standard/71696.html.
38. C. Li, "An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing," Ph.D. thesis, Rice University (2010).
39. K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," IEEE Transactions on Image Process. **16**, 2080–2095 (2007).
40. F. Luisier, T. Blu, and M. Unser, "Image denoising in mixed poisson–gaussian noise," IEEE Transactions on Image Process. **20**, 696–708 (2011).
41. K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," IEEE Transactions on Image Process. **26**, 4509–4522 (2017).
42. H. K. Aggarwal, M. P. Mani, and M. Jacob, "Modl: Model-based deep learning architecture for inverse problems," IEEE Transactions on Med. Imaging **38**, 394–405 (2019).
43. N. Ducros and A. Lorente Mur, "Single-Pixel Hyperspectral Imaging dataset Version 1.0," https://gitlab.in2p3.fr/nicolas.ducros/spihim (2020).