PAPER • OPEN ACCESS

Neural blind deconvolution with Poisson data

To cite this article: A Benfenati *et al* 2023 *Inverse Problems* **39** 054003

View the article online for updates and enhancements.

Neural blind deconvolution with Poisson data

A Benfenati[1,*] , A Catozzi[2] and V Ruggiero[3]

[1] Dipartimento di Scienze e Politiche Ambientali. Università di Milano, Via Celoria, 2, 20133 Milano, Italy
[2] Dipartimento di Matematica, Fisica e Informatica. Università di Parma, Parco Area delle Scienze, 7/A, 43124 Parma, Italy
[3] Dipartimento di Matematica e Informatica, Università di Ferrara, Via Machiavelli, 30, 44121 Ferrara, Italy

E-mail: alessandro.benfenati@unimi.it

CrossMark

Abstract

Blind Deconvolution problem is a challenging task in several scientific imaging domains, such as Microscopy, Medicine and Astronomy. The Point Spread Function inducing the blur effect on the acquired image can be solely approximately known, or just a mathematical model may be available. Blind deconvolution aims to reconstruct the image when only the recorded data is available. In the last years, among the standard variational approaches, Deep Learning techniques have gained interest thanks to their impressive performances. The Deep Image Prior framework has been employed for solving this task, giving rise to the so-called neural blind deconvolution (NBD), where the unknown blur and image are estimated via two different neural networks. In this paper, we consider microscopy images, where the predominant noise is of Poisson type, hence signal-dependent: this leads to consider the generalized Kullback–Leibler as loss function and to couple it with regularization terms on both the blur operator and on the image. Furthermore, we propose to modify the standard NBD formulation problem, by including for the blur kernel an upper bound which depends on the optical instrument. A numerical solution is obtained by an alternating Proximal Gradient Descent-Ascent procedure, which results in the Double Deep Image Prior for Poisson noise algorithm.

* Author to whom any correspondence should be addressed.

We evaluate the proposed strategy on both synthetic and real-world images, achieving promising results and proving that the correct choice of the loss and regularization functions strongly depends on the application at hand.

(Some figures may appear in colour only in the online journal)

1. Introduction

Blind deconvolution (BD) consists in a image deblurring problem when the point spread function (PSF) is not known or only approximately known, e.g. only a rough approximation from observation or a mathematical model, containing a limited number of unknown parameters, is known. For a linear model of the acquisition process, the näive formulation of the BD is to solve the equation $\mathbf{g} = \mathbf{h} * \mathbf{x}$, where $*$ denotes the convolution product and both the PSF $\mathbf{h}$ and the object $\mathbf{x}$ are unknown and must be estimated from the recorded image $\mathbf{g}$. A convenient and realistic assumption is to consider a space-invariant model for the PSF. Nevertheless, the problem is extremely under-determined and there exists an infinite set of solutions. One of them is the trivial one, i.e. $\mathbf{x} = \mathbf{g}$ and $\mathbf{h} = \boldsymbol{\delta}$, where $\boldsymbol{\delta}$ is the Dirac's delta, which is 1 at the center and zero elsewhere. Moreover, if the pair $\{\tilde{\mathbf{h}}, \tilde{\mathbf{x}}\}$ is a solution and R is an injective linear operation commuting with the cyclic convolution, then the pair $\{R\tilde{\mathbf{h}}, R^{-1}\tilde{\mathbf{x}}\}$ is also a solution.

BD is the subject of a wide literature and the different approaches concern specific classes of images and PSFs. For instance, approaches applicable to natural images may not be suitable in Microscopy or Astronomy; approaches developed for motion blur are not applicable to other classes of blur, and so on [1]. The obvious reason is that, since the problem is extremely ill-posed, specific kinds of prior knowledge must be introduced, for both the object and the PSF, in order to reduce the class of possible solutions [2]; as concerns natural images, a survey paper [3] contains a critical analysis of proposed methods as well as several relevant references.

In this paper we consider images acquired by a confocal microscopy, restricting the discussion to the methods of BD proposed in the framework of Poisson data. The model of the acquisition process becomes then

$$\mathbf{g} = P(\mathbf{h} * \mathbf{x} + b) \tag{1}$$

where the acquired image $\mathbf{g}$ is a realization of a Poisson multivalued random variable whose average is $\mathbf{h} * \mathbf{x} + b$; here b is a background function, usually a given constant function[4]. For a description of a confocal microscopy system and the related mathematical modelling of the imaging process, see [1]. We recall that, in the Bayesian approach, the PSF $\mathbf{h}^*$ and the object $\mathbf{x}^*$ are considered realizations of multi-valued random variables, respectively $\mathcal{H}$ and $\mathcal{X}$, so that the conditional probability of $\mathbf{h}, \mathbf{x}$ for a given value $\mathbf{g}$ of the random variable $\mathcal{G}$ of the image domain is given by

$$\mathbb{P}_{\mathcal{H},\mathcal{X}}(\mathbf{h}, \mathbf{x}|\mathbf{g}) = \frac{\mathbb{P}_{\mathcal{G}}(\mathbf{g}|\mathbf{h}, \mathbf{x})\mathbb{P}_{\mathcal{H}}(\mathbf{h})\mathbb{P}_{\mathcal{X}}(\mathbf{x})}{\mathbb{P}_{\mathcal{G}}(\mathbf{g})}, \tag{2}$$

where $\mathbb{P}_{\mathcal{H}}(\mathbf{h})$ and $\mathbb{P}_{\mathcal{X}}(\mathbf{x})$ are the priors of the PSF and object, respectively. By assuming Gibb's priors and replacing $\mathbb{P}_{\mathcal{G}}(\mathbf{g}|\mathbf{h}, \mathbf{x})$ with the likelihood function, i.e. the Poisson probability

---

[4] In microscopy a background emission can arise from auto-fluorescence, inadequate removal of fluorescence staining material, offset levels of the detector gain or other electronic sources [4, 5].

distribution of $\mathcal{G}$ for the realization $\mathbf{g}$, the maximum a posteriori (MAP) estimates $\bar{\mathbf{h}}, \bar{\mathbf{x}}$ of $\mathbf{h}^*$ and $\mathbf{x}^*$, respectively, can be obtained by maximizing (2) with respect to $\mathbf{h}$ and $\mathbf{x}$, or equivalently by minimizing the negative logarithm of the conditional probability $\mathbb{P}_{\mathcal{H}, \mathcal{X}}(\mathbf{h}, \mathbf{x} | \mathbf{g})$:

$$\min_{\mathbf{h} \in C_h, \mathbf{x} \in C_x} f^B(\mathbf{h}, \mathbf{x}; \mathbf{g}) \equiv f_0^B(\mathbf{h}, \mathbf{x}; \mathbf{g}) + f_1(\mathbf{h}) + f_2(\mathbf{x}). \tag{3}$$

Under the Poisson noise framework, the data-fidelity function $f_0^B(\mathbf{h}, \mathbf{x}; \mathbf{g})$ is the generalized Kullback–Leibler (KL) divergence, expressed as

$$\begin{aligned} f_0^B(\mathbf{h}, \mathbf{x}; \mathbf{g}) &= \mathrm{KL}(\mathbf{g}; \mathbf{h} * \mathbf{x} + b) \\ &= \sum_i g_i \ln \left(\frac{g_i}{(\mathbf{h} * \mathbf{x} + b)_i} \right) + (\mathbf{h} * \mathbf{x} + b)_i - g_i, \end{aligned} \tag{4}$$

where $g_i \ln g_i = 0$ if $g_i = 0$. Furthermore, f_1 and f_2 are the regularization functions of the PSF and object, respectively. In order to devise a meaningful solution between spurious local minima, simple physical constraints C_h and C_x for the two blocks of variables can be specified, as, for example, $C_h = \{\mathbf{h} \in \mathbb{R}^{d_h} : 0 \leqslant h_i, \sum_{i=1}^{d_h} h_i = 1\}$ and $C_x = \{\mathbf{x} \in \mathbb{R}^{d_x} : 0 \leqslant x_i, i = 1, \ldots, d_x\}$.

In any case we have to solve a constrained and non-convex minimization problem. Assuming that f_1 and f_2 are convex functions, since the objective function is convex with respect to each blocks of variables, keeping the other fixed, a standard approach to the solution of (3) is the so-called *alternating optimization method*, also known as *non-linear block Gauss-Seidel* or *block coordinate descent method* ([6, chapter 2]); it consists in solving problem (3) by successively minimizing the objective function with respect to each block of variables, over the corresponding constraint set, by keeping the other fixed. Remarkable convergence results are in [7] under the assumption that the exact solutions of the two minimization subproblems can be obtained at any iteration and, above all, in [8] where the stationary of limit points of the sequence of inexact solutions is stated. This *inexact* block coordinate descent approach, where at each iteration a gradient projection step based on an Armijo line-search along the feasible direction with variable step-size is performed for any block, is very useful in the practical applications (see for example [9]). In the deterministic framework, many advances have been made to deal with BD problems in the case of natural images degraded by additive noise, by introducing novel regularization terms or suitable strategies to improve the alternating schemes (see, for example [10–13] and references therein).

Recently, deep convolutional neural networks (CNNs) trained on large datasets are used to predict either the blur or the sharp image directly from the blurred examples (see [14–17] and reference therein, and the surveys [18, 19]). In [20], the BD problem is addressed by training two separate generative networks, the first to produce sharp images and the second to generate blur kernels. Then, the BD problem is formulated as a minimization problem with respect to the inputs of the two pre-trained networks which provide as output the recovered image and the blur kernel; employing suitable regularization terms, this problem is solved by an alternating gradient descent scheme. A further proposal is to train only the network related to the blur kernel, using an unsupervised technique for training the network for the image. This technique is very similar to the one proposed in [21] and described in the upcoming sections. We highlight that these data-driven methods are limited by the capacity of the training datasets and then they exhibit a lack of generalization when not trained with enough data. Moreover, in many real applications, it is practically impossible to build a sufficiently large dataset with both ground truth and degraded data.

In an unsupervised deep learning framework for image restoration, the seminal papers [22, 23] propose a Deep Image Prior (DIP) approach, exploiting the fact that the structure of a deep

CNN generator is sufficient to capture a large amount of low-level image statistics prior to any learning, thus inducing an implicit regularization. Specifically, the visual object of interest can be parameterized as the output of a neural network. Borrowing the DIP approach, it is natural to parameterized both the estimated image and the blur separately by generative networks, following the Double-DIP approach in [24]. These nets are not pre-trained on any datasets; they are adopted merely to capture the priors of either the image and the blur for the optimization process, performing a so-called neural blind deconvolution (NBD). Indeed, this strategy is adopted by [21, 25]; the two approaches differ for the structure of the two nets. In particular in [25] the two nets are convolutional encoder-decoder networks, while in [21] only the image is the output of a convolutional U-net and the blur kernel is generated by a more simple two-layers FC net. This last method is called SelfDeblur and it is able to produce high-quality results for blur images, but multiple runs can produce outputs with a wide range in quality. This is due to the randomly initialization of the neural nets, giving rise to a non-deterministic methods. In [26] the authors improve the method in terms of consistency, by introducing a suitable initialization of the nets, multiscale processing and regularization. In [27], others techniques are discussed to address blurred and noisy images, also in presence of kernel with unknown size.

The contribution of this paper is to tailor the ideas developed in [21, 26, 27] for NBD in confocal microscopy. In particular, the PSF **h** and the image **x** are derived from the parametric outputs of two generative networks $\mathcal{N}_h(\boldsymbol{\theta}_h)$ and $\mathcal{N}_x(\boldsymbol{\theta}_x)$; the problem (3) is reformulated so that the minimization is performed with respect to the weights $\boldsymbol{\theta}_h, \boldsymbol{\theta}_x$ of the two nets. For modeling **x**, an autoencoder with skip connections $\mathcal{N}_x$ can be selected, so that $\mathbf{x} = \mathcal{N}_x(\boldsymbol{\theta}_x)$ while, for the more simple blur kernel **h**, an FC network can be adopted, although the activation functions are sin functions and not standard ones. The input data of the two nets are random tensors. Following the suggestion in [9] (see also [1, section 8.3]), in order to reduce the set of possible solutions of the NBD problem, we introduce a further constraint provided by the knowledge of the Strehl ratio (SR) of the optical instrument. We recall that the SR is the ratio between the maximum value of an aberrated PSF versus that of a perfect PSF. This information implies an upper bound on the PSF, in addition to the lower bound of non-negativity and to the normalization; therefore, with such a constraint, the trivial solution provided by the $\boldsymbol{\delta}$-array as PSF is not allowed. Suitable regularizations are introduced in the reformulated objective function. In view of a sharp image restoration, we employ an edge-preserving non-smooth term, such as a Total Variation-like function. We restate the minimization problem in the form of a nonconvex-concave min-max problem, similarly to what is described in [28]. For its numerical solution, we develop a tailored version of the alternating proximal gradient descent-ascent (PGDA) method [29, 30]. In order to drive the method to a meaningful solution, we use an initialization technique to determine the weights related to suitable starting values for **x** and **h**. Finally, we evaluate the effectiveness of the proposed approach by describing the numerical results of the NBD of synthetic images and of real images, showing the effect of the additional upper bound introduced on the blur kernel.

In summary, the contributions we propose in this work are listed below.

- We address the issue of images degraded by Poisson noise in an NBD problem; this is obtained by introducing the generalized KL divergence as discrepancy function instead of the standard least-squares data fidelity.
- We propose a further NBD formulation by introducing in the standard one an upper bound for the blur kernel; this constraint allows to exclude trivial solutions and even to identify coherent solutions with the optical instrument.

- We exploit two generative networks, aimed to recover approximations of the blur kernel and the image; the first is based on the SIREN architecture, instead of a simple FC network; the second is inspired by the one proposed in [21] and has an additional convolutional layer in the decoder block.
- We use pixel-wise weighted regularization terms for the blur kernel and the image in NBD problem; they are adapted to local patterns. In particular for the image local versions of Total Variation or ℓ_1 function are employed; the presence of regularization terms counterbalances the random input of the nets and the non-convexity of the problem, by better delimiting the set of possible solutions.
- We expand the stopping criterion in [27, 31] by adding a further control on the normalized KL [32].

The paper is organized as follows. Section 2 is devoted to present the problem, the network's structure and the choice for the regularization functionals. Section 3 presents the theoretical framework of the PGDA method employed for the minimization of (3) along its implementation details in algorithm. The numerical tests are presented in section 4, on both synthetic and real microscopic images. Section 5 gathers the results achieved in this work and presents some possible, future research strategies.

1.1. Notation

The symbol $\| \cdot \|$ denotes the standard Euclidean norm, $\| \cdot \|_\mathrm{F}$ denotes the Frobenius norm. Bold letters refer to vectors or tensors (it will be made clear by the context), whilst Greek and Latin letters refer to scalar. The Euclidean scalar product can be denoted by $\langle \cdot, \cdot \rangle$. We recall that the proximal operator of a convex function $F : \mathbb{R}^d \to \mathbb{R}$ at a vector $\mathbf{x} \in \mathbb{R}^d$ is defined as

$$\mathrm{prox}_F(\mathbf{x}) = \underset{\mathbf{w} \in \mathbb{R}^d}{\mathrm{argmin}}\, F(\mathbf{w}) + \frac{1}{2}\|\mathbf{w} - \mathbf{x}\|^2. \tag{5}$$

The proximal operator is well-defined for any convex function [33]. The proximal operator of the indicator function ι_C of a closed and convex set C at a vector $\mathbf{x}$ is the standard projection of $\mathbf{x}$ on C, i.e. $\mathrm{prox}_{\iota_C}(\mathbf{x}) = \Pi_C(\mathbf{x})$. Furthermore, for any vector $\mathbf{w} \in \mathbb{R}^2$, we define the shrinkage operator of $\mathbf{w}$ with parameter $\beta > 0$ as $\mathrm{shrink}_\beta(\mathbf{w}) = \underset{\mathbf{x} \in \mathbb{R}^2}{\mathrm{argmin}}\, \beta\|\mathbf{x}\| + \frac{1}{2}\|\mathbf{x} - \mathbf{w}\|^2 = \frac{\mathbf{w}}{\|\mathbf{w}\|} \max(0, \|\mathbf{w}\| - \beta)$. The symbol $\mathbf{1}$ denotes an array with all entries equal to 1.

2. The NBD problem

In a Bayesian approach, the BD problem can be formulated as the optimization problem (3), where, in presence of Poisson data, the data-fidelity function is expressed as (4). Motivated by the DIP idea [22, 23], following [21], in the BD problem the unknown blur kernel $\mathbf{h}$ and the image $\mathbf{x}$ can be replaced by the outputs of two generative networks $\mathcal{N}_h(\boldsymbol{\theta}_h)$ and $\mathcal{N}_x(\boldsymbol{\theta}_x)$, whose input data $\mathbf{z}_h$ and $\mathbf{z}_x$ are random samples. Combining this approach with the MAP minimization (3), the NBD problem can be formulated as

$$\min_{\boldsymbol{\theta}_h, \boldsymbol{\theta}_x}\ f_0^B(\mathcal{N}_h(\boldsymbol{\theta}_h), \mathcal{N}_x(\boldsymbol{\theta}_x); \mathbf{g}) + f_1(\mathcal{N}_h(\boldsymbol{\theta}_h)) + f_2(\mathcal{N}_x(\boldsymbol{\theta}_x)),$$

$$\mathrm{s.\,t.} \quad 0 \leqslant \mathcal{N}_x(\boldsymbol{\theta}_x) \leqslant 1, \quad \mathcal{N}_h(\boldsymbol{\theta}_h) \geqslant 0, \quad \sum_i (\mathcal{N}_h(\boldsymbol{\theta}_h))_i = 1 \tag{6}$$

where f_1 and f_2 are regularization terms for the outputs of the two nets; the constraints in (6) can be automatically meet, by setting as output layers of the two nets the sigmoid nonlinearity for $\mathcal{N}_x$ and the SoftMax nonlinearity for $\mathcal{N}_h$ respectively. By denoting the solution of the minimization problem as $(\boldsymbol{\theta}_h^*, \boldsymbol{\theta}_x^*)$, the restored image $\mathbf{x}^*$ can be viewed as the output of the generative network $\mathcal{N}_x$ where the network weights $\boldsymbol{\theta}_x^*$ are a parametrization of $\mathbf{x}^*$, i.e. $\mathbf{x}^* = \mathcal{N}_x(\boldsymbol{\theta}_x^*)$. Similarly, the blur kernel $\mathbf{h}^* = \mathcal{N}_h(\boldsymbol{\theta}_h^*)$ and the network weights $\boldsymbol{\theta}_h^*$ are a parametrization of $\mathbf{h}^*$. The formulation (6) can include the trivial solution provided by the δ-array as PSF. Indeed, when an iterative method is used to find an reliable approximate solution, numerical experiments show how crucial it is to determine at which iteration one should stop, given that, in general, the iterative procedure can lead to the trivial solution. Hence, the numerical experience highlights that the sequence of iterates $\mathcal{N}_h(\theta_h^{(k)})$ tends to collapse to a single point with value 1 and, consequently, the quality of the recovered images deteriorates. This fact can be observed in the numerical experiments of section 4, in particular in figures 4–6. Here, for any test problem of the considered dataset, we observe that the peak values of the PSF sequence exceed the maximum value of $\mathbf{h}^*$, continuing to increase and collapsing the iterates into one point (see the behaviour of the PSF peak value with respect the iterations in figures 4(a), 5(a) and 6(a)). To avoid this drawback, we follow the suggestion in [9], by requiring that the outputs of the network $\mathcal{N}_h$ are bounded from above by a prefixed peak value as well as being constrained by the non-negativity and normalization to 1. Thus, the problem (6) can be restated as follows:

$$\min_{\boldsymbol{\theta}_h, \boldsymbol{\theta}_x} \; f_0^B(\Pi_{C_{\mathbf{V}}}(\mathcal{N}_h(\boldsymbol{\theta}_h)), \mathcal{N}_x(\boldsymbol{\theta}_x); \mathbf{g}) + f_1\left(\Pi_{C_{\mathbf{V}}}(\mathcal{N}_h(\boldsymbol{\theta}_h))\right) + f_2\left(\mathcal{N}_x(\boldsymbol{\theta}_x)\right),$$
$$\text{s. t.} \quad 0 \leqslant \mathcal{N}_x(\boldsymbol{\theta}_x) \leqslant 1, \tag{7}$$

with $C_{\mathbf{V}} = \{0 \leqslant \mathbf{V} \leqslant H\mathbf{1}, \quad \sum_i (\mathbf{V})_i = 1\}$, $0 < H < 1$. We point out that, while the output of the net $\mathcal{N}_x$ automatically satisfies the box constraints, the output of $\mathcal{N}_h$ has to be projected in the convex set, $C_{\mathbf{V}} = \{0 \leqslant \mathbf{V} \leqslant H\mathbf{1}, \quad \sum_i (\mathbf{V})_i = 1\}$; this condition requires to be able to compute the projection operator on the set $C_{\mathbf{V}}$ and this can be implemented by exploiting efficient algorithms, such as [34–36]. Unlike what holds for (6), in this case $\mathbf{h}^* = \Pi_{C_{\mathbf{V}}}(\mathcal{N}_h(\boldsymbol{\theta}_h^*))$. The peak value H is strictly dependent on the optical instrument; its physical features enable to define an ideal PSF. The ideal PSF is different from the real PSF and it is possible to estimate the SR, finding a reliable value for H (see for example [37, 38]).

2.1. The two generative networks

The networks employed in the experimentation are inspired by the architectures used in [21, 26]. The network $\mathcal{N}_x$ is a classical U-net, with a depth of 5, i.e. the encoder and the decoder have five different levels of compression and decompression, respectively. Each block of the decoder comprehends a convolutional layer, with stride 1, followed by convolutional layer with stride 2 which halves the spatial dimensions, a batch normalization (BN) Layer, a LeakyReLU (with parameter 0.2), then again a convolutional layer with stride 1, a BN layer and a LeakyReLU layer. The decoder's blocks have a similar structure: a depth concatenation layer for the connection between the encoder and the decoder parts, followed by a BN layer, then a sequence composed by a convolutional layer with stride 1, a BN layer and a LeakyReLU is repeated twice. Finally, a resize layer is employed for doubling the spatial dimensions. The seminal paper on DIP [23] points out that the choice of the upsampling method (bilinear, nearest neighbour or transposed convolution) does not impact on the final performance. The resize layer is then followed by a BN and by a LeakyReLU. The input of this network is a
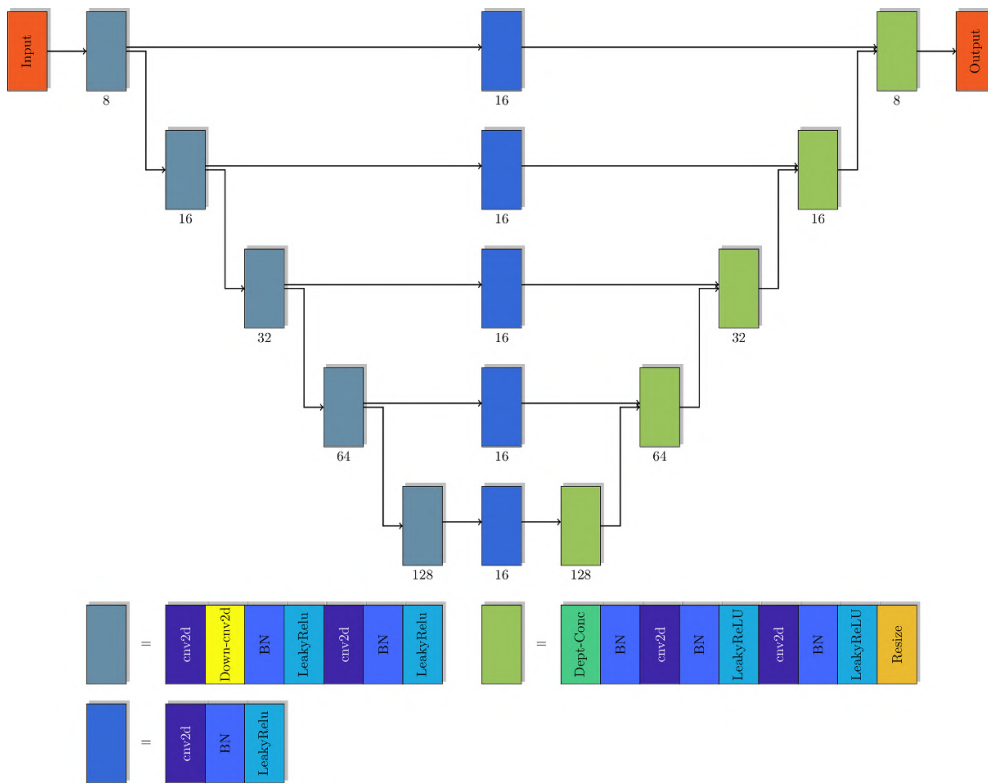
Figure 1. The visual scheme for the U-Net used in the numerical experimentation. The bottom row explain the structure of the encoder, decoder and skip connection blocks, respectively. The numbers at the bottom of each block refers to the number of filters used in the convolutional layers.

random 3D tensor of dimension $n \times n \times 32$ (see section 4 for the details of the probability distribution of such tensor). All the convolutional layers use convolutional filters of dimension 3×3. See figure 1 for a visual inspection of the $\mathcal{N}_x$ structure.

In the original papers [21, 26] the network for recovering of the PSF is a shallow, fully connected (FC) one: the motivation is based on the fact that the PSF does not have particular structures that require convolutional filters to be captured; hence a simple FC network is sufficient. The very recent work [27] implements the different architecture SIREN [39], which uses the sin function as neuronal activation. We follow such strategy since it has been observed in the numerical experiments and in real world scenarios that SIREN networks can learn high-frequency components in a better way than DIP [39]. The structure of the SIREN network employed in this work is depicted in figure 2: an input layer of 200 random components is followed by four instances of an FC layer followed by a sin activation layer. The last two layers are an FC one (in order to retrieve the desired dimension for the PSF) and a SoftMax layer. The initialization of such network is done following the suggestion depicted in [39].

2.2. Regularization terms

Although the generative nets have high impedance to image noise [40], numerical experience shows that multiple runs on the same input for the minimization of the objective data-fidelity
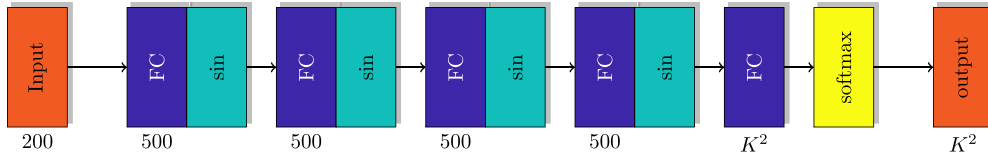
7

Figure 2. SIREN network for the reconstruction of the PSF. K^2 is the number of elements of the PSF.

function without regularization terms produce very different results [26]. This can be motivated by the random initialization of the nets, but also by the non-convexity of both the original problem and the reformulated one, that can admit many solutions. In order to address these drawbacks and make the approach more consistent, in (3) and, consequently in (6) and (7), two regularization terms are introduced, as in [26, 27]. For both the terms, we adopt a pixel-wise weighted regularization (see [41, 42]). To describe the choices made in this paper, we assume for simplicity to deal with a 2D $k \times k$ blur kernel, reshaped as vector of $\mathbb{R}^K$, $K = k \times k$, and $n \times n$ image, reshaped as vector of $\mathbb{R}^N$, $N = n \times n$. For the blur kernel we consider an adaptive Tikhonov-like regularization of order 0, which is defined as the sum of a set $\overline{\lambda}$ of positive parameters, locally adapted to the kernel pattern, combined with the terms of the standard regularization, that is

$$f_1(\mathbf{h}) = \frac{1}{2} \sum_{i=1}^{K} \overline{\lambda}_i \mathbf{h}_i^2. \tag{8}$$

Similarly, to devise a restored sharp image, we use a pixel-wise weighted regularization term, locally adapted to the image pattern, combined with the terms of a standard regularization function, well-suited to microscopy images, such as the ℓ_1 function [43] or the total variation (TV) [28]. Specifically f_2 can be written as

$$f_2(\mathbf{x}) = \sum_{i=1}^{N} \tilde{\lambda}_i R_i(\mathbf{x}), \tag{9}$$

where $R_i(\mathbf{x})$ is defined as follows

$$R_i(\mathbf{x}) = |\mathbf{x}_i| \qquad \text{for } \ell_1 \text{ regularization,} \tag{10}$$

$$R_i(\mathbf{x}) = \|\mathbf{A}_i \mathbf{x}\| \qquad \text{for TV regularization.} \tag{11}$$

Here $\mathbf{A}_i$ is the 2D discrete first order difference operator at the pixel i of the vector-reshaped image $\mathbf{x} \in \mathbb{R}^N$ and the matrices $\mathbf{A}_i \in \mathbb{R}^{2 \times N}$, $i = 1, \ldots, N$ are sub-matrices of a matrix $\mathbf{A} \in \mathbb{R}^{2N \times N}$, $\mathbf{A} = \left(\mathbf{A}_1^\top, \ldots, \mathbf{A}_N^\top \right)^\top$.

Both the selected regularization terms are convex, but the f_2 term is non-smooth. Thus, it should be inappropriate to use a gradient-like method, to address the problems (6) and (7), due to the non-existence of the f_2 gradient in neighborhoods with constant values of the image iterate. Nevertheless, the proximal operator of a suitable reformulation of f_2 can be computed by a closed formula. Consequently, we propose to address problem (7) by using a suitable scheme in the framework of alternating PGDA methods.

3. PGDA-like approach

Both the problems (6) and (7) can be restated as min-max problems, by introducing auxiliary variables. In particular, we discuss the case of problem (7), pointing out that for (6) the reformulation is based on the introduction of only one auxiliary variable, the one which deals with the regularization term f_2. In the case of (7), we have to consider two auxiliary variables, the first to realize the final projection of the output of $\mathcal{N}_h$ on $C_\mathbf{V}$ and the second to deal with the regularization term f_2. By considering as f_2 the case of the TV variant (9) and (11), the auxiliary variables are defined as follows:

$$\mathbf{V} = \mathcal{N}_h(\boldsymbol{\theta}_h), \qquad \mathbf{Y} = \mathbf{A}\mathcal{N}_x(\boldsymbol{\theta}_x). \tag{12}$$

Taking into account that the constraints on $\mathcal{N}_x(\theta_x)$ in (7) are automatically meet by the output of the U-net, the problem can be restated in the following form:

$$\min_{\boldsymbol{\theta}_h, \mathbf{V}, \boldsymbol{\theta}_x, \mathbf{Y}} f_0^B(\mathbf{V}, \mathcal{N}_x(\boldsymbol{\theta}_x); \mathbf{g}) + \frac{1}{2}\sum_{i=0}^{K}\overline{\lambda}_i \mathbf{V}_i^2 + \iota_{C_\mathbf{V}}(\mathbf{V}) + \sum_i \tilde{\lambda}_i\|\mathbf{Y}_i\|$$

$$\text{s. t.} \quad \mathbf{V} = \mathcal{N}_h(\boldsymbol{\theta}_h), \quad \mathbf{Y} = \mathbf{A}\mathcal{N}_x(\boldsymbol{\theta}_x) \tag{13}$$

where $\iota_{C_\mathbf{V}}(\mathbf{V})$ is the indicator function of the convex set $C_\mathbf{V} = \{0 \leqslant \mathbf{V} \leqslant H1, \quad \sum_i(\mathbf{V})_i = 1\}$. By introducing penalty terms for the equality constraints, the corresponding augmented Lagrangian function can be written as

$$\mathcal{L}_{\gamma_\mathbf{V}, \gamma_\mathbf{Y}}(\boldsymbol{\theta}_h, \mathbf{V}, \boldsymbol{\theta}_x, \mathbf{Y}, \boldsymbol{\mu}_\mathbf{V}, \boldsymbol{\mu}_\mathbf{V}) = f_0^B(\mathbf{V}, \mathcal{N}_x(\boldsymbol{\theta}_x); \mathbf{g}) + \frac{1}{2}\sum_{i=0}^{K}\overline{\lambda}_i\mathbf{V}_i^2 + \iota_{C_\mathbf{V}}(\mathbf{V}) + \sum_i\tilde{\lambda}_i\|\mathbf{Y}_i\|$$

$$+ \frac{\gamma_\mathbf{V}}{2}\|\mathbf{V} - \mathcal{N}_h(\boldsymbol{\theta}_h)\|^2 + \langle\boldsymbol{\mu}_\mathbf{V}, \mathbf{V} - \mathcal{N}_h(\boldsymbol{\theta}_h)\rangle$$

$$+ \frac{\gamma_\mathbf{Y}}{2}\|\mathbf{Y} - \mathbf{A}\mathcal{N}_x(\boldsymbol{\theta}_x)\|^2 + \langle\boldsymbol{\mu}_\mathbf{V}, \mathbf{Y} - \mathbf{A}\mathcal{N}_x(\boldsymbol{\theta}_x)\rangle, \tag{14}$$

where $\gamma_\mathbf{V}$ and $\gamma_\mathbf{Y}$ are positive penalty parameters and $\boldsymbol{\mu}_\mathbf{V}, \boldsymbol{\mu}_\mathbf{V}$ are the multipliers of the equality constraints. By setting $\mathbf{P} = (\boldsymbol{\theta}_h, \mathbf{V}, \boldsymbol{\theta}_x, \mathbf{Y})$ and $\boldsymbol{\mu} = (\boldsymbol{\mu}_\mathbf{V}, \boldsymbol{\mu}_\mathbf{V})$, we can bring the numerical solution of problem (7) to that of the following saddle point problem

$$\min_\mathbf{P} \max_{\boldsymbol{\mu}} \mathcal{L}_{\gamma_\mathbf{V}, \gamma_\mathbf{Y}}(\mathbf{P}, \boldsymbol{\mu}). \tag{15}$$

We observe that the augmented Lagrangian function can be subdivided into two terms:

$$\mathcal{L}_{\gamma_\mathbf{V}, \gamma_\mathbf{Y}}(\mathbf{P}, \boldsymbol{\mu}) = K(\mathbf{P}, \boldsymbol{\mu}) + R(\mathbf{P}), \tag{16}$$

where we have

$$K(\mathbf{P}, \boldsymbol{\mu}) = f_0^B(\mathbf{V}, \mathcal{N}_x(\boldsymbol{\theta}_x); \mathbf{g}) + \frac{1}{2}\sum_{i=0}^{K}\overline{\lambda}_i\mathbf{V}_i^2$$

$$+ \frac{\gamma_\mathbf{V}}{2}\|\mathbf{V} - \mathcal{N}_h(\boldsymbol{\theta}_h)\|^2 + \langle\boldsymbol{\mu}_\mathbf{V}, \mathbf{V} - \mathcal{N}_h(\boldsymbol{\theta}_h)\rangle$$

$$+ \frac{\gamma_\mathbf{Y}}{2}\|\mathbf{Y} - \mathbf{A}\mathcal{N}_x(\boldsymbol{\theta}_x)\|^2 + \langle\boldsymbol{\mu}_\mathbf{V}, \mathbf{Y} - \mathbf{A}\mathcal{N}_x(\boldsymbol{\theta}_x)\rangle$$

$$R(\mathbf{P}) = \sum_i\tilde{\lambda}_i\|\mathbf{Y}_i\| + \iota_{C_\mathbf{V}}(\mathbf{V}). \tag{17}$$

The function $K(\mathbf{P}, \boldsymbol{\mu}_\mathbf{V})$ is smooth; the non-smooth $R(\mathbf{P})$ term is actually a separable function of $\mathbf{V}$ and $\mathbf{Y}$; there exists a closed formula to compute the proximal operator of each term $\|\mathbf{Y}_i\|$,

$i = 1, \ldots, N$, whereas the proximity operator of $\iota_{C_V}(\mathbf{V})$ is the projection on the convex set C_V, numerically obtainable by means of efficient algorithms, as for example the ones in [34–36]. Thus the saddle point problem (15) can be formulated as

$$\min_{\mathbf{P}} \max_{\boldsymbol{\mu}} \; K(\mathbf{P}, \boldsymbol{\mu}) + R(\mathbf{P}),$$

which is the typical form of the problems addressed by alternating PGDA methods, recently developed in [29, 30] and already adopted for DIP models in [28]. Upon suitable initialization of the variables, the kth iteration of PGDA iterative algorithm reads as

$$\mathbf{P}^{(k+1)} = \text{prox}_{\alpha_{\mathbf{P}} R}(\mathbf{P}^{(k)} - \alpha_{\mathbf{P}} \nabla_{\mathbf{P}} K(\mathbf{P}^{(k)}, \boldsymbol{\mu}^{(k)}))$$
$$\boldsymbol{\mu}^{(k+1)} = \boldsymbol{\mu}^{(k)} + \alpha_{\boldsymbol{\mu}} \nabla_{\boldsymbol{\mu}} K(\mathbf{P}^{(k+1)}, \boldsymbol{\mu}^{(k+1)}), \tag{18}$$

where $\alpha_{\mathbf{P}}$ and $\alpha_{\boldsymbol{\mu}}$ are positive learning rates. Due to the separability of the functions $K(\mathbf{P}, \boldsymbol{\mu})$ and $R(\mathbf{P})$, the general scheme (18) can be detailed as follows:

$$\boldsymbol{\theta}_h^{(k+1)} = \boldsymbol{\theta}_h^{(k)} - \alpha_{\mathbf{P}} \nabla_{\boldsymbol{\theta}_h} \left[\frac{\gamma_{\mathbf{V}}}{2} \|\mathbf{V}^{(k)} - \mathcal{N}_h(\boldsymbol{\theta}_h^{(k)})\|^2 + \left\langle \boldsymbol{\mu}_{\mathbf{V}}^{(k)}, \mathbf{V}^{(k)} - \mathcal{N}_h(\boldsymbol{\theta}_h^{(k)}) \right\rangle \right]$$

$$\mathbf{V}^{(k+1)} = \text{prox}_{\alpha_{\mathbf{P}} \iota_{C_V}} \left(\mathbf{V}^{(k)} - \alpha_{\mathbf{P}} \nabla_{\mathbf{V}} \left[f_0^B(\mathbf{V}^{(k)}, \mathcal{N}_x(\boldsymbol{\theta}_x^{(k)}); \mathbf{g}) + \frac{1}{2} \sum_{i=0}^{K} \overline{\lambda}_i \mathbf{V}_i^2 \right. \right.$$

$$\left. \left. + \frac{\gamma_{\mathbf{V}}}{2} \|\mathbf{V}^{(k)} - \mathcal{N}_h(\boldsymbol{\theta}_h^{(k)})\|^2 + \left\langle \boldsymbol{\mu}_{\mathbf{V}}^{(k)}, \mathbf{V}^{(k)} - \mathcal{N}_h(\boldsymbol{\theta}_h^{(k)}) \right\rangle \right] \right)$$

$$\boldsymbol{\theta}_x^{(k+1)} = \boldsymbol{\theta}_x^{(k)} - \alpha_{\mathbf{P}} \nabla_{\boldsymbol{\theta}_x} \left[f_0^B(\mathbf{V}^{(k)}, \mathcal{N}_x(\boldsymbol{\theta}_x^{(k)}); \mathbf{g}) \right.$$

$$\left. + \frac{\gamma_{\mathbf{Y}}}{2} \|\mathbf{Y}^{(k)} - \mathbf{A}\mathcal{N}_x(\boldsymbol{\theta}_x^{(k)})\|^2 + \left\langle \boldsymbol{\mu}_{\mathbf{V}}^{(k)}, \mathbf{Y}^{(k)} - \mathbf{A}\mathcal{N}_x(\boldsymbol{\theta}_x^{(k)}) \right\rangle \right]$$

$$\mathbf{Y}^{(k+1)} = \text{prox}_{\alpha_{\mathbf{P}} f_2} \left(\mathbf{Y}^{(k)} - \alpha_{\mathbf{P}} \nabla_{\mathbf{Y}} \left[\frac{\gamma_{\mathbf{Y}}}{2} \|\mathbf{Y}^{(k)} - \mathbf{A}\mathcal{N}_x(\boldsymbol{\theta}_x^{(k)})\|^2 \right. \right.$$

$$\left. \left. + \left\langle \boldsymbol{\mu}_{\mathbf{V}}^{(k)}, \mathbf{Y}^{(k)} - \mathbf{A}\mathcal{N}_x(\boldsymbol{\theta}_x^{(k)}) \right\rangle \right] \right)$$

$$\boldsymbol{\mu}_{\mathbf{V}}^{(k+1)} = \boldsymbol{\mu}_{\mathbf{V}}^{(k)} + \alpha_{\boldsymbol{\mu}} \left(\mathbf{V}^{(k+1)} - \mathcal{N}_h(\boldsymbol{\theta}_h^{(k+1)}) \right)$$

$$\boldsymbol{\mu}_{\mathbf{V}}^{(k+1)} = \boldsymbol{\mu}_{\mathbf{V}}^{(k)} + \alpha_{\boldsymbol{\mu}} \left(\mathbf{Y}^{(k+1)} - \mathbf{A}\mathcal{N}_x(\boldsymbol{\theta}_x^{(k+1)}) \right).$$

Then, the explicit formulation of the above iteration is depicted as follows:

$$\boldsymbol{\theta}_h^{(k+1)} = \boldsymbol{\theta}_h^{(k)} - \alpha_{\mathbf{P}} \nabla_{\boldsymbol{\theta}_h} \mathcal{N}_h(\boldsymbol{\theta}_h^{(k)})(\gamma_{\mathbf{V}}(\mathcal{N}_h(\boldsymbol{\theta}_h^{(k)}) - \mathbf{V}^{(k)}) - \boldsymbol{\mu}_{\mathbf{V}}^{(k)}) \tag{19a}$$

$$\mathbf{V}^{(k+1)} = \Pi_{C_V} \left(\mathbf{V}^{(k)} - \alpha_{\mathbf{P}} (\nabla_{\mathbf{V}} [f_0^B(\mathbf{V}^{(k)}, \mathcal{N}_x(\boldsymbol{\theta}_x^{(k)}); \mathbf{g})] \right.$$

$$\left. + \sum_{i=0}^{K} \overline{\lambda}_i \mathbf{V}_i + \gamma_{\mathbf{V}}(\mathbf{V}^{(k)} - \mathcal{N}_h(\boldsymbol{\theta}_h^{(k)})) + \boldsymbol{\mu}_{\mathbf{V}}^{(k)}) \right) \tag{19b}$$

$$\boldsymbol{\theta}_x^{(k+1)} = \boldsymbol{\theta}_x^{(k)} - \alpha_{\mathbf{P}} \nabla_{\boldsymbol{\theta}_x} \mathcal{N}_x(\boldsymbol{\theta}_x^{(k)}) \left(\nabla_{\mathcal{N}_x(\boldsymbol{\theta}_x)} \left[f_0^B(\mathbf{V}^{(k)}, \mathcal{N}_x(\boldsymbol{\theta}_x^{(k)}); \mathbf{g}) \right] + \right.$$

$$\left. - \gamma_{\mathbf{Y}} \mathbf{A}^\top \left(Y^{(k)} - \mathbf{A}\mathcal{N}_x(\boldsymbol{\theta}_x^{(k)}) + \frac{\boldsymbol{\mu}_{\mathbf{V}}}{\gamma_{\mathbf{Y}}} \right) \right) \tag{19c}$$

$$\mathbf{Y}_i^{(k+1)} = \underset{\alpha_{\mathbf{P}} \tilde{\lambda}_i}{\text{shrink}} \left(\mathbf{Y}_i^{(k)} - \alpha_{\mathbf{P}} \gamma_{\mathbf{Y}} \left(\mathbf{Y}_i^{(k)} - \mathbf{A}_i \mathcal{N}_x(\boldsymbol{\theta}_x^{(k)}) \right) - \alpha_{\mathbf{P}} (\boldsymbol{\mu}_{\mathbf{V}}^{(k)})_i \right) \qquad (19d)$$

$$\boldsymbol{\mu}_{\mathbf{V}}^{(k+1)} = \boldsymbol{\mu}_{\mathbf{V}}^{(k)} + \alpha_{\boldsymbol{\mu}} \left(\mathbf{V}^{(k+1)} - \mathcal{N}_h(\boldsymbol{\theta}_h^{(k+1)}) \right) \qquad (19e)$$

$$\boldsymbol{\mu}_{\mathbf{V}}^{(k+1)} = \boldsymbol{\mu}_{\mathbf{V}}^{(k)} + \alpha_{\boldsymbol{\mu}} \left(\mathbf{Y}^{(k+1)} - \mathbf{A} \mathcal{N}_x(\boldsymbol{\theta}_x^{(k+1)}) \right) \qquad (19f)$$

where the operator shrinkage is applied to any sub-vector $(\mathbf{Y})_i$ corresponding to $\mathbf{A}_i \mathcal{N}_x(\boldsymbol{\theta}_x)$, $i = 1, \ldots, N$. The steps (19a) and (19c) can be easily obtained by one step of a stochastic gradient method applied to functions $\frac{\gamma_{\mathbf{V}}}{2} \|\mathbf{V}^{(k)} - \mathcal{N}_h(\boldsymbol{\theta}_h)\|^2 + \left\langle \boldsymbol{\mu}_{\mathbf{V}}^{(k)}, \mathbf{V}^{(k)} - \mathcal{N}_h(\boldsymbol{\theta}_h) \right\rangle$ and $f_0^B(\mathbf{V}^{(k)}, \mathcal{N}_x(\boldsymbol{\theta}_x); \mathbf{g}) + \frac{\gamma_{\mathbf{Y}}}{2} \|\mathbf{Y}^{(k)} - \mathbf{A} \mathcal{N}_x(\boldsymbol{\theta}_x)\|^2 + \left\langle \boldsymbol{\mu}_{\mathbf{V}}^{(k)}, \mathbf{Y}^{(k)} - \mathbf{A} \mathcal{N}_x(\boldsymbol{\theta}_x) \right\rangle$ with learning rate $\alpha_{\mathbf{P}}$, respectively; the updating rule (19b) for $\mathbf{V}$ can be obtained by applying the algorithm proposed in [35]; in view of the physical assumption $KH > 1$, the projection sub-problem is well-defined; the computation of the proximal operator related to the regularization term (9) (with parameter $\alpha_{\mathbf{P}}$) can be easily obtained in a closed form; the updating rule is well-defined, since f_2 is a convex function. Finally, the last updating rules for the multiplier vectors are ascent steps with learning rate $\alpha_{\boldsymbol{\mu}}$.

Remark 1. For the selection of parameters $\overline{\lambda}_i$ and $\tilde{\lambda}_i$ we follow [28, 41], varying these parameters along the iterations, according to the Uniform PENalty principle [43]. In particular, we set

$$\overline{\lambda}_i^{(k)} = \frac{1}{2K} \frac{f_0^B(\mathbf{V}^{(k)}, \mathcal{N}_x(\boldsymbol{\theta}_x^{(k)}); \mathbf{g})}{\frac{1}{2}(\mathbf{V}_i^{(k)})^2}, \quad \tilde{\lambda}_i^{(k)} = \frac{1}{2N} \frac{f_0^B(\mathbf{V}^{(k)}, \mathcal{N}_x(\boldsymbol{\theta}_x^{(k)}); \mathbf{g})}{\|\mathbf{A}_i \mathcal{N}_x(\boldsymbol{\theta}_x^{(k)})\|}. \qquad (20)$$

As a consequence, the pixels where the value of the local components $(\mathbf{V}_i^{(k)})^2$ and $\|\mathbf{A}_i \mathcal{N}_x(\boldsymbol{\theta}_x^{(k+1)})\|$ are smaller, have a greater regularization. In the experimentation, we modify the denominator as $\frac{1}{2}(\mathbf{V}_i^{(k)})^2 + \rho^2$ and $\|\mathbf{A}_i \mathcal{N}_x(\boldsymbol{\theta}_x^{(k+1)}) + \rho^2\|$ respectively, with $\rho \ll 1$, in order to avoid the annihilation of the denominators.

Remark 2. When the regularization term (9) and (10) is adopted for the image, the auxiliary variable $\mathbf{Y}$ is defined as the N dimensional vector $\mathbf{Y} = \mathcal{N}_x(\boldsymbol{\theta}_x)$. Consequently, the formulation of the problem (13) and the definition of augmented Lagrangian function (14) are accordingly simplified. The discussion on the implementation of PGDA remains unchanged, by adopting small modifications in (19c), (19d) and (19f) (replacement of $\mathbf{A}$ with the identity of order N and consistent computation of the proximal operator). Furthermore, the definition of regularization parameters is

$$\tilde{\lambda}_i^{(k)} = \frac{1}{2N} \frac{f_0^B(\mathcal{N}_h(\boldsymbol{\theta}_h^{(k)}), \mathcal{N}_x(\boldsymbol{\theta}_x^{(k)}); \mathbf{g})}{|\mathcal{N}_x(\boldsymbol{\theta}_x^{(k+1)})_i| + \rho^2}, \qquad (21)$$

with $\rho \ll 1$, to prevent the annihilation of the denominator.

Remark 3. The theoretical properties of the alternating PGDA method are discussed in [29]. With reference to (16), under the assumptions that $K(\mathbf{P}, \boldsymbol{\mu})$ is ρ-weakly convex ($\rho > 0$) and Lipschitz continuous in $\mathbf{P}$ uniformly in the second component, concave with Lipschitz continuous gradient in $\boldsymbol{\mu}_{\mathbf{V}}$ uniformly in the first component, and $R(\mathbf{P})$ is proper, convex, lower semicontinous and Lipschitz continuous on its domain, it is stated that an ϵ-stationary point

[29] can be visited in a finite number of iterations depending on ϵ. The learning rates $\alpha_{\mathbf{P}}$ and $\alpha_{\boldsymbol{\mu}_{\mathbf{V}}}$ have to satisfy upper bounds involving the unknown Lipschitz parameters. In practice, sufficiently small values of the two learning rates are fixed *a priori*.

3.1. Starting vectors

In order to obtain meaningful results by the iterative scheme (19*a*)–(19*f*), a crucial point is to initialize the weights of the nets $\mathcal{N}_h$ and $\mathcal{N}_x$ so that the deblurring loop has a suitable starting point (see [26] for more insights and deeper explanation). Thus, $\mathbf{x}^{(0)} = \mathbf{g}$, whereas $\mathbf{h}^{(0)}$ is chosen accordingly to the application at hand, for example a Gaussian kernel or via a preset model (see section 4 for more details). To initialize the parameters of the two nets, the following mean square error problems must be solved

$$\min_{\boldsymbol{\theta}_h} \|\mathcal{N}_h(\boldsymbol{\theta}_h) - \mathbf{h}^{(0)}\|^2, \tag{22}$$

$$\min_{\boldsymbol{\theta}_x} \|\mathcal{N}_x(\boldsymbol{\theta}_x) - \mathbf{x}^{(0)}\|^2. \tag{23}$$

Problems (22) and (23) are quickly solved via the Adam method: the procedure is stopped upon reaching the maximum number of iteration T_h, T_x for (22) and (23), respectively, or when the loss function reaches a value at most equal to a percentage (in our experiments, 0.1%) of its initial values.

3.2. Algorithm details

The whole procedure of the Double Deep Image Prior for Poisson data (DDIPP) is depicted in algorithm.

Instead of running the algorithm until it reaches the maximum number of iteration T, we implement the windowed-moving-variance strategy of [27, 31]. Such strategy consists in storing, at iteration k, the last $W \in \mathbb{N}$ recovered images in a queue $\mathcal{Q}$: $\{\mathcal{N}_x(\boldsymbol{\theta}_x^{(k-i)})\}_{i=0,\ldots,W-1}$ and compute their variance. The last $p \in \mathbb{N}$ minimum variances are memorized: if these p variances stagnate, then the procedure is stopped and the output consists in the last achieved recovered image and its related PSF. The parameter p takes the name of *patience*. The variance at iteration k has the following expression

$$\mathrm{WMV}(k) = \frac{1}{W}\sum_{i=0}^{W-1}\left\|\mathbf{x}^{(k-i)} - \frac{1}{W}\sum_{j=0}^{W-1}\mathbf{x}^{(k-j)}\right\|_{\mathrm{F}}^2, \tag{24}$$

where $\mathbf{x}^{(k)} = \mathcal{N}_x(\boldsymbol{\theta}_x^{(k)})$. The implementation of this procedure consists in checking that the last p minimum variances have an absolute difference less than a given tolerance ε_p.

A further suggestion about the stopping of the DDIPP can be provided from [1, lemma 4.1] (see also [32, 44, 45]), where it is affirmed that, since $\mathbf{g}_i$ can be viewed as a realization of a Poisson random variable with expected value $(\mathbf{h} * \mathbf{x} + b)_i$, if $\mathbf{g}_i$ is sufficiently large, the expected value of $2\left(\mathbf{g}_i \log\left(\frac{\mathbf{g}_i}{(\mathbf{h}*\mathbf{x}+b)_i}\right) + (\mathbf{h} * \mathbf{x} + b)_i - \mathbf{g}_i\right)$ is approximately 1. When the current values of $\mathbf{V}^{(k)}$ and $\mathbf{x}^{(k)}$ are near to the solutions, one can expect that the value of normalized discrepancy, i.e.

$$\mathcal{D}(k) = \frac{2}{N}f_0^B(\mathbf{V}^{(k)}, \mathbf{x}^{(k)}; \mathbf{g}), \tag{25}$$

decreases and tends to fluctuate around 1. As remarked in [1], the use of the normalized discrepancy is strictly related to the assumption that the data satisfy Poisson statistics and that the blur kernel provides is a good approximation of the blur process. However, it seems reasonable to assume that the region in which an approximate solution is found is the one for which $\mathcal{D}(k) < S$, with S of the order of the unity, for example $S = 2$. The stopping test based on the windowed-moving-variance strategy can be performed only when the normalized discrepancy is less than S. As shown in the following numerical experiments, this strategy enables us to avoid to prematurely stop the method: indeed, it may happen that in the early iterations the recovered images are actually not completely reliable but at the same time their differences are minimal. This leads the criterion (24) to be satisfied, but the reconstruction is still far away from a reliable solution.

Remark. Two technical observations about the actual implementation of the proposed procedure are reported below.

(i) Algorithm requires a gradient step for the update of the networks' parameters. The numerical experience showed that such simple step is not sufficient to achieve reliable results in a reasonable amount of time, hence we resort to use the Adam algorithm for updating the nets' parameters in algorithm.

(ii) The numerical experience showed that updating $\mathbf{Y}^{(k+1)}$ using $\mathcal{N}_x(\boldsymbol{\theta}_x^{(k+1)})$ instead of $\mathcal{N}_x(\boldsymbol{\theta}_x^{(k)})$ induces an appreciable speed-up.

4. Numerical experiments

This section is devoted to assess the performance of the proposed procedure on synthetic images and on real microscopy images. Our code has been implemented in MatLab® R2022b, using the Deep Learning and Image Processing toolboxes. The code is available online at https://github.com/AleBenfe/DDIPP. The settings for algorithm and its simplified version for model (6) are detailed in the following paragraphs; we remark that the steplengths $\alpha_{\mathbf{P}}$ and $\alpha_{\boldsymbol{\mu}}$ are set to small values in order to ensure the convergence of PGDA, without a special tuning to decrease the number of iterations; furthermore, in all cases, we set $T_h = T_x = 200$ and the maximum number T of iteration is 1000. The random inputs of the networks, namely $\mathbf{z}_x$ for $\mathcal{N}_x$ and $\mathbf{z}_h$ for $\mathcal{N}_h$, are initially drawn from a uniform distribution in $[0, 1]$ and in $[-1, 1]$ respectively; at each iteration $\mathbf{z}_x$ is perturbed with Gaussian noise with zero mean and variance equal to 0.001 whilst $\mathbf{z}_h$ is left unperturbed, according to the strategy depicted in [21, 26]. The patience p is set to 20 and the queue length W is 10, $\varepsilon_p = 10^{-3}$ and $S = 2$. The regularization parameters for the blur kernel and the image are adaptively updated as in (20), (21). Following the strategy depicted in [27], the size of the recovered PSF is overestimated, depending on the image size, even if, in case of synthetic data, the true PSF has been generated with a lower number of pixels. For synthetic data, we employ the peak signal-to-noise ratio (PSNR) between the recovered data and the ground truths $\mathbf{x}^*, \mathbf{h}^*$ as a figure of merit, as well as the structural similarity index measure (SSIM) of the reconstructed image with respect to $\mathbf{x}^*$.

Algorithm 1. DDIPP: Double DIP for Poisson noise

Set the parameters $\alpha_{\mathbf{P}}, \alpha_{\boldsymbol{\mu}}, \gamma_{\mathbf{V}}, \gamma_{\mathbf{Y}}$; set $r_{\min} = +\infty$ and choose ε_p and S.
Choose the length W of the queue $\mathcal{Q}$ and the patience p; choose T_h, T_x, T
Initialise $\mathcal{N}_x$ solving (23) with a max number of iterations equal to T_x
Initialise $\mathcal{N}_h$ solving (22) with a max number of iterations equal to T_h
for $t = 1, 2, \ldots, T$ **do**
   Update $\boldsymbol{\theta}_h^{(k+1)}$ as in (19a)
   Update $\mathbf{V}^{(k+1)}$ as in (19b)
   Update $\boldsymbol{\theta}_x^{(k+1)}$ as in (19c)
   Store $\mathcal{N}_x(\boldsymbol{\theta}_x^{(k+1)})$ in the queue $\mathcal{Q}$
  if $|\mathcal{Q}| \geqslant W$ **then**
    Compute the WMV of the last W elements in $\mathcal{Q}$
    $\hat{\mathbf{x}} = \mathcal{N}_x(\boldsymbol{\theta}_x^{(k+1)})$
    $\hat{\mathbf{h}} = \mathbf{V}^{(k+1)}$
    if WMV$< \min\{r_{\min}\}$ **then**
       $r_{\min} \hookleftarrow$ WMV
    end if
    if $\sum \mathtt{diff}(r_{\min}) < \varepsilon_p(p-1)$ and $\mathcal{D}(k) < S$ **then**
       Output $\hat{\mathbf{x}}, \hat{\mathbf{h}}$
    end if
  end if
  Set $\bar{\lambda}_i$ as in (20)
  Update $\mathbf{Y}^{(k+1)}$ as in (19d), employing $\boldsymbol{\theta}_x^{(k+1)}$
  Update $\boldsymbol{\mu}_{\mathbf{V}}^{(k+1)}$ as in (19e)
  Update $\boldsymbol{\mu}_{\mathbf{V}}^{(k+1)}$ as in (19f)
end for

In the following, all the images are displayed in the range $[0, 1]$, whereas the renderings of the blur kernel show the scale on the right.

4.1. Synthetic dataset

The first experimentation is carried on the following dataset, where each blurred image is perturbed with Poisson noise via the MatLab® function `imnoise`:

- `rice`: this image belongs to the MatLab® Image Processing Toolbox. It is a 256×256 image with pixel values in $[0, 255]$; the blur operator consists in a Gaussian kernel with size 17×17 and standard deviation equal to 1.7; the peak value of the PSF is 0.055; furthermore, $\mathrm{PSNR}(\mathbf{g}) = 22.67$, $\mathrm{SSIM}(\mathbf{g}) = 0.51$.
- `micro`: the original image is a phantom of size 128×128 described in [46]; the PSF consists in a Gaussian blur with standard deviation equal to $\sqrt{5}$ and peak value 0.032; furthermore, $\mathrm{PSNR}(\mathbf{g}) = 24.51$, $\mathrm{SSIM}(\mathbf{g}) = 0.84$.
- `synth001`: this is a synthetic simulation of real-world microscopy images; the procedure employed for the generation of such image is explained in [47] and the code is available at https://github.com/AleBenfe/upU-net_Perlin. The PSF used for blurring these images is obtained via the software available at http://bigwww.epfl.ch/algorithms/psfgenerator/ (see [48–50] for more technical details). A 3D PSF (with size $64 \times 64 \times 11$) is generated
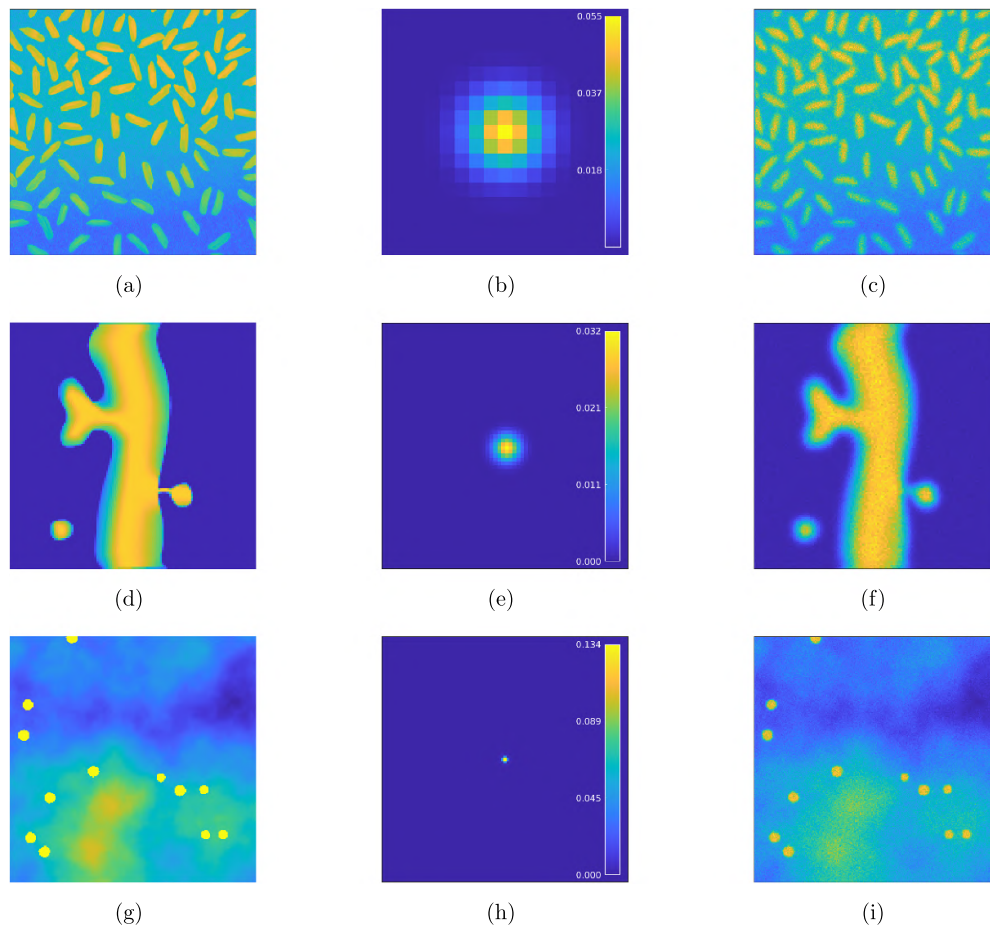
Figure 3. (a): ground truth image of `rice` test problem. (b) PSF of `rice` test problem (peak value 0.055). (c) noisy blurred image **g** for `rice` test problem. (d): ground truth image of `micro` test problem. (e) PSF of `micro` test problem (peak value 0.032). (f) noisy blurred image **g** for `micro` test problem. (g): ground truth image of `synth001` test problem. (h) PSF of `synth001` test problem (peak value 0.134). (i) noisy blurred image **g** for `synth001` test problem.

using the Born and Wolf 3D model: then the central frame is normalized to 1 and used as blur operator; the peak value of PSF is 0.134; furthermore, $PSNR(\mathbf{g}) = 21.96$, $SSIM(\mathbf{g}) = 0.52$.

Figure 3 provides a visual inspection of the dataset.

4.1.1. Comparison between the models (6) and (7). In the first experiment we compare the model (6) with the novel reformulation in (7), in order to check the effectiveness of the additional constraint on the output of $\mathcal{N}_h$. For both the models, the regularization terms (8) and (9),(11) are used for the PSF and the image respectively. In the following, we denote this choice with the suffix $\ell_2 - \mathrm{TV}$.
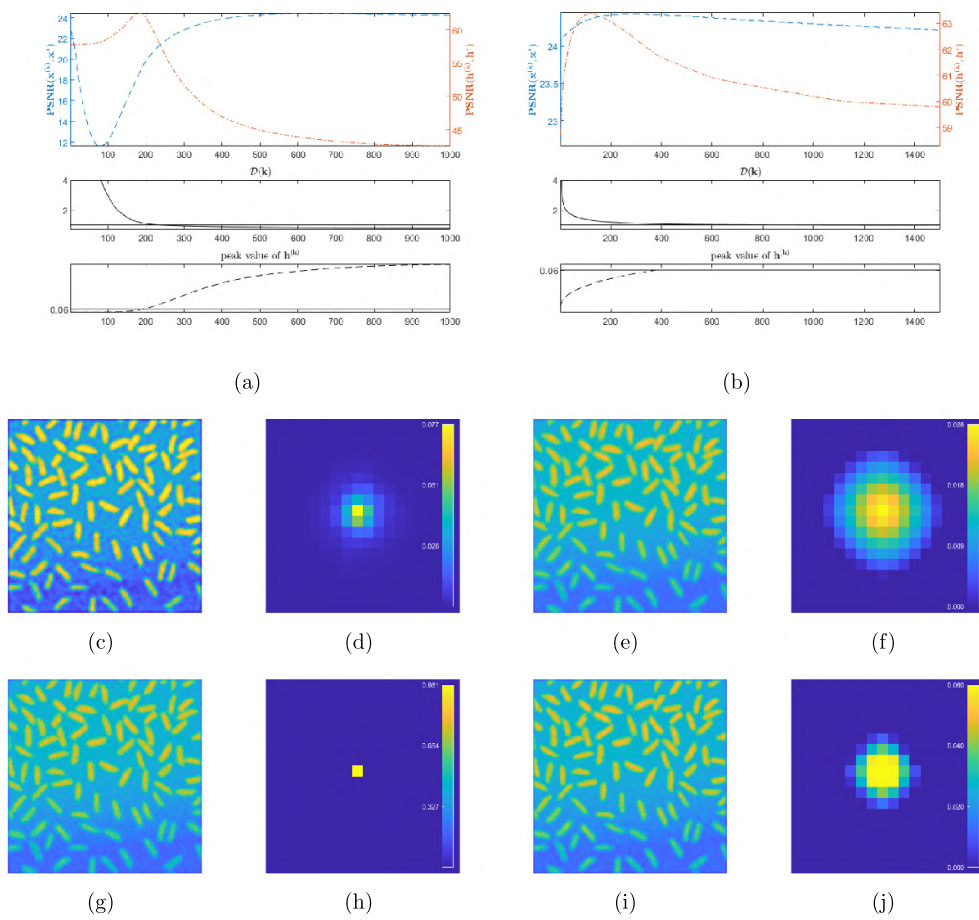
(a)   (b)

(c)   (d)   (e)   (f)

(g)   (h)   (i)   (j)

Figure 4. `rice` test problem—The panels on the left show the behaviour of PGDA method for the model (6) $\ell_2 - TV$ ($\alpha_{\mathbf{P}} = 5 \times 10^{-5}, \alpha_{\boldsymbol{\mu}} = 5 \times 10^{-4}, \gamma_{\mathbf{Y}} = 1$): (a) PSNR of $\mathbf{x}^{(k)}$ and $\mathbf{h}^{(k)}$, normalized discrepancy $\mathcal{D}(k)$ and peak value of $\mathbf{h}^{(k)}$, (c)–(d) reconstructions of the image and the PSF at the iteration 200, (g)–(h) reconstructions of the image and the PSF at the iteration $T = 1000$. The panels on the right show the behaviour of PGDA method for the model (7) $\ell_2 - TV$: ($\alpha_{\mathbf{P}} = 10^{-6}, \alpha_{\boldsymbol{\mu}} = 10^{-5}, \gamma_{\mathbf{Y}} = 1, \gamma_{\mathbf{V}} = 10^{-3}$): (b) PSNR of $\mathbf{x}^{(k)}$ and $\mathbf{h}^{(k)}$, normalized discrepancy $\mathcal{D}(k)$ and peak value of $\mathbf{h}^{(k)}$, (e)–(f) reconstructions of the image and the PSF at the iteration $K = 60$, (i)–(j) reconstructions of the image and the PSF at the iteration $T = 1000$.

In both cases, the numerical solution is addressed with the alternating PGDA method: in the latter we used the implementation of algorithm, while in the former a tailored version of this algorithm for the model (6). In figures 4–6 we show for both the models the behaviour of the PSNR of the PSF and of the image, i.e. PSNR($\mathbf{h}^{(k)}$) and PSNR($\mathbf{x}^{(k)}$), the images and the blur kernels recovered at the iteration K where the stopping criterion is satisfied (or at an intermediate iteration where it is not satisfied), and at the last iteration T.

In table 1 we report for the three test problems some figures of merit of the recovered images and PSFs for both models (6) and (7) $\ell_2 - TV$. For test problems `rice` and `synth001`,
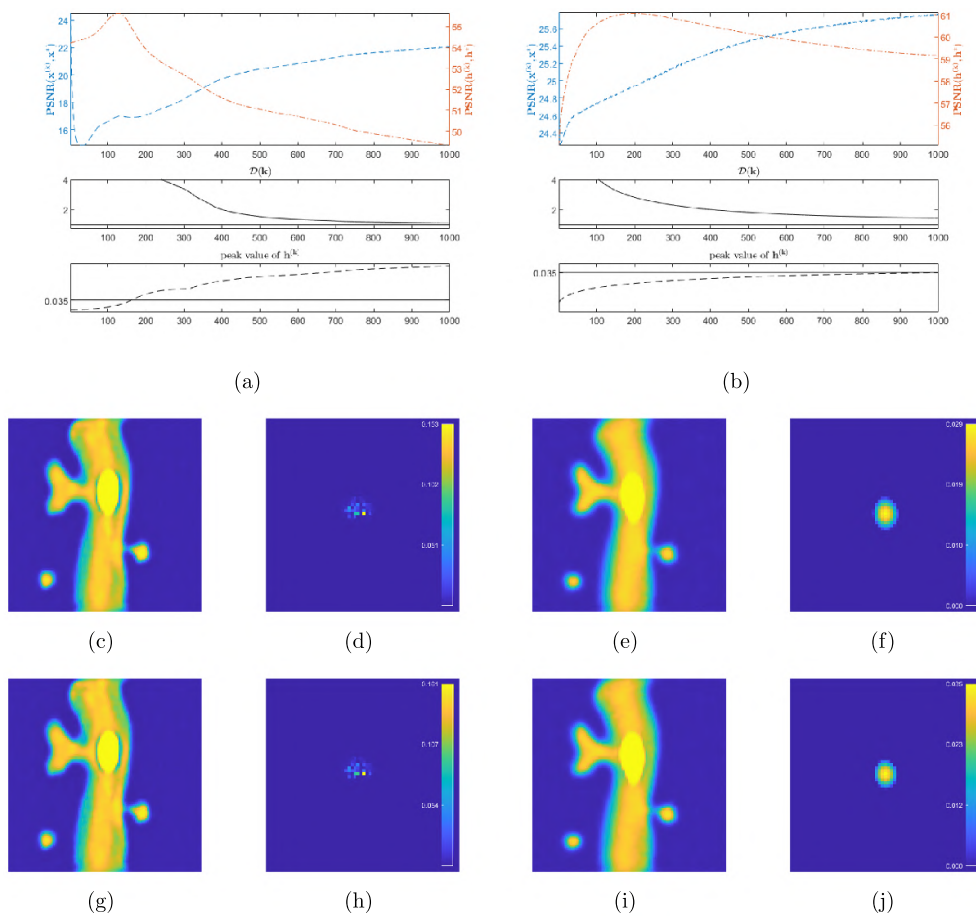
(a) (b)

(c) (d) (e) (f)

(g) (h) (i) (j)

Figure 5. `micro` test problem—The panels on the left show the behaviour of PGDA method for the model (6) $\ell_2 - \text{TV}$ ($\alpha_\mathbf{P} = 5 \times 10^{-5}, \alpha_\mu = 5 \times 10^{-4}, \gamma_\mathbf{Y} = 1$): (a) PSNR of $\mathbf{x}^{(k)}$ and $\mathbf{h}^{(k)}$, normalized discrepancy $\mathcal{D}(k)$ and peak value of $\mathbf{h}^{(k)}$, (c)–(d) reconstructions of the image and the PSF at the iteration $K = 860$, (g)–(h) reconstructions of the image and the PSF at the iteration $T = 1000$. The panels on the right show the behaviour of PGDA method for the model (7) $\ell_2 - \text{TV}$: (b) ($\alpha_\mathbf{P} = 10^{-6}, \alpha_\mu = 10^{-5}, \gamma_\mathbf{Y} = 1, \gamma_\mathbf{V} = 10^{-3}$): (b) PSNR of $\mathbf{x}^{(k)}$ and $\mathbf{h}^{(k)}$, normalized discrepancy $\mathcal{D}(k)$ and peak value of $\mathbf{h}^{(k)}$, (e)–(f) reconstructions of the image and the PSF at the iteration $K = 400$, (i)–(j) reconstructions of the image and the PSF at the iteration $T = 1000$.

the stopping criterion of PGDA is not satisfied in the case of model (6). Consequently, for `rice` and `synth001` the results related to the iteration 200 and 240, respectively, are reported.

From the table 1 and the previous figures, we can drawn these remarks:

- the plots of the behaviour of PSNRs show the convergent trend of the method PGDA, although this convergence is very slow due to the small value of the steplengths; for the model (7) the differences between the recovered images at the iteration K and at the iteration T are visually unappreciable, showing that the stopping criterion seems effective;
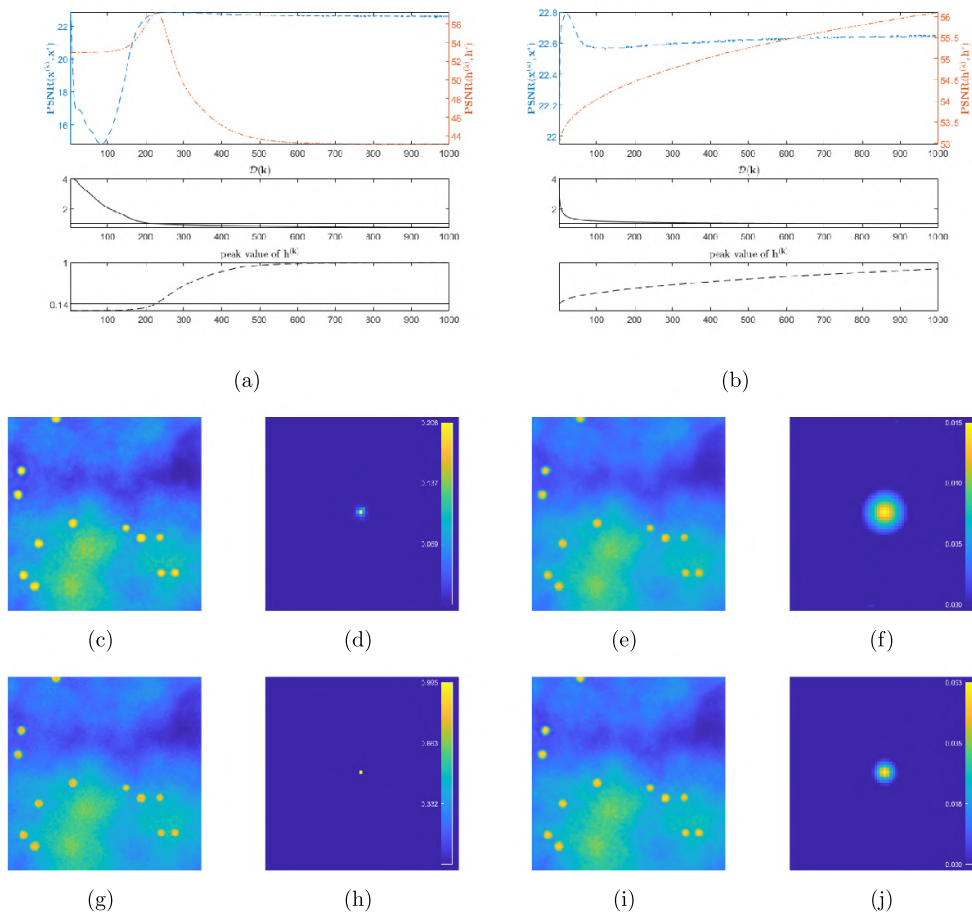
(a)   (b)

(c)   (d)   (e)   (f)

(g)   (h)   (i)   (j)

Figure 6. `synth001` test problem—The panels on the left show the behaviour of PGDA method for the model (6) $\ell_2 - \text{TV}$ ($\alpha_{\mathbf{P}} = 5 \times 10^{-5}, \alpha_{\boldsymbol{\mu}} = 5 \times 10^{-4}, \gamma_{\mathbf{Y}} = 1$): (a) PSNR of $\mathbf{x}^{(k)}$ and $\mathbf{h}^{(k)}$, normalized discrepancy $\mathcal{D}(k)$ and peak value of $\mathbf{h}^{(k)}$, (c)–(d) reconstructions of the image and the PSF at the iteration 240, (g)–(h) reconstructions of the image and the PSF at the iteration $T = 1000$. The panels on the right show the behaviour of PGDA method for the model (7) $\ell_2 - \text{TV}$: ($\alpha_{\mathbf{P}} = 10^{-6}, \alpha_{\boldsymbol{\mu}} = 10^{-5}, \gamma_{\mathbf{Y}} = 1, \gamma_{\mathbf{V}} = 10^{-3}$): (b) PSNR of $\mathbf{x}^{(k)}$ and $\mathbf{h}^{(k)}$, normalized discrepancy $\mathcal{D}(k)$ and peak value of $\mathbf{h}^{(k)}$, (e)–(f) reconstructions of the image and the PSF at the iteration $K = 70$, (i)–(l) reconstructions of the image and the PSF at the iteration $T = 1000$.

- regarding model (6), we observe that the PSNR of $\mathbf{h}^{(k)}$ reaches an high peak and then decreases in a very fast manner; in general, the peak does not correspond to a value of the recovered image satisfying the stopping criterion, although at this iteration the normalized discrepancy is close to 1 and the PSF peak value is near to the true upper bound of $\mathbf{h}^*$; the additional constraint of the PSF imposed in (7) seems to avoid this drawback; indeed, the stopping criterion provides more reliable results, above all on the reconstruction of the PSF. Furthermore, the reconstructions of the images obtained by solving (6) present several artifacts, for example in `rice` we observe an undesired high contrast and in both `micro` and `synth001` some halos are present.

Table 1. Figures of merit for the results obtained for the models (6) and (7) $\ell_2 - \text{TV}$. K denotes the iteration for which the stopping criterion is satisfied, k is an iteration intermediate which exhibits suitable results, although the stopping criterion is not satisfied, T is the iteration corresponding to the maximum number of steps.

Model	Iters.	PSNR($\mathbf{x}^{(k)}$)	PSNR($\mathbf{h}^{(k)}$)	SSIM($\mathbf{x}^{(k)}$)
			rice	
(6)	$k = 200$	19.81	61.81	0.59
	$T = 1000$	24.26	42.54	0.64
(7)	$K = 60$	24.26	62.85	0.65
	$T = 1000$	24.30	60.18	0.66
			micro	
(6)	$K = 860$	21.77	49.70	0.84
	$T = 1000$	22.09	49.30	0.84
(7)	$K = 400$	25.32	60.49	0.90
	$T = 1000$	25.76	59.16	0.91
			synth001	
(6)	$k = 240$	22.82	56.77	0.83
	$T = 1000$	22.61	43.07	0.82
(7)	$K = 70$	22.60	53.86	0.87
	$T = 1000$	22.65	56.09	0.86

Table 2. Figures of merit for the results obtained for the model (7) $\ell_2 - \ell_1$.

Iters	PSNR($\mathbf{x}^{(k)}$)	PSNR($\mathbf{h}^{(k)}$)	SSIM($\mathbf{x}^{(k)}$)
		rice	
$K = 120$	24.39	63.37	0.66
$T = 1000$	24.43	59.76	0.66
		micro	
$K = 600$	25.17	59.12	0.90
$T = 1000$	25.23	58.21	0.90
		synth001	
$K = 60$	22.62	53.82	0.87
$T = 1000$	22.65	56.40	0.86

4.1.2. DDIPP with different regularization terms on the image. In figure 7 we show the results obtained by DDIPP equipped with the regularization (8) for the blur and the ℓ_1-like for the image (equations (9) and (10)), i.e. $\ell_2 - \ell_1$. These results are obtained with the same settings used for the model (7) $\ell_2 - \text{TV}$. For completeness, in table 2, we report the merit figures obtained for the three test problems. From the comparison of figures 4–6 with figure 7, we observe that model (7) provides very similar results when it is combined with the TV-like or the ℓ_1-like regularization terms for the image. This is confirmed from the comparison of merit figures in tables 1 and 2.
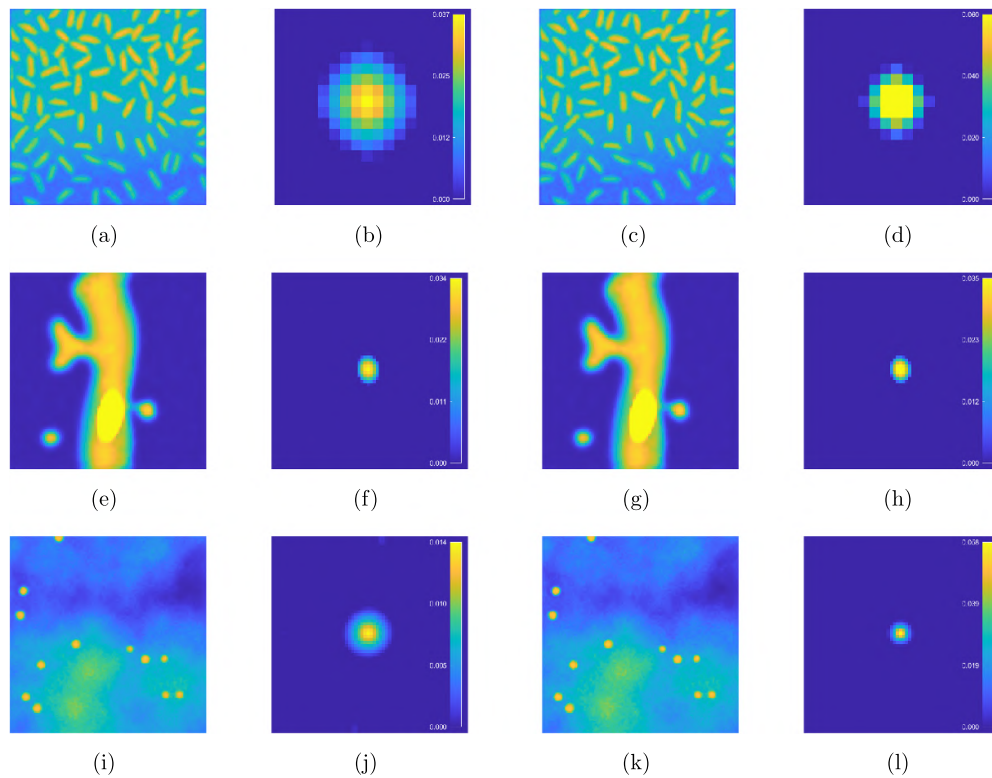
Figure 7. Results obtained for the model (7) $\ell_2 - \ell_1$. The first row shows the results for the problem `rice`: (a), (b) image and PSF at the iteration $K = 120$; (c), (d) image and PSF at the iteration $T = 1000$. The second row shows the results for the problem `micro`: (e), (f) image and PSF at the iteration $K = 600$; (g), (h) image and PSF at the iteration $T = 1000$. The third row shows the results for the problem `synth001`: (i), (l) image and PSF at the iteration $K = 60$; (m), (n) image and PSF at the iteration $T = 1000$.

4.1.3. Behaviour for different noise levels. For data corrupted by Poisson noise, the noise level depends on the values of any pixel of the object [1, section 3.4]. For example, for the `rice` object, with pixel values in the interval $[0, 255]$, a synthetic detected image is the one in figure 3(c); if the object is rescaled by a factor 10 or 0.5, the noise free image has the same morphology but the noisy blurred images are very different, as it can viewed by comparing figure 3(c) with figures 8(a) and (d) corresponding to the object rescaled by 10 and 0.5 respectively. The merit figures for the test problem rescaled by 10 are $\text{PSNR}(\mathbf{g}) = 23.08$ and $\text{SSIM}(\mathbf{g}) = 0.63$, whereas they are $\text{PSNR}(\mathbf{g}) = 21.73$ and $\text{SSIM}(\mathbf{g}) = 0.44$ when the object is rescaled by 0.5. The numerical results obtained for the model (7) $\ell_2 - \text{TV}$ are reported in figures 8(b) and (c) for the object rescaled by 10 and in figures 8(e) and (f) for the one rescaled by 0.5. For the former case, the stopping criterion is satisfied at the iteration $K = 400$ and the merit figures are $\text{PSNR}(\mathbf{x}^{(K)}) = 23.19$, $\text{PSNR}(\mathbf{h}^{(K)}) = 61.76$ and $\text{SSIM}(\mathbf{x}^{(k)}) = 0.65$; for the latter case, the stopping criterion is satisfied at the iteration $K = 90$ and the merit figures at this iteration are $\text{PSNR}(\mathbf{x}^{(K)}) = 24.16$, $\text{PSNR}(\mathbf{h}^{(K)}) = 63.39$ and $\text{SSIM}(\mathbf{x}^{(K)}) = 0.64$. The numerical results allow to observe that the model (7) appears robust with respect to the different noise levels. In particular, the reconstruction of the PSF appears to be sufficiently accurate in all the numerical tests. A similar behaviour has been observed in the other test problems.
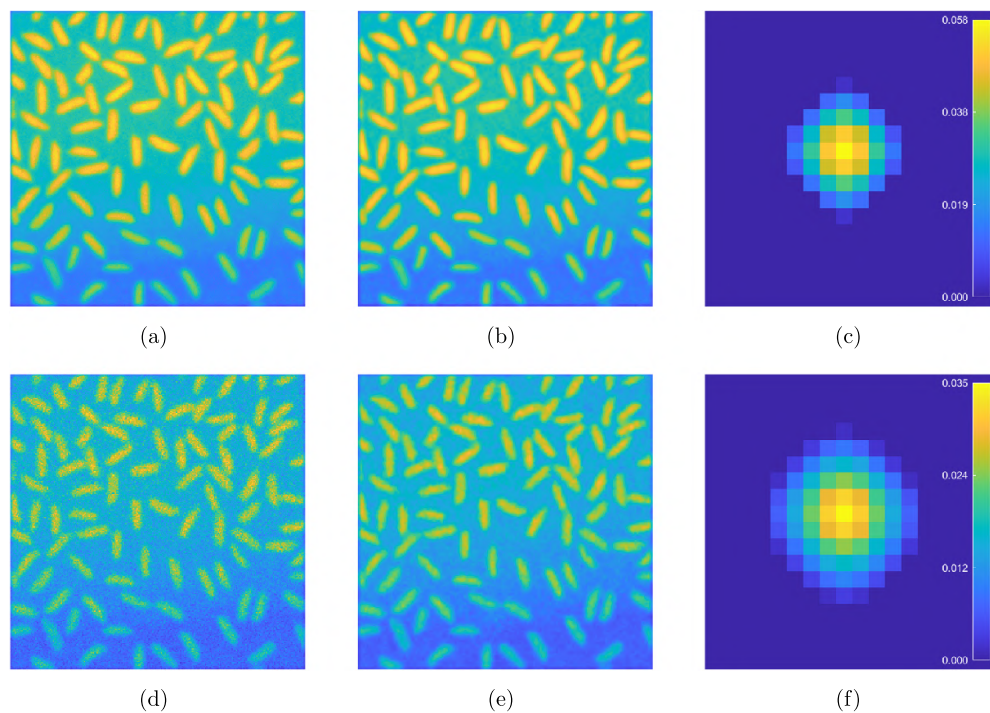
Figure 8. Results for different noise levels—In the upper panel: (a) **g** for the test problem `rice` when the object is rescaled by a factor 10; (b), (c) recovered image and PSF for the model (7) $\ell_2 -$ TV at the iteration $K = 400$. In the lower panel: (d) **g** for the test problem `rice` when the object is rescaled by a factor 0.5; (e), (f) recovered image and PSF for the model (7) $\ell_2 -$ TV at the iteration $K = 90$.

4.1.4. Comparison between DDIPP, SelfDeblur, improved SelfDeblur [26] and algorithm 3 in [20]. A further experiment is aimed to compare the results obtained by DDIPP with the ones achieved via other approaches. To the best of our knowledge, no method for NBD bases its model on the KL generalized divergence. Furthermore, most of the approaches use pre-trained networks. Therefore it seems difficult to identify methods with which to carry out a comparison. Based on these reasons, we consider the two methods which inspired DDIPP, namely SelfDeblur [21] and its improved version proposed in [26], referred to in this work as ImpSD. As regards SelfDeblur, we use the code available at https://github.com/csdwren/SelfDeblur and running it on Google Colab. On the other hand, since the code for ImpSD is not available online, the method has been implemented in MatLab®, by replicating the networks of SelfDeblur. The procedure in [26], while preserving the basic approach of SelfDeblur, adds regularization terms and suggests the initialization of the nets, as depicted in section 3.1. In the code, the MatLab® implementation of the functionals employed in the formulation (Least square, SSIM, TV) are used. In particular, the objective functional is modified among the iterations: following [26], the first 2000 iterations employ the Least Square loss together with the ℓ_2 regularization on the PSF, then it switches to the SSIM functional coupled with the TV

on the image. The regularization parameters are chosen as in [26]: 0.01 and 0.1 for TV and ℓ_2, respectively.

Another recent proposal is the third algorithm in [20], denoted in the following as ASA3. Indeed, the authors in [20] primarily use a different approach: the two networks $\mathcal{N}_h$ and $\mathcal{N}_x$ are *pre-trained* generative networks, and the procedure updates the *inputs* $\mathbf{z}_h$ and $\mathbf{z}_x$ of the networks and not their weights. One must have *a-priori* information on the type of images and on the blur type, together with large relative datasets, in order to have properly trained networks. Then, to overcome the issue of having a large image dataset for the training of $\mathcal{N}_x$, the authors propose to update also the weights of this network, together with its input and with $\mathbf{z}_h$. The functional employed in this setting is

$$\mathcal{F}(\mathbf{z}_h, \mathbf{z}_x, \boldsymbol{\theta}_x) = \|\mathbf{g} - \mathcal{N}_h(\mathbf{z}_h; \boldsymbol{\theta}_h) * \mathcal{N}_x(\mathbf{z}_x; \boldsymbol{\theta}_x)\|^2 + \beta_K \|\mathbf{z}_h\|^2 + \beta_X \text{TV}(\mathcal{N}_x(\mathbf{z}_x; \boldsymbol{\theta}_x)). \tag{26}$$

With an abuse of notation, in (26) we explicit the dependence of the nets on the random inputs $\mathbf{z}_h$ and $\mathbf{z}_x$. In our experimentation, in order to implement the ASA3 method, the network $\mathcal{N}_h$ has been trained for generate Gaussian PSFs; this network presents a different architecture with respect to the one used in [20] since the structure of the PSF is less articulate than the motion blurs used in [20]. Since the code for ASA3 in [20] is not available online, we re-coded it entirely in MatLab$^{\circledR}$. For clearness, we report the details of ASA3 in algorithm. In the numerical experiment, as in [20], we set different steplengths for the descent step: namely $\eta = 10^{-3}$ for updating the nets' inputs and $\eta = 10^{-4}$ for updating the weights. Moreover, we set $\beta_K = 10^{-2}$ and $\beta_X = 10^{-3}$. The number of iterations is set to $T = 5000$ and $T_x = 500$.

Algorithm 2. ASA3 [20]

Set η, draw $\mathbf{z}_h \sim \mathcal{N}(0,1), \mathbf{z}_x \sim \mathcal{N}(0,1)$. Choose T_x, T.
Initialise $\mathcal{N}_x$ solving (23) with a max number of iterations equal to T_x
    for $t = 1, 2, \ldots, T$ **do**
    $\mathbf{z}_h^{(k+1)} \leftarrow \mathbf{z}_h^{(k)} - \eta \nabla_{\mathbf{z}_h} \mathcal{F}(\mathbf{z}_h^{(k)}, \mathbf{z}_x^{(k)}, \boldsymbol{\theta}_x^{(k)})$
    $\mathbf{z}_x^{(k+1)} \leftarrow \mathbf{z}_x^{(k)} - \eta \nabla_{\mathbf{z}_x} \mathcal{F}(\mathbf{z}_h^{(k)}, \mathbf{z}_x^{(k)}, \boldsymbol{\theta}_x^{(k)})$
    $\boldsymbol{\theta}_x^{(k+1)} \leftarrow \boldsymbol{\theta}_x^{(k)} - \eta \nabla_{\boldsymbol{\theta}_x} \mathcal{F}(\mathbf{z}_h^{(k)}, \mathbf{z}_x^{(k)}, \boldsymbol{\theta}_x^{(k)})$
end for

Figure 9 and table 3 present the results of SelfDeblur, ImpSD and ASA3 carried on the syntetic database; bearing in mind figures 4–6 and table 1 (see also figure 7 and table 2), it is possible to make a comparison between the results of DDIPP and the ones obtained by the aforementioned methods. As noted in [27, section 3.1.2], the cropping procedure in SelfDeblur produces artifacts and misplacements in the final reconstruction. Indeed, unlike SelfDeblur, the proposed DDIPP strategy considers a larger size of just the PSF and not of the image, since the boundary conditions are encompassed in the convolutional filters. ImpSD instead provides slightly better results, in terms of placement, but several details are missing; in particular, the image `synth001` suffers from several artifacts. Finally, the algorithm ASA3 seems to recover reliable image and PSF at an intermediate iteration. Indeed, figure 9 shows the results of ASA3 at the iteration K where we observe the highest PSNR value of the image $\mathbf{x}^{(K)}$ over 5000 iterations, but for real applications, where the ground truth is not known, this is not generally possible. In some cases, the recovered image has a high PSNR, despite having artifacts (see
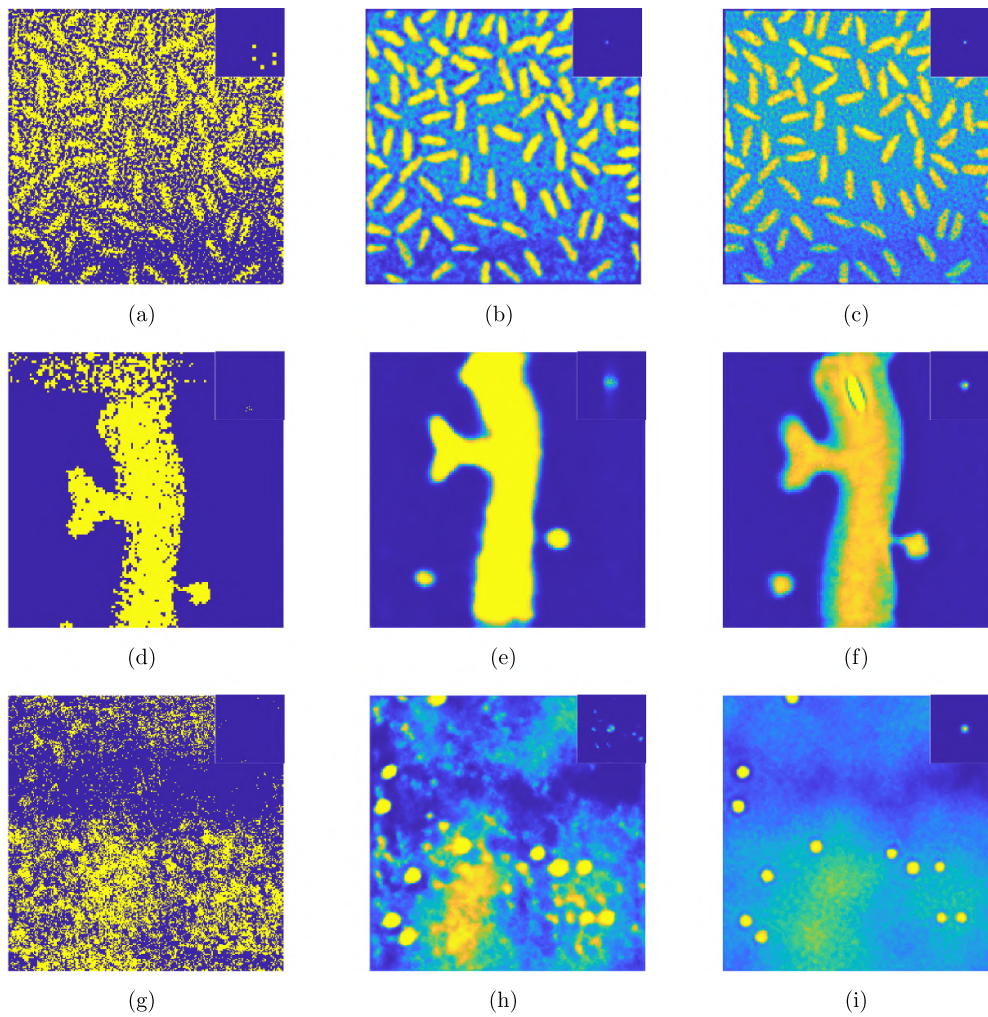
Figure 9. Results of the comparison for the synthetic database—In the first row `rice`: (a) recovered image via SelfDeblur, (b) recovered images via ImpSD; (c) recovered image via ASA3. In the second row `micro`: (d) recovered image via SelfDeblur, (e) recovered images via ImpSD; (f) recovered image via ASA3. In the third row `synth001`: (g) recovered image via SelfDeblur, (h) recovered images via ImpSD; (i) recovered image via ASA3.

`micro`), in other cases it remains noisy (see `rice`, `synth001`). At the iteration *K*, the recovered blur kernels also appear reliable, although slightly asymmetrical. However, after the iteration *K*, we observe a subsequent degradation to the point form. Finally, we observe that DDIPP in general provides a better PSF reconstruction.

Table 3. Figures of merit for the results obtained with SelfDeblur (5000 iterations), ImpSD (5000 iterations) and ASA3, For the ASA3 method we report the results obtained at the iteration K which provides the maximum PSNR of the image within 5000 iterations.

Algorithm	Iters	PSNR($\mathbf{x}^{(k)}$)	PSNR($\mathbf{h}^{(k)}$)	SSIM($\mathbf{x}^{(k)}$)
			rice	
SelfDeblur	$T = 5000$	6.45	17.67	0.01
ImpSD	$T = 5000$	14.54	60.70	0.41
ASA3	$K = 400$	17.30	64.98	0.42
			micro	
SelfDeblur	$T = 5000$	9.34	29.12	0.20
ImpSD	$T = 5000$	16.24	55.53	.71
ASA3	$K = 300$	26.48	65.03	.88
			synth001	
SelfDeblur	$T = 5000$	5.41	26.54	0.08
ImpSD	$T = 5000$	15.15	53.12	0.30
ASA3	$K = 150$	22.74	55.45	0.66

4.2. Real dataset

This section is devoted to apply the proposed DDIPP to real-world microscopic images. Such images are the scanning of a 3D volume, with dimension $64 \times 64 \times 4.1$ μm: this volume is then recorded in an array with dimension $512 \times 512 \times 10$ voxels. The radius of the spherical particles is 1.5 μm and they are suspended in a $\sim 70\% - 30\%$ glycerol/water mixture (viscosity of approximately 0.017 Pa s). The microscope employed for acquiring this dataset is a Zeiss LSM 700 with a 100 $\times$ NA 1.4 oil immersion objective (Zeiss Plan-APOCHROMAT). The architectures for $\mathcal{N}_x$ and $\mathcal{N}_h$ take as input just 2D images, not 3D volumes: hence we consider one frame of the 3D volume at time. This dataset contains several hundreds of images, but due to hardware limitation we consider only 3 of them, shown in figure 10. On the base of the technical documentation of the instrument, the parameter H that imposes the upper bound on the peak of the PSF is approximately estimated as 0.2.

The three problems employed DDIPP for the model (7) $\ell_2 - \text{TV}$ and $\ell_2 - \ell_1$. The results are carried out with $\alpha_{\mathbf{P}} = 10^{-6}, \alpha_{\boldsymbol{\mu}} = 10^{-5}, \gamma_{\mathbf{Y}} = 1, \gamma_{\mathbf{V}} = 10^{-3}$; the other parameters are as specified at the beginning of section 4. The results are depicted in figures 11 and 12 in the case of $\ell_2 - \text{TV}$ and $\ell_2 - \ell_1$ respectively. In both the figures, the recovered images and related PSFs are reported. The image and PSF referring to the first test problem (first column) are obtained at the maximum number $T = 1000$ of iterations for both cases; the results related to the second and third test problems (second and third columns respectively) are obtained at the iterations satisfying the stopping criterion, i.e. $K = 570$ and $K = 680$, respectively, for
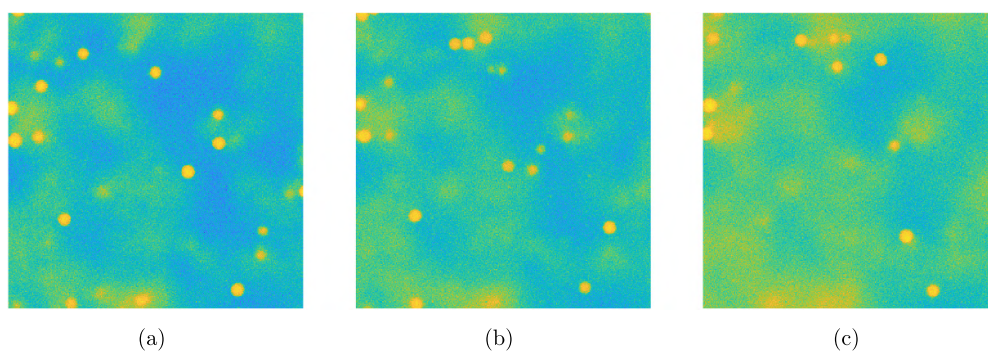
Figure 10. Detected microscopic images; their size is 512×512.
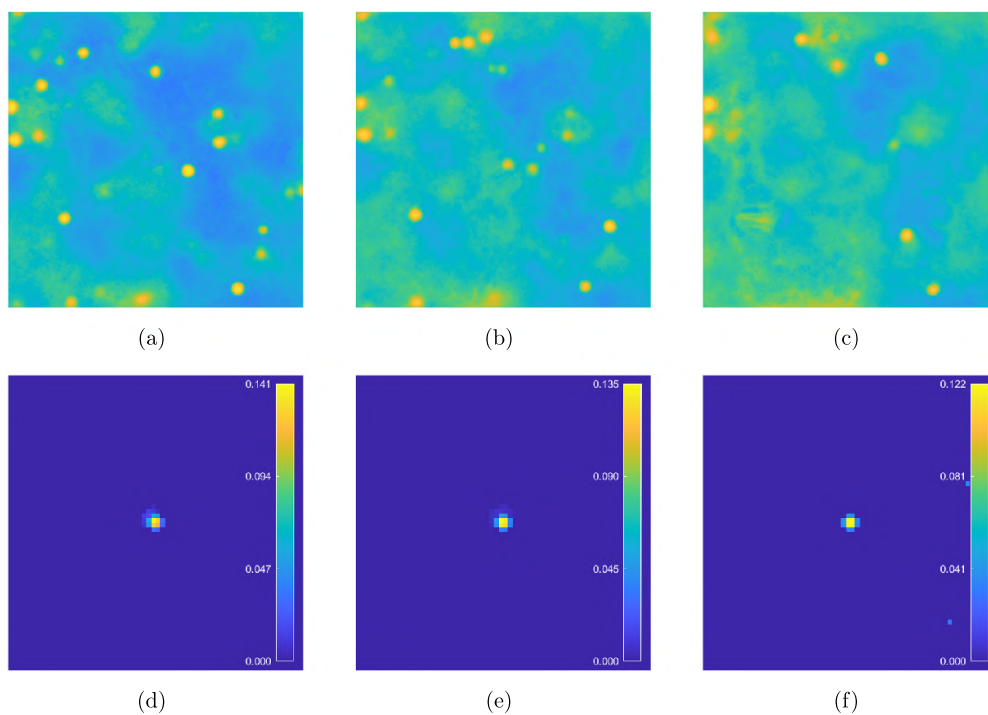

Figure 11. Real data : results for the model (7) $\ell_2 - \text{TV}$; in particular (a),(d): recovered image and PSF for the detected image in figure 10(a) at the iteration 1000; (b),(e): recovered image and PSF for the detected image in figure 10(b) at the iteration $K = 570$; (c), (f): and PSF for the detected image in figure 10(c) at the iteration $K = 680$.
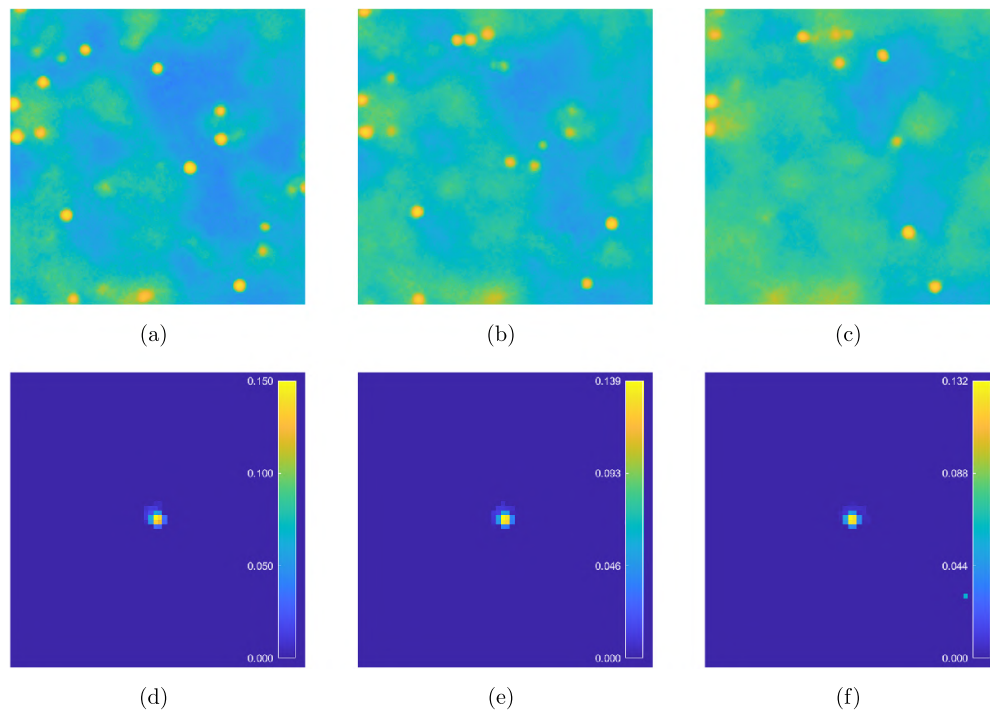
Figure 12. Real data : results for the model (7) $\ell_2 - \ell_1$; in particular (a), (d): recovered image and PSF for the detected image in figure 10(a) at the iteration 1000; (b), (e): recovered image and PSF for the detected image in figure 10(b) at the iteration 1000; (c), (f): recovered image and PSF for the detected image in figure 10(c) at the iteration 1000.

the model (7) $\ell_2 - \text{TV}$ and at the maximum number $T = 1000$ of iterations for the model (7) $\ell_2 - \ell_1$. We remark that, although the value of the maximum number of iterations has been reached, the stopping criterion based on the condition (24) has been satisfied in all cases but the value of the normalized discrepancy (25) has remained greater than 2 (but less than 3).

It is clear that, as anticipated from the experiments on synthetic data, the contrast is improved, the particles are more enhanced with respect to the background, both in the more diffuse regions and in the darker ones. Furthermore, as expected, the recovered PSF is very similar in all cases, with a peak value between 0.12 and 0.15. No substantial differences are noticed between the use of $\ell_2 - \text{TV}$ and $\ell_2 - \ell_1$; in the case of some balls not very distinct from the background in the detected image, the reconstruction with ℓ_1 seems to recover slightly better their position (see figures 11(c) and 12(c)).

The importance of the right choice for the functional to minimize and the role of the regularizers are again evident from the results achieved by SelfDeblur, ImpSD and ASA3 after 5000 iterations, depicted in figure 13. In general, the behaviour of ImpSD seems to be just denoising and not deblurring, since the recover of the PSF is the Dirac's delta and only perceivable effect is the one by the TV.
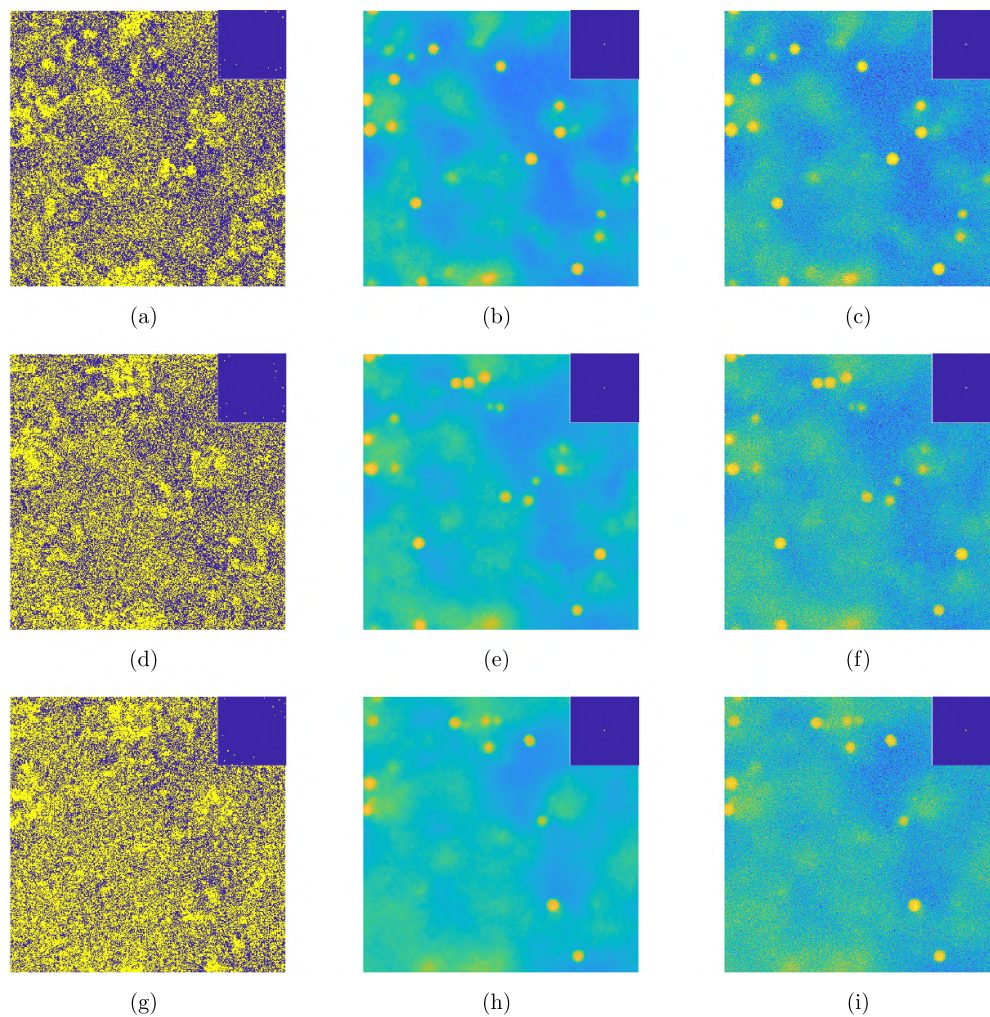
Figure 13. Recovered images for the detected data in figures 10(a)–(c) achieved after 5000 iterations via SelfDeblur (first column), via ImpSD (second column) and via ASA3 (third column).

5. Conclusion

We developed a novel procedure, named DDIPP, for a BD approach, based on PGDA-like algorithm, for the recovering of microscopy images affected by Poisson noise. We adopted a double Deep Image Prior strategy, both on the image and on the PSF, where the loss function comprehends the KL functional, for handling of Poisson noise, and two adaptive regularization terms: a squared ℓ_2 norm on the PSF and a TV or a ℓ_1 norm on the image. Furthermore technical information on the estimated peak of the blur kernel is exploited to avoid trivial solutions. The numerical experience on synthetic and real images showed early promising results: the contrast on final images is improved and the object of interest, in this case the spherical beads, are well recovered.

Future work foresees the improvement of the model, by including other typical features of the image formation system, as for example, the flux preservation. A further possible research direction is to test different and more performing architectures for the networks and to deepen the analysis about suitable early stopping criteria. Another crucial point is to achieve a more efficient implementation of the code, in order to consider 3D images, as the one used in section 4.2 and therefore to obtain the estimation of more realistic PSFs in 3 dimensions.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/AleBenfe/DDIPP.

ORCID iDs

A Benfenati ⓘ https://orcid.org/0000-0002-2985-374X
A Catozzi ⓘ https://orcid.org/0000-0001-6378-3063
V Ruggiero ⓘ https://orcid.org/0000-0001-5885-1703

References

[1] Bertero M, Boccacci P and Ruggiero V 2018 *Inverse Imaging With Poisson Data* (Bristol: IOP Publish) (https://doi.org/10.1088/2053-2563/aae109)
[2] Campisi P and Egiazarian K (eds) 2016 *Blind Image Deconvolution: Theory and Applications* (Boca Raton, FL: CRC Press)
[3] Levin A, Weiss Y, Durand F and Freeman W T 2009 Understanding and evaluating blind deconvolution algorithm *IEEE Conf. Computer Vision and Pattern Recognition* pp 1964–71
[4] van Kempen G M P, van Vliet L J, Verveer P J and van der Voort H T M 1997 A quantitative comparison of image restoration methods for confocal microscopy *J. Microsc.* **185** 354–65
[5] van Kempen G M P and van Vliet L J 2000 Background estimation in nonlinear image restoration *J. Opt. Soc. Am.* A **17** 425–33
[6] Bertsekas D P 1999 *Nonlinear Programming* 2nd edn (Belmont: Athena Scientific)
[7] Grippo L and Sciandrone M 2000 On the convergence of the block nonlinear Gauss-Seidel method under convex constraints *Oper. Res. Lett.* **26** 127–36
[8] Bonettini S 2011 Inexact block coordinate descent methods with application to the nonnegative matrix factorization *IMA J. Num. Anal.* **31** 1431–52
[9] Prato M, La Camera A, Bonettini S and Bertero M 2013 A convergent blind deconvolution method for post-adaptive-optics astronomical imaging *Inverse Problems* **29** 065017
[10] Zuo W, Ren D, Zhang D, Gu S and Zhang L 2016 Learning iteration-wise generalized shrinkage–thresholding operators for blind deconvolution *IEEE Trans. Image Process.* **25** 1751–64
[11] Pan J, Sun D, Pfister H and Yang M H 2018 Deblurring images via dark channel prior *IEEE Trans. Pattern Anal. Mach. Intell.* **40** 2315–28

[12] Zhang M, Fang Y, Ni G and Zeng T 2022 Pixel screening based intermediate correction for blind deblurring *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)* pp 5892–900

[13] Liu J, Yan M and Zeng T 2021 Surface-aware blind image deblurring *IEEE Trans. Pattern Anal. Mach. Intell.* **43** 1041–55

[14] Nah S, Hyun Kim T and Mu Lee K 2017 Deep multi-scale convolutional neural network for dynamic scene deblurring *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* pp 257–65

[15] Tao X, Gao H, Shen X, Wang J and Jia J 2018 Scale-recurrent network for deep image deblurring *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 8174–82

[16] Zhang J, Pan J, Ren J, Song Y, Bao L, Lau R W H and Yang M 2018 Dynamic scene deblurring using spatially variant recurrent neural networks *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition*
pp 2521–9

[17] Gao H, Tao X, Shen X and Jia J 2019 Dynamic scene deblurring with parameter selective sharing and nested skip connections *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* pp 3843–51

[18] Koh J, Lee J and Yoon S 2021 Single-image deblurring with neural networks: a comparative survey *Comput. Vis. Image Underst.* **203** 103134

[19] Tran P, Tran A, Phung Q and Hoai M 2021 Explore image deblurring via encoded blur kernel space *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* p 2021

[20] Asim M, Shamshad F and Ahmed A 2020 Blind image deconvolution using deep generative priors *IEEE Trans. Comput. Imaging* **6** 1493–506

[21] Ren D, Zhang K, Wang Q, Hu Q and Zuo W 2020 Neural blind deconvolution using deep priors *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 3338–47

[22] Ulyanov D, Vedaldi A, Lempitsky V Prior D I 2018 *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* pp 9446–54

[23] Ulyanov D, Vedaldi A and Lempitsky V 2020 Deep Image Prior *Int. J. Comput. Vis.* **128** 1867–88

[24] Gandelsman Y, Shocher A and Irani M 2019 Double-DIP: unsupervised image decomposition via coupled deepimage- priors *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*

[25] Wang Z, Wang Z, Li Q and Bilen H 2019 Image deconvolution with deep image and kernel priors *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*

[26] Kotera J, Šroubek F and Šmídl W 2021 Improving neural blind deconvolution *2021 IEEE Int. Conf. on Image Processing (ICIP)* pp 1954–8

[27] Zhuang Z, Li T, Wang H and Sun J 2022 Blind image deblurring with unknown kernel size and substantial noise (arXiv:2208.09483)

[28] Cascarano P, Franchini G, Kobler E, Porta F and Sebastiani A 2022 Constrained and unconstrained deep image prior optimization models with automatic regularization *Comput. Optim. Appl.* **84** 1–25

[29] Boţ R I and Böhm A 2022 Alternating proximal-gradient steps for (stochastic) nonconvex-concave minimax problems (arXiv:2007.13605v3)

[30] Chen Z, Zhou Y, Xu T and Liang Y 2021 Proximal gradient descent-ascent: variable convergence under kl geometry (arXiv:2102.04653)

[31] Wang H, Li T, Zhuang Z, Chen T, Liang H and Sun J 2021 Early stopping for deep image prior *CoRR* (arXiv:abs/2112.06074)

[32] Bertero M, Boccacci P, Talenti G, Zanella R and Zanni L 2010 A discrepancy principle for Poisson data *Inverse Problems* **26** 105004

[33] Bauschke H H and Combettes P L 2011 *Convex Analysis and Monotone Operator Theory in Hilbert Spaces* (New York: Springer Science & Business Media)

[34] Pardalos P M and Kovoor N 1990 An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds *Math. Program.* **46** 321–8

[35] Dai Y H and Fletcher R 2006 New algorithms for singly linearly constrained quadratic programming problems subject to lower and upper bounds *Math. Program.* **106** 403–21

[36] Kiwiel K C 2008 Breakpoint searching algorithms for the continuous quadratic knapsack problem *Math. Program.* **112** 473–91

[37] Chen W T, Zhu A Y, Khorasaninejad M, Shi Z, Sanjeev V and Capasso F 2017 Immersion meta-lenses at visible wavelengths for nanoscale imaging *Nano Lett.* **17** 3188–94

[38] Ashida Y, Honma Y, Miura N, Shibuya T, Kikuchi H, Tamada Y, Kamei Y, Matsuda A and Hattori M 2020 Imaging performance of microscopy adaptive-optics system using scene-based wavefront sensing *J. Biomed. Opt.* **25** 123707

[39] Sitzmann V, Martel J N P, Bergman A W, Lindell D B and Wetzstein G 2020 Implicit neural representations with periodic activation functions *CoRR* (arXiv:abs/2006.09661)

[40] Cascarano P, Franchini G, Porta F and Sebastiani A 2023 On the First-Order Optimization Methods in Deep Image Prior *J. Verif. Valid. Uncertain. Quantif.* **7** 041002

[41] Borgia G C, Brown R J S and Fantazzini P 1998 Uniform-penalty inversion of multiexponential decay data *J. Magn. Reson.* **132** 65–77

[42] Grasmair M 2009 Locally adaptive total variation regularization *Scale Space and Variational Methods in Computer Vision* (*Lecture Notes in Computer Science*) (Berlin: Springer) pp 331–42

[43] Bortolotti V, Brown R, Fantazzini P, Landi G and Zama F 2016 Uniform penalty inversion of two-dimensional NMR relaxation data *Inverse Problems* **33** 103134

[44] Zanella R, Boccacci P, Zanni L and Bertero M 2009 Efficient gradient projection methods for edge-preserving removal of poisson noise *Inverse Problems* **25** 045010

[45] Zanella R, Boccacci P, Zanni L and Bertero M 2013 Corrigendum: efficient gradient projection methods for edge-preserving removal of poisson noise *Inverse Problems* **29** 119501

[46] Willett R M and Nowak R D 2003 Platelets: a multiscale approach for recovering edges and surfaces in photon limited medical imaging *IEEE Trans. Med. Imaging* **22** 332–50

[47] Benfenati A 2022 upU-Net approaches for background emission removal in fluorescence microscopy *J. Imaging* **8** 142

[48] Kirshner H, Aguet F, Sage D and Unser M 2013 3-D PSF fitting for fluorescence microscopy: Implementation and localization application *J. Microsc.* **249** 13–25

[49] Sage D, Donati L, Soulez F, Fortun D, Schmit G, Seitz A, Guiet R, Vonesch C and Unser M 2017 DeconvolutionLab2: an open-source software for deconvolution microscopy *Methods Image Process. Biol.* **115** 28–41

[50] Sage D *et al* 2019 Super-resolution fight club: assessment of 2D and 3D single-molecule localization microscopy software *Nat. Methods Tech. Life Sci. Chem.* **16** 387–95