



HAL
open science

Deep-Blur : Blind Identification and Deblurring with Convolutional Neural Networks

Valentin Debarnot, Pierre Weiss

► **To cite this version:**

Valentin Debarnot, Pierre Weiss. Deep-Blur : Blind Identification and Deblurring with Convolutional Neural Networks. 2022. hal-03687822v1

HAL Id: hal-03687822

<https://hal.science/hal-03687822v1>

Preprint submitted on 3 Jun 2022 (v1), last revised 15 Nov 2024 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Deep-Blur : Blind Identification and Deblurring with Convolutional Neural Networks

Valentin Debarnot & Pierre Weiss

Abstract—We propose a neural network architecture and a training procedure to estimate blurring operators and deblur images from a single degraded image. The key assumption is that the forward operators can be parameterized by a low-dimensional vector. The considered models include a description of the point spread function with Zernike polynomials in the pupil plane or product-convolution expansions, which allow to tackle space varying operators. Numerical experiments reveal that the proposed method recovers the blur parameters robustly even for large noise levels. This estimate can then be used as an input of an unrolled neural network to deblur the image. It is trained with a specific sampling procedure adapted to a family of parameterized operators. This strategy provides a fast and accurate blind inverse problem solver, requiring nearly no human interaction once the operator parameterization has been set up. It adapts to arbitrary noise levels and considerably improves alternative blind deblurring softwares in the examples considered in this paper, especially in the high noise regimes.¹

Index Terms—Blind deblurring, Blind inverse problems, Identification network, Spatially variant blur, Deep learning, Unrolled network

I. INTRODUCTION

Image deblurring and super-resolution consist in recovering a sharp image $\bar{\mathbf{x}}$ from its blurred and sub-sampled version $\mathbf{y} = \mathcal{P}(\bar{\mathbf{A}}\bar{\mathbf{x}})$, where $\bar{\mathbf{A}} \in \mathbb{R}^{M \times N}$ is a discretized linear integral operator describing the acquisition process and $\mathcal{P} : \mathbb{R}^M \rightarrow \mathbb{R}^M$ is some perturbation modelling noise, quantization, saturation... It plays an important role in bio-medical and astronomical imaging, where physical phenomena such as diffraction and turbulence strongly reduce the achievable resolution. It also received a continual attention in computer vision, where objects that are moving or out-of-focus create significant artifacts. When the operator $\bar{\mathbf{A}}$ describing the optical system is available, this problem can be solved with mature variational inverse problem solvers [2] or data-driven approaches [3].

However, deriving a precise forward model requires specific calibration procedures, well controlled imaging environments and/or highly qualified staff. In addition, model mismatches result in distorted reconstructions. This can lead to dramatic performance loss, especially for super-resolution applications [4].

An alternative to a careful calibration step consists in solving the problem blindly: the forward model $\bar{\mathbf{A}}$ is estimated together with the sharp image $\bar{\mathbf{x}}$. Unfortunately, this blind inverse problem is highly degenerate. There is no hope to recover the sharp image without prior assumptions on $\bar{\mathbf{x}}$ and $\bar{\mathbf{A}}$. For instance, assume that $\bar{\mathbf{A}}$ is a discrete convolution

operator with some kernel $\bar{\mathbf{h}}$, *i.e.*, $\mathbf{y} = \bar{\mathbf{h}} \star \bar{\mathbf{x}}$. Then the couple $(\bar{\mathbf{h}}, \bar{\mathbf{x}})$ can be recovered only up to a large group of transformations [5]. For instance, the identity and blurred image are a trivial solution, the image and kernels can be shifted in opposite directions or scaled with inverse factors. Therefore, it is critical to introduce regularization terms both for the operator $\bar{\mathbf{A}}$ and the signal \mathbf{x} .

The main objective of this work is to design a blind inverse problem solver under the two assumptions below:

- The operator $\bar{\mathbf{A}}$ can be parameterized by a low-dimensional vector. In what follows, we let $\mathbf{A} : \mathbb{R}^K \rightarrow \mathbb{R}^{M \times N}$ denote the operator mapping and we assume that $\bar{\mathbf{A}} = \mathbf{A}(\bar{\gamma})$ for some $\bar{\gamma} \in \mathbb{R}^K$.
- The signal $\bar{\mathbf{x}}$ lives in a family \mathcal{X} with some known distribution $\mathcal{L}_{\mathcal{X}}$.

We propose a specific convolutional neural architecture and a training procedure to recover the couple $(\bar{\gamma}, \bar{\mathbf{x}})$ from the degraded data \mathbf{y} and the mapping $\mathbf{A}(\gamma)$. A first network identifies the parameterization $\bar{\gamma}$, while the second uses this parameterization to estimate the image $\bar{\mathbf{y}}$. This results in an efficient algorithm to sequentially estimate the blur operator and the sharp image $\bar{\mathbf{x}}$. The network architecture is shown on Fig. 1. Even though the work can be adapted to arbitrary inverse problems beyond image deblurring, we showcase its efficiency for challenging deblurring tasks involving convolutions, but also more advanced space-varying operators.

A. Related works

Solving blind deblurring problems is a challenging task that started being studied in the 1970's [6]. Fifty years later, it seems impossible to perform an exhaustive review of existing methods and the following description will be lacunary. The prevailing approach is to estimate the original signal and the blur operator by solving variational problems of the form:

$$\inf_{\mathbf{A} \in \mathbb{R}^{M \times N}, \mathbf{x} \in \mathbb{R}^N} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + R_A(\mathbf{A}) + R_x(\mathbf{x}), \quad (1)$$

where $R_A : \mathbb{R}^{M \times N} \rightarrow \mathbb{R} \cup \{+\infty\}$ and $R_x : \mathbb{R}^N \rightarrow \mathbb{R} \cup \{+\infty\}$ are regularization terms for the operator and the signal respectively. This problem can then be attacked with various types of alternating minimization procedures [7]. The solutions can be understood as maximum a posteriori estimates [8]. Before the advent of data-driven approaches, the regularizers were carefully designed to target specific features. The point spread functions can be considered as sparse and compactly supported for motion deblurring [9], [10], [11], [12], [13], [14], [15], [16]. They are smooth for diffraction limited systems [17] and can also be parameterized with Zernike polynomials

¹A preliminary version of this work was published in IEEE ISBI 2021 [1].

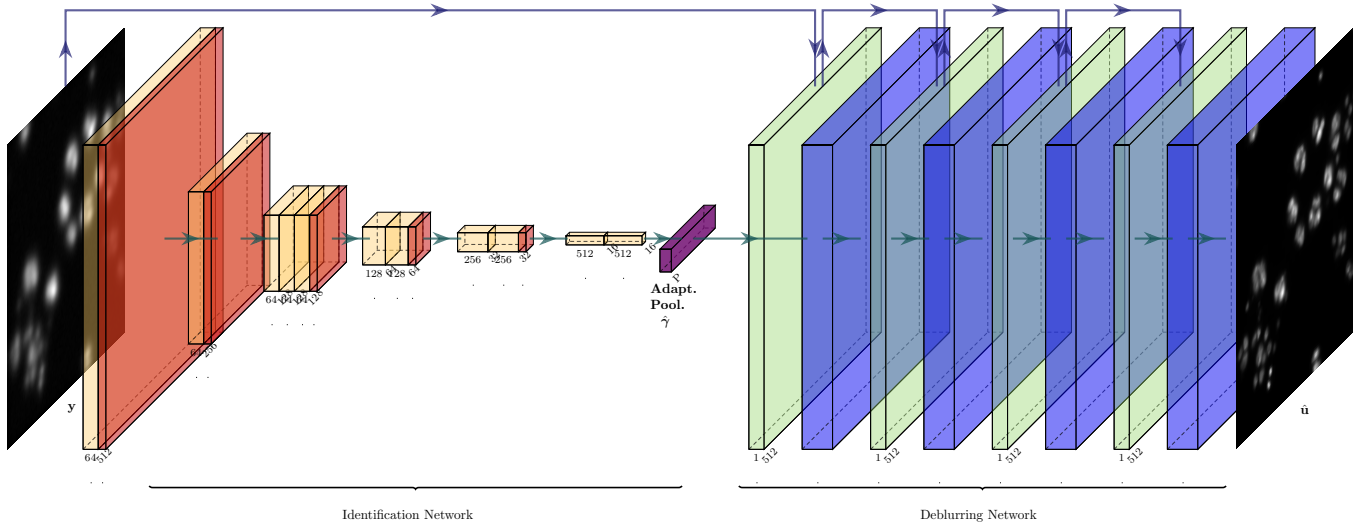


Fig. 1: The Deep-Blur architecture. The first part of the network identifies the parameterization $\hat{\gamma}$. In this paper, we use a ResNet architecture. The estimated parameterization $\hat{\gamma}$ is given as an input of a second deblurring network. This one is an unrolled Douglas-Rachford algorithm. The yellow blocks are convolution layers with ReLU and batch normalization. The red ones are average pooling layers. The green ones are regularized inverse layers of the form $\mathbf{x}_{t+1} = (\mathbf{A}^*(\hat{\gamma})\mathbf{A}(\hat{\gamma}) + \lambda\mathbf{I})^{-1}\mathbf{A}(\hat{\gamma})\mathbf{y}$. The violet blocks are UNET like neural networks with weights learned to provide a sharp image $\hat{\mathbf{x}}$.

in the pupil plane [18], [19], [20], [21], [5]. The images can be considered as sparse in microscopy and astronomical imaging [22], [23] or piecewise constant for natural images. The typical regularizer R_x is then the total variation, or more advanced priors on the image gradient [17], [24], [25], [13], [14], [15]. Some authors also advocate for the use of priors on the image spectral [26], [27], which allow transforming the blind deconvolution problem as a phase retrieval problem under ideal conditions.

The most recent variants of these approaches provide excellent results for specific applications such as text deblurring, see *e.g.*, [28]. However, they strongly rely on the detection of specific features (points, edges, textures) which may be absent or inaccurate models of the typical image features. In addition, the problem (1) or its derivatives are usually highly non-convex and the initialization must be chosen very carefully to ensure local convergence to the right minimizer. As a result, these methods need to be carefully tuned for a specific application fields for success.

In the last five years, machine learning approaches have emerged and now seem to outperform carefully handcrafted ones, at least in well controlled conditions. These approaches can be divided in two categories. The first category concerns end-to-end methods that directly estimate the reconstructed image from the observation [29], [30], [31], [32], [33], [34]. The second category contains approaches that produce an estimation of the blur operator and the original image. These approaches are specifically tuned for applications in computer vision [35], [29], [36], [37], [38] (motion and out-of-focus blurs) or diffraction limited systems [39], [40], [41]. Our work rather falls in the second category since space varying motion blurs are hardly captured by a low-dimensional parameterization. Notice that while the diversity of motion blurs is higher, their inversion is typically easier since their Fourier decay is

much slower.

In this list of references, the idea closest to ours is from Shajkofci and Liebling [40], [41]. Therein, the authors estimate a decomposition of the point spread function from a single image using a low dimensional parameterization such as a decomposition over Zernike polynomials. The spatial variations are then estimated by splitting the observation domain in patches where the blur is assumed locally invariant. The image can then be deblurred using a Richardson-Lucy algorithm based on the estimated operator.

B. Contributions

In this work, we propose to use a pair of convolutional neural networks to first estimate the operator parameterization $\bar{\gamma} \in \mathbb{R}^K$ and then use this parameterization to estimate the sharp image $\bar{\mathbf{x}} \in \mathbb{R}^N$ with a second convolutional neural network. The first network is the popular ResNet [42] as in [40]. The second network has the structure of an unrolled algorithm, which offers the advantage of adapting to the forward operator [43], [44], [45]. We call the resulting algorithm Deep-Blur, see Fig. 1. This work contains various original features:

- It includes space varying blur operators. In particular, we show that this approach allows to use the characterization of an optical system as a low-dimensional subspace of operators proposed in [46], [47]. Most approaches in the literature decompose the observation space into patches and treat each patch independently. In this work, we consider operators with an impulse response that varies continuously in the field of view.
- The resulting deblurring network is able to adapt to different forward models and to handle model mismatches naturally. This issue is an important concern for the use of model-based inverse problem solvers [48], [49], [50].

As will be discussed later, our approach can be seen as an intermediate step between the plug&play algorithms [51], [52] and the unrolled algorithms [43].

- To train Deep-Blur, we propose a specific sampling procedure of parameterized operators, assuming that the operators parameters live in a compact set.
- We evaluate the efficiency, robustness and stability of the proposed approach on various challenging problems, showing that the method is reliable and accurate.

The *Python* implementation of our method is freely available: www.github.com/DeepBlur.

C. Outline

In Section III, we introduce and motivate the mathematical framework. In Section IV, we present the Deep-Blur architecture and learning framework. In Section V, we validate the method on synthetic examples.

II. NOTATION

The bold font refers to vectors, matrices or vector-valued functions. The i -th value of a vector \mathbf{x} is denoted $\mathbf{x}[i]$ or x_i . \mathbf{I}_M denotes the identity matrix of size $M \times M$. For two operators \mathbf{A} and \mathbf{B} in $\mathbb{R}^{M \times N}$, we let $\langle \mathbf{A}, \mathbf{B} \rangle \stackrel{\text{def.}}{=} \text{Trace}(\mathbf{A}^T \mathbf{B})$ and the Frobenius norm is defined by $\|\mathbf{A}\|_F = \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle}$. A family of K operators $(\mathbf{A}_k) \in \mathbb{R}^{M \times N}$ is said to be orthogonal if

$$\langle \mathbf{A}_k, \mathbf{A}_{k'} \rangle = \begin{cases} 0 & \text{if } k \neq k' \\ 1 & \text{otherwise} \end{cases}.$$

The Hausdorff distance between two sets $\mathcal{X}, \mathcal{Y} \subset \mathbb{R}^M$ is defined by

$$\text{dist}(\mathcal{X}, \mathcal{Y}) \stackrel{\text{def.}}{=} \max \left(\sup_{\mathbf{x} \in \mathcal{X}} \inf_{\mathbf{y} \in \mathcal{Y}} \|\mathbf{x} - \mathbf{y}\|_2, \sup_{\mathbf{y} \in \mathcal{Y}} \inf_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\|_2 \right).$$

We let $S_{P-1} = \{\boldsymbol{\lambda} \in \mathbb{R}^P, \lambda_p \geq 0, \sum_{p=1}^P \lambda_p = 1\}$ denote the unit simplex. Let $(\boldsymbol{\gamma}_p)_{1 \leq p \leq P} \in \mathbb{R}^K$ denote a set of P points in \mathbb{R}^K , the convex cone generated by these vectors is defined by

$$\text{cone}(\boldsymbol{\gamma}_p, 1 \leq p \leq P) \stackrel{\text{def.}}{=} \left\{ \sum_{p=1}^P \lambda[p] \boldsymbol{\gamma}_p, \boldsymbol{\lambda} \in \mathbb{R}_+^P \right\}.$$

The convex hull is given by

$$\text{conv}(\boldsymbol{\gamma}_p, 1 \leq p \leq P) \stackrel{\text{def.}}{=} \left\{ \sum_{p=1}^P \lambda[p] \boldsymbol{\gamma}_p, \boldsymbol{\lambda} \in S_{P-1} \right\}.$$

III. FORWARD MODELLING

In this paper, we assume that the degraded signal $\mathbf{y} \in \mathbb{R}^M$ is generated according to the following equation:

$$\mathbf{y} = \mathcal{P}(\mathbf{A}(\tilde{\boldsymbol{\gamma}})\bar{\mathbf{x}}), \quad (2)$$

where $\mathbf{A}(\boldsymbol{\gamma}) : \mathbb{R}^N \rightarrow \mathbb{R}^M$ is a linear operator depending on the unknown parameter $\tilde{\boldsymbol{\gamma}} \in \mathbb{R}^P$. The mapping $\mathcal{P} : \mathbb{R}^N \rightarrow \mathbb{R}^M$ can model various deterministic or stochastic perturbations occurring in real systems such as:

- *Additive white Gaussian noise*: $\mathcal{P}(\bar{\mathbf{y}}) = \bar{\mathbf{y}} + \mathbf{b}$ with $\mathbf{b} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_M)$.
- *Poisson noise*: In that case, $\mathbf{y} \in \mathbb{N}^M$ denotes a number of photons.
- *Quantization*: $\mathcal{P}(\bar{\mathbf{y}}) = \frac{1}{\Delta} \lfloor \mathbf{y} \cdot \Delta \rfloor$, where $\Delta > 0$ is the quantization step.
- *Clipping*: $\mathcal{P}(\bar{\mathbf{y}}) = \min(\bar{\mathbf{y}}, U)$, where $U > 0$ denotes a saturation parameter indicating the maximum available dynamics of the image.
- *Combinations*: the above perturbations can be combined to produce more accurate models.

In this paper, we will focus on the additive white Gaussian noise only, though all these models can be incorporated easily in the proposed framework at the learning stage. A critical aspect of this paper is the parameterization of the forward operator \mathbf{A} . We discuss this aspect below.

A. Linear parameterization

The simplest choice is probably to assume that \mathbf{A} belongs to a subspace of operators. This leads to the linear model below.

Assumption III.1 (Subspace of operators). *The forward operator in Problem (2) belongs to a known K -dimensional subspace of operators:*

$$\mathbf{A}(\boldsymbol{\gamma}) \stackrel{\text{def.}}{=} \sum_{k=1}^K \boldsymbol{\gamma}[k] \mathbf{A}_k, \quad (3)$$

with known elementary operators $\mathbf{A}_k \in \mathbb{R}^{M \times N}$.

Up to a Gram-Schmidt orthogonalization, we can always assume that the operators (\mathbf{A}_k) have unit norm and are pairwise orthogonal with respect to the Frobenius scalar product.

The main technical asset of the above model is the following property.

Proposition III.1. *Under Assumption III.1, we have*

$$\|\mathbf{A}(\boldsymbol{\gamma})\|_F = \|\boldsymbol{\gamma}\|_2 \text{ and } \|\mathbf{A}(\boldsymbol{\gamma}) - \mathbf{A}(\boldsymbol{\gamma}')\|_F = \|\boldsymbol{\gamma} - \boldsymbol{\gamma}'\|_2$$

for all $\boldsymbol{\gamma}, \boldsymbol{\gamma}' \in \mathbb{R}^P$

Proof.

$$\begin{aligned} \|\mathbf{A}(\boldsymbol{\gamma})\|_F^2 &= \left\langle \sum_{k=1}^K \boldsymbol{\gamma}[k] \mathbf{A}_k, \sum_{k=1}^K \boldsymbol{\gamma}[k] \mathbf{A}_k \right\rangle \\ &= \sum_{k=1}^K \sum_{k'=1}^K \boldsymbol{\gamma}[k] \boldsymbol{\gamma}[k'] \langle \mathbf{A}_k, \mathbf{A}_{k'} \rangle = \sum_{k=1}^K \boldsymbol{\gamma}[k]^2, \end{aligned}$$

by orthogonality of the family (\mathbf{A}_k) . The second identity results from the linearity of the parameterization. \square

To further improve the identifiability of the problem, we will sometimes assume the following:

Assumption III.2. *The parameter $\boldsymbol{\gamma}$ belongs to a known convex cone $\Gamma \subset \mathbb{R}^K$, defined with P extreme rays:*

$$\Gamma = \text{cone}(\boldsymbol{\gamma}_p, 1 \leq p \leq P). \quad (4)$$

Let us now discuss how these assumptions can be used in practice for real imaging systems.

1) *Convolution models and eigen-PSF bases*: By far the most wide-spread blurring model in imaging is based on convolution operators: the point spread function is identical whatever the position in space. This model is accurate for small fields of view, which are widespread in applications. Assuming that there is no sub-sampling in the model, we can set $M = N$ and $\mathbf{A}\mathbf{x} = \mathbf{h} \star \mathbf{x}$ for some unknown convolution kernel \mathbf{h} .

The convolution model strongly simplifies the blur identification problem since we are now looking for a vector of size N instead of a huge $N \times N$ matrix. Yet, the blind deconvolution problem is known to suffer from many degeneracies and possesses a huge number of possible solutions, see e.g. [5]. To further restrict the space of admissible operators and therefore improve the identifiability, we can expand the kernel \mathbf{h} in an *eigen-PSF basis*. This leads to the following low dimensional model.

Model III.1 (Convolution and eigen-PSFs). *We assume that*

$$\mathbf{A}(\boldsymbol{\gamma}) = \sum_{k=1}^K \gamma[k] \mathbf{e}_k \star \mathbf{x},$$

where (\mathbf{e}_k) is an orthogonal family of convolution kernels coined *eigen-PSF basis*. This amounts to setting

$$\mathbf{A}_k \mathbf{x} = \mathbf{e}_k \star \mathbf{x}$$

in the frame of Assumption III.1.

In the above model the elementary operators (\mathbf{A}_k) are pairwise orthogonal since the basis (\mathbf{e}_k) is orthogonal. Defining an eigen-PSF basis can be achieved by computing a principal component analysis of a family of observed or theoretical point spread functions [53]. The authors recently developed the Fiji plugin Eigen-PSF extractor to this end. An example of experimental eigen-PSF basis is shown on Figure 2, top.

Let $(\mathbf{h}_p)_{1 \leq p \leq P}$ define a set of P point spread functions in \mathbb{R}^N . They can be decomposed over the basis (\mathbf{e}_k) , and represented by vectors $\boldsymbol{\gamma}_p$ of size K . The set Γ of admissible parameterization can then be defined as the conical hull generated by the projection coefficients $(\boldsymbol{\gamma}_p)$ as in (4). This method nicely captures some physical properties of the point spread functions. For instance, non-negativity and smoothness are preserved by taking convex combinations. Hence, computing a conical hull is a natural way to learn models of impulse responses and interpolate between them.

2) *Space variant models and product-convolution expansions*: The convolution model III.1 can only capture *space invariant* impulse responses. When dealing with large field of views, this model becomes inaccurate. One way to overcome this limitation is to use product-convolution expansions [54], [55], [47], which efficiently encode space-varying systems.

Model III.2 (Product-convolution expansions). *Let $(\mathbf{e}_i)_{1 \leq i \leq I}$ and $(\mathbf{f}_j)_{1 \leq j \leq J}$ define two orthogonal families of \mathbb{R}^N . For $k = (i, j)$, we define the k -th elementary operator \mathbf{A}_k by*

$$\mathbf{A}_k \mathbf{x} = \mathbf{e}_i * (\mathbf{f}_j \odot \mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^N, \quad (5)$$

where \odot indicates the coordinate-wise (Hadamard) product.

In the above model, the basis (\mathbf{e}_i) can still be interpreted as an eigen-PSF basis. Indeed, we have for all locations $z \in \{1, \dots, N\}$:

$$\mathbf{A}(\boldsymbol{\gamma}) \boldsymbol{\delta}_z = \sum_{i=1}^I \left(\sum_{j=1}^J \gamma[i, j] \mathbf{f}_j[z] \right) \mathbf{e}_i[\cdot - z].$$

Hence, we see that each impulse response is expressed in the basis (\mathbf{e}_i) . The basis (\mathbf{f}_j) on its side, can be interpreted as an eigen-space variation basis: it describes how the point spread functions can vary in space. In optical devices such as microscopes, the estimation of the families (\mathbf{e}_i) and (\mathbf{f}_j) can be accomplished by observing several images of micro-beads [56], [47].

An example of experimental product-convolution family is shown in Figure 2 for a wide-field microscope. In that case, the dimension K of the subspace is $K = I \cdot J = 16$. Again, taking the conical hull of observed decompositions is an efficient way to restrict the family.

B. Nonlinear parameterization and Zernike polynomials

In the previous paragraph, we considered linear parameterizations which have many assets. They are easy to implement and can lead to convex models if one variable ($\bar{\gamma}$ or $\bar{\mathbf{x}}$) has to be estimated while the other is fixed. In addition, they are quite natural models that can be efficiently estimated from experimental data.

However, noise on the data may lead to inaccuracies making it impossible to capture some fine features of the point spread functions. An alternative is to use theoretical ones, leading to the nonlinear world. A popular and effective model in microscopy and astronomy consists in using the Fresnel approximation and to approximate the pupil function with a finite number of Zernike polynomials [57], [18]. This model leads to some of the state-of-the-art algorithms for blind deconvolution and super-resolution in microscopy and astronomy [21], [20], [58], [59].

Model III.3 (Fresnel approximation and a Zernike basis). *We assume that the forward model is a convolution with a continuous 3D kernel $h(x, y, z)$. The 3D kernel can be expressed through the 2D pupil function ϕ as:*

$$h(x, y, z) = \left| \int \phi(w_1, w_2) \exp(2i\pi z d(w_1, w_2)) \exp(2i\pi(w_1 x + w_2 y)) dw_1 dw_2 \right|^2$$

with

$$d(w_1, w_2) = \sqrt{(n/\lambda)^2 - (w_1 + w_2)^2},$$

where n is the refractive index of the immersion medium and λ is the wavelength of the observation light.

The complex pupil function ϕ can be expanded with Zernike polynomials Z_k :

$$\phi = \exp \left(2i \sum_{k=1}^K \gamma[k] Z_k \right),$$

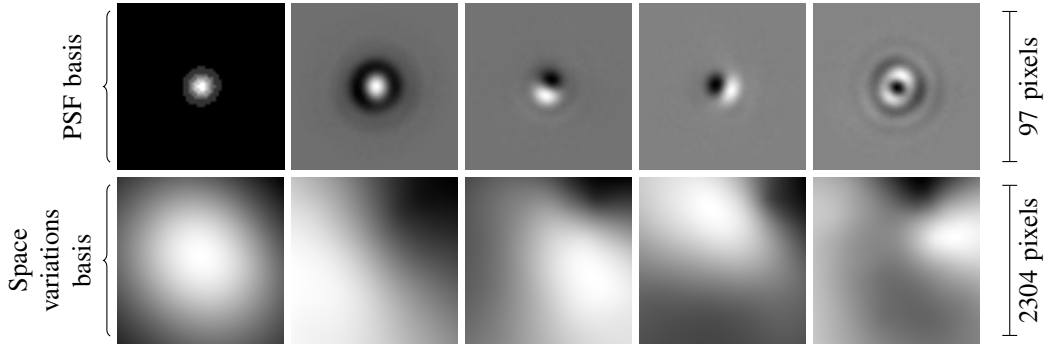


Fig. 2: Examples of eigen-PSF and eigen-space variation bases for a wide-field microscope [47].

where the coefficients $\gamma[k] \in \mathbb{R}$ are real numbers ².

A few examples of slices of point spread functions generated with Model III.3 are displayed in Figure 4. In that case, we used $K = 7$ Zernike polynomials and set the coefficients γ as uniform random variables with amplitude less than 0.15. As can be seen, a rich variety of impulse responses can be generated with this low dimensional model.

C. The limits of a low-dimensional parameterization

The above models are sufficiently rich to describe most optical devices accurately. In microscopy, they can capture defocus, refractive index mismatches, changes of temperature, tilts of optical components, usual optical aberrations with a parameter dimension K smaller than 20.

Notice however that some phenomena can hardly be modelled by low-dimensional parameterizations. In microscopy for instance, diffraction by the sample itself can lead to extremely complicated and diverse forward models better described by nonlinear equations, see *e.g.*, [60]. Similarly, in computer vision, motion and defocus blurs can vary abruptly with the movement and depth of the objects. The resulting operators would likely require a large number of parameters, which is out of the scope of this paper.

Overall, we see that the proposed contribution is well adapted to the fine characterization of optical systems, but probably do not extend easily to variations induced by the observed objects themselves.

IV. THE PROPOSED METHODOLOGY

In this section, we discuss the core of our contribution: we first detail the neural network architecture and then explain how to train it.

A. The Deep-Blur architecture

We propose to train two different neural networks sequentially:

- The first one denoted IN is an *identification network*. It depends on weights θ . The mapping $\text{IN}(\theta) : \mathbb{R}^M \rightarrow \mathbb{R}^K$

²In this model, only the phase of the pupil function varies. In all generality, the amplitude could vary as well at a slower rate. Most models in the literature assume a constant amplitude.

takes as an input a degraded image $\mathbf{y} \in \mathbb{R}^M$ and provides an estimate $\hat{\gamma}$ of $\tilde{\gamma}$ in \mathbb{R}^K .

- The second one denoted DN is a *deblurring network*. It depends on weights ξ . The mapping $\text{DN}(\xi) : (\mathbb{R}^M, \mathbb{R}^K) \rightarrow \mathbb{R}^N$ takes as input parameters the blurry image \mathbf{y} and an operator parameterization γ . It outputs an estimate $\hat{\mathbf{x}}$ of the sharp image $\bar{\mathbf{x}}$.

1) *The identification network*: Traditional estimation of a blur kernel relies on the detection of cues in the image such as points (direct observation), edges in different orientations (Radon transform of the kernel [9]) or textures (power spectrum) followed by adapted inversion procedures. This whole process can be modelled by a set of linear operations (filtering) and nonlinear operations (*e.g.*, thresholding). A convolutional neural network, composed of similar operations, should therefore be expressive enough to estimate the blur parameters. This is the case for the Deep-Blur identification architecture, a ResNet encoder [42] shown on Fig. 1, left. It consists of a succession of convolutions, ReLU activation, batch normalization and average pooling layers, allowing to sequentially reduce the image dimensions. The last layer is an adaptive average pooling layer, mapping the output of the penultimate layer to a vector of size K . The weights θ to be trained are the convolution kernels in the convolution layers and the biases. The encoder structure has been proven to be particularly effective for a large panel of signal processing tasks. It is a basic block of the U-Net architecture [61]. In our experiments, the vector θ has size $|\theta| = 11,178,448$.

2) *The deblurring network*: The proposed deblurring network mimics a Douglas-Rachford algorithm [62]. It is sometimes called an unrolled or unfolded network. This type of network currently achieves near state-of-the-art performance for a wide range of inverse problems (see *e.g.*, [45]). It has the advantages of having a natural interpretation as an approximate solution of a variational problem and naturally adapts to changes of the observation operators.

a) *Deep unrolling*: For $\lambda > 0$, let $\mathbf{R}_{\gamma,\lambda}$ denote the following regularized inverse:

$$\mathbf{R}_{\gamma,\lambda} = (\mathbf{A}(\gamma)^T \mathbf{A}(\gamma) + \lambda \mathbf{I})^{-1} \mathbf{A}(\gamma)^T.$$

For a parameter γ describing the forward operator and an input image \mathbf{y} , the Douglas-Rachford algorithm can be described

by the following sequence of operations, ran from $t = 0$ to $t = T - 1$ with $T \in \mathbb{N}$.

Algorithm 1 The Douglas-Rachford deblurring network DN

Require: iteration number $T \in \mathbb{N}$, operator γ , scale $\lambda \in \mathbb{R}_+$

$$\mathbf{z}_0 = \mathbf{R}_{\gamma, \lambda}(\mathbf{y})$$

for all $t = 0 \rightarrow T - 1$ **do**

$$\mathbf{x}_{t+1} = \text{PN}_t(\mathbf{z}_t)$$

$$\mathbf{z}_{t+1} = \mathbf{z}_t + \mathbf{R}_{\gamma, \lambda}(2\mathbf{x}_t - \mathbf{z}_t) - \mathbf{x}_t$$

end for

The initial guess \mathbf{z}_0 corresponds to the solution of

$$\mathbf{z}_0 = \underset{\mathbf{z} \in \mathbb{R}^N}{\text{argmin}} \frac{1}{2} \|\mathbf{A}(\gamma)\mathbf{z} - \mathbf{y}\|_2^2 + \frac{\lambda}{2} \|\mathbf{z}\|_2^2.$$

It can be evaluated approximately with a conjugate gradient algorithm run for a few iterations (10 in our implementation).

The mapping $\text{PN}_t(\xi_t) : \mathbb{R}^N \rightarrow \mathbb{R}^N$ can be interpreted as a ‘‘proximal neural network’’. Proximal operators [62] have been used massively in the last twenty years to regularize inverse problems. A popular example is the soft-thresholding operator, which is known to promote sparse solutions. Here, we propose to learn the regularizer as a neural network denoted PN_t , which may change from one iteration to the next. It corresponds to the green layers in Fig.1.

The parameters ξ that are learned, are the weights ξ_t defining the t -th proximal neural network PN_t . In our experiments, the networks PN_t have the same architecture for all $1 \leq t \leq T$. We used the current state-of-the-art network used in plug-and-play algorithms called DRUNET [52], [63]. We set $T = 4$ iterations. Each of the 4 proximal networks contain 8, 159, 808 parameters, resulting in a total of $|\xi| = 32, 639, 232$ parameters to be trained.

b) Differences with plug&play and deep unrolling:

The proposed architecture follows closely the usual unrolled algorithms [43], [44], [45], [64], [38]. There is however a major difference: traditionally, these unrolled architectures are trained to invert a *single operator*. In this paper, we train the network with a *family* of operators. We recently showed in [65] that this approach only results in marginal performance loss for a given operator if the family is sufficiently small, while providing a massive improvement in adaptivity to all the operators. In a sense, the proposed approach can be seen as an intermediate step between the plug&play priors which are designed to adapt to all possible operators, and the traditional unrolled algorithm adapted to a single one.

B. Designing the objective functions

Now that the neural network architectures have been described, let us turn to the description of the training procedure for the weights θ and ξ .

We propose to first train the identification network $\text{IN}(\theta)$ alone and then train the deblurring network $\text{DN}(\xi)$ with the output of the identification network as an input parameter. This sequential approach presents two advantages:

- First, the memory consumption is lower. The automatic differentiation only needs to store the parameters of the

individual networks, instead of both. This significantly reduces the memory footprint.

- Second, the identification network can be used independently of the other and it is therefore tempting to train it separately. In metrology applications for instance, where the aim is to follow the state of an optical system through time, the identification network IN is the most relevant brick. In some applications, such as super-resolution from single molecules, the deblurring network could be replaced by a more standard total variation based solver [66], once the operator is estimated.

In what follows, we let $\mathcal{X} \subset \mathbb{R}^N$ denote a dataset of admissible images/signals and $\mathcal{L}_{\mathcal{X}}$ denote a sampling distribution over \mathcal{X} . We let \mathcal{L}_{Γ} denote a sampling distribution supported on the set $\Gamma \subset \mathbb{R}^K$. In our experiments, the perturbation \mathcal{P} in Equation (2) is assumed to be additive white Gaussian noise $\mathcal{N}(0, \sigma^2 \mathbf{I}_M)$ with a fixed noise amplitude σ^2 .

1) *Training the identification network:* Motivated by Proposition III.1, the estimation error for the parameter γ can be measured by the ℓ^2 -norm in \mathbb{R}^K . This leads to the following empirical risk minimization:

$$\inf_{\theta \in \mathbb{R}^{|\theta|}} \mathbb{E}_{\substack{\mathbf{x} \sim \mathcal{L}_{\mathcal{X}} \\ \gamma \sim \mathcal{L}_{\Gamma} \\ \mathbf{b} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_M)}} \left[\frac{1}{2} \|\text{IN}(\theta)(\mathbf{A}(\gamma)\mathbf{x} + \mathbf{b}) - \gamma\|_2^2 \right]. \quad (6)$$

Problem (6) can be attacked using stochastic gradient descents type algorithms. In our experiments, we used the Adam optimizer [67] with the default parameters.

2) *Training the deblurring network:* Once the identification network IN is trained, we turn to the deblurring network. Similarly, we propose to minimize the following empirical risk:

$$\inf_{\xi \in \mathbb{R}^{|\xi|}} \mathbb{E}_{\substack{\mathbf{x} \sim \mathcal{L}_{\mathcal{X}} \\ \gamma \sim \mathcal{L}_{\Gamma} \\ \mathbf{b} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_M)}} \left[\frac{1}{2} \|\text{DN}(\xi)(\mathbf{y}, \hat{\gamma}) - \mathbf{x}\|_2^2 \right], \quad (7)$$

where $\mathbf{y} = \mathbf{A}(\gamma)\mathbf{x} + \mathbf{b}$ is the degraded image and $\hat{\gamma} = \text{IN}(\theta)(\mathbf{y})$ is the estimated parameter. Of importance, notice that we do not plug the true parameter γ in 7, but rather the estimated one $\hat{\gamma}$. This way, the deblurring network DN can learn to correct model mismatches that may occur at the estimation step.

3) *Sampling:* Applying the stochastic gradient machinery requires to design the distributions $\mathcal{L}_{\mathcal{X}}$ and \mathcal{L}_{Γ} , *i.e.*, to find a way to draw the images \mathbf{x} and the parameters γ .

For $\mathcal{L}_{\mathcal{X}}$, we simply use a uniform distribution over a large database of images. Another option could be to simulate the images according to realistic processes for specific applications such as single molecule localization.

C. Sampling a compact set of operators

A nonconventional aspect of this work is the need for a sampling distribution over the parameters in Γ . The easiest way to sample operators is to proceed as with images: take them uniformly at random among a discrete set of observations. Unfortunately, this approach is not always recommendable or even feasible. First, the distribution of observations might be

uneven, leading to bad recovery performance for rare operators. In addition, estimating an operator from experimental data can be complicated and time consuming.

To overcome these issues, we first propose to interpolate a few observed operators, to construct a richer set. We then sample the resulting set as evenly as possible with a strategy described hereafter. An example of set was justified in paragraph III-A1 with the convex hull of observations. Alternatives might exist, such as using generative models [68] to describe the space of admissible parameters. We do not explore this possibility in this paper.

a) *MaxiMin and MaxiSumMin*: In what follows, we assume that γ lives in a compact set $\Gamma \subset \mathbb{R}^K$. Our objective is to cover the set Γ as uniformly as possible with a fixed budget of $L \in \mathbb{N}$ points. This problem is tightly related to the sphere packing problem and is known to be extremely hard to solve exactly (see *e.g.*, [69], or the 7-th problem of Smale’s list for the XXIst century). One possible way to express it is the maximin formulation [70], [71]:

$$\sup_{\Delta=(\delta_1, \dots, \delta_L) \in \Gamma^L} \min_{l' \neq l} \|\delta_l - \delta_{l'}\|_2. \quad (8)$$

The idea is to spread the points as much as possible while keeping them within the constraint set Γ . The cost function is nonconvex and non-differentiable. It typically possesses a huge number of critical points. As far as we know, there is currently no available software to solve this problem – even approximately – for the dimensions of our problem [71]. In what follows, we propose an original optimization approach scaling well to dimensions of the order $D > 100$ and a number of sampling points $L > 10000$.

Problem (8) could possibly be solved with a projected subgradient approach. Unfortunately, the convergence would be extremely slow: only the two closest points in the set Δ would move at each time step. To accelerate the convergence, we propose to solve instead:

$$\sup_{\Delta=(\delta_1, \dots, \delta_L) \in \Gamma^L} F(\Delta) \stackrel{\text{def.}}{=} \sum_{l=1}^L \min_{l' \neq l} \|\delta_l - \delta_{l'}\|_2. \quad (9)$$

The interest of solving (9) instead of (8) can be seen when looking at the expression of the subgradient:

$$\partial_l F(\Delta) = \frac{\delta_l - \delta_{l^*(l)}}{\|\delta_l - \delta_{l^*(l)}\|_2}, \quad (10)$$

where $l^*(l)$ is the index of the point closest to δ_l . The sign $=$ can be replaced by \ni to obtain an element of the Clarke subdifferential. Assuming that a projector Π_Γ onto the set Γ is available, we can then attack (9) with as subgradient algorithm:

$$\Delta_{t+1} = \Pi_{\Gamma^L}(\Delta_t + \tau_t \partial F(\Delta_t)), \quad (11)$$

where (τ_t) is a sequence of step-sizes and $\partial F(\Delta_t)$ is the vector with coordinates given by (10). Contrarily to (8), we see that at each iteration, all the points will move simultaneously in a direction opposite to their closest neighbour.

The usual theories [72] for subgradient descents recommend a decaying step-size such as $\tau_t = \frac{c}{\sqrt{t}}$. In this work, we experimented various choices (*e.g.*, Barzilai-Borwein) and

finally opted for a constant step-size for this nonconvex, non differentiable function. Such a choice does not lead to a convergent sequence (Δ_t) in general. It typically finishes oscillating around a local maximizer of the energy with a radius that depends on the step-size. This is sufficient for our application since we - in any cases - end up close to a local maximizer only and do not require a precise solution.

b) *Handling convex hulls*: In the specific case where $\Gamma = \text{conv}(\gamma_p, 1 \leq p \leq P)$, we have $\Gamma = \Gamma S_{P-1}$, where $\Gamma = [\gamma_1, \dots, \gamma_P]$ is the matrix formed by concatenating all the vectors γ_p . Computing the orthogonal projection on Γ is non-trivial. To solve this issue, we propose to replace (9) by

$$\sup_{\Lambda=(\lambda_1, \dots, \lambda_L) \in S_{P-1}^L} F(\Gamma \Lambda), \quad (12)$$

and use a subgradient descent as well:

$$\Delta_{k+1} = \Pi_{S_{P-1}^L}(\Delta_k + \tau_k \Gamma^T \partial F(\Gamma \Delta_k)). \quad (13)$$

The projection on the simplex $\Pi_{S_{P-1}}$ can be evaluated exactly in $O(P)$ operations using dynamic programming [73]. The algorithm is stopped when the minimal distance between all pairs of points do not vary significantly after 10 iterations.

Letting $\hat{\Lambda} = (\hat{\lambda}_1, \dots, \hat{\lambda}_L)$ denote the last iterate of the algorithm, the points δ_l in the previous formulation are given by $\delta_l = \Gamma \hat{\lambda}_l$.

c) *Evaluating the subgradient*: The most complicated part of the code is to find the closest neighbor of each point δ_l . A naive implementation would result in a complexity $O(L^2)$, which is not viable for large L (say larger than 10,000). Hopefully, finding nearest neighbors efficiently is a problem of high importance and a very efficient implementation was proposed in [74]. The authors propose a CPU/GPU implementation returning approximate or exact nearest neighbors scaling up to hundreds of billion vectors in high dimension.

d) *Examples*: To conclude this section, we provide a simple example of the algorithm dynamics in Fig. 3 in 2D.

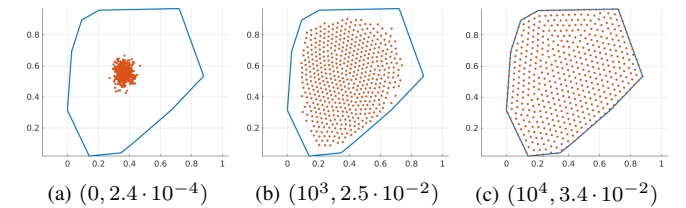


Fig. 3: Dynamics of the maxisummin algorithm for $L = 500$ points. The pair of numbers below the figures are the iterate number and the distance between the closest pair of points in the design. As can be seen, after 10,000 iterations, the points are near perfectly spread in the convex hull of 8 points. Here, we set $\tau_t = 10^{-3}$.

V. NUMERICAL EXPERIMENTS

In this section, we illustrate the different ideas proposed in this paper. In all our experiments, we trained the neural networks using the MS Coco dataset [75]. It contains 118,287 images in the training set and 40,670 images in the test set.

A. The interest of operator space sampling

We illustrate the importance of carefully sampling the set of admissible operators using the method described in Section IV-C. To this end, we use the blur operator generated by Model III.2. The functions $(\mathbf{e}_i)_{1 \leq i \leq I}$ and $(\mathbf{f}_j)_{1 \leq j \leq J}$ are generated from a real wide-field fluorescence microscope. The dimension of the subspace is $K = 4 \times 4$ and the cone Γ is defined by 400 operators. Details about this construction can be found in [47]. We use Equation (13) to generate a set of 10,000 well spread admissible operators. The coefficients γ are then drawn uniformly at random among this set. We also add Gaussian noise with amplitude $\sigma = 0.01$.

We first train the identification network IN with operators among the discretization of the cone Γ containing 10,000 points. We train a second network with a more naive approach where IN is trained using only the 400 blur operators defining the set Γ . After training over 150,000 iterations with batches of 16 images, we evaluate the results on 100 images and 100 operators randomly sampled from the discretization of Γ , bringing the total number of test samples to 10,000. We report the SNR between the true operators and the estimations by the network IN in Table I. Sampling from the discretization of the set Γ comes with almost no additional computational cost and allows to significantly increase the SNR of the estimated blur operators.

MaxiSumMin	Extreme points
27.6 ± 3	22.4 ± 4

TABLE I: Importance of sampling the cone Γ of admissible operators. SNR between the true operator and the one estimated by IN for two different set of operators used during training.

B. Convolution operators

In this section, we evaluate the accuracy of the identification and deblurring networks for convolution (i.e. space invariant) operators. We assess them for images generated with point spread functions expanded in Zernike polynomial.

1) *Identifying convolution operators*: In this paragraph, we assess the ability of a residual network to identify the point spread function generated by Model III.3. A similar study was carried out in [41] with $K = 3$ coefficients. Here, we extend the study to $K = 7$ coefficients allowing to represent the following aberrations in the Noll nomenclature [76]: defocus, primary astigmatism, primary coma, trefoil and primary spherical.

We generate random PSFs by drawing the coefficients $\gamma[k]$ (see Model III.3) uniformly in the range $[-\eta, \eta]$. The higher η , the more spread and oscillating the PSF. Hence η can be interpreted as a measure of PSF complexity. The model was trained for a value of $\eta = 0.15$.

Figure 4 shows the identification results for 3 images taken at random from the test set and 3 operators taken at random in the operator set. On these examples, the network provides faithful estimates despite an important noise level and images with little contents.

To further characterize the network efficiency, we measure the distribution of signal-to-noise-ratio (SNR) in the noiseless regime. For a kernel \mathbf{h} , the error of the estimated kernel $\hat{\mathbf{h}}$ is defined by:

$$\text{SNR}(\mathbf{h}, \hat{\mathbf{h}}) = -10 \log_{10} \left(\frac{\|\hat{\mathbf{h}} - \mathbf{h}\|_2^2}{\|\mathbf{h}\|_2^2} \right). \quad (14)$$

Fig. 5 summarizes the conclusions. In average, the identification network outputs estimates with a relative error below 5%.

Finally, we study the stability to the noise level σ in Fig. 6a – and to the PSF complexity η in Fig. 6b. As can be seen, the identification outputs predictions with less than 10% error with probability larger than 0.5 up to a large noise level of $\sigma = 0.1$. The dependency on the kernel’s complexity, measured through the Zernike polynomials amplitude η is very clear with typical errors below 2% for $\eta < 0.1$ and then a relatively fast increase. It is nonetheless remarkable that the identification returns estimates with less than 15% error for $\eta = 0.2$, which produces significantly more complex PSFs than those observed during the training phase, showing some ability of the network to extrapolate.

2) *Evaluating the deblurring network*: In this section, we evaluate the performance of the proposed deblurring network for convolution operators defined using the Fresnel approximation. Figures 7, 8, 9 display some deconvolution results for different methods. The corresponding image quality measures are displayed in Tables II, III, IV.

Notice that this problem is particularly involved: there is complete loss of information in the high frequencies since the convolution kernels are bandlimited and we treat different noise levels up to rather high values (here $\sigma = 0.05$ for images in the range $[0, 1]$). Despite this challenging setting, it can be seen both perceptually and from the SSIM (Structural Similarity Index Measure) that the image quality is significantly improved whatever the noise level. It is also remarkable to observe that the proposed network architecture allows to treat images with different noise levels. This is an important feature of the DRUNET used as a proximal network [52].

We also propose some comparisons with other methods from the literature. Whenever possible, we optimized the hyperparameters by hand for each noise level to produce the best possible output. We chose the following methods:

- The ℓ^0 -gradient prior [28], [82]. This method is one of the state-of-the-art handcrafted blind deblurring method. An efficient implementation was recently proposed in [77].
- In [27], the authors proposed a kernel estimation method based on the assumption that the image spectrum amplitude has a specific decaying distribution in the Fourier plane. The kernel estimation then boils down to a phase retrieval problem. An efficient implementation was recently proposed in [78].
- In [79], the authors propose a regularized version of the ℓ^0 -gradient prior allowing to treat blurred images containing higher noise levels. This method also implements the FFDNET [83] as a pre-processing step to remove the noise prior to the estimation.

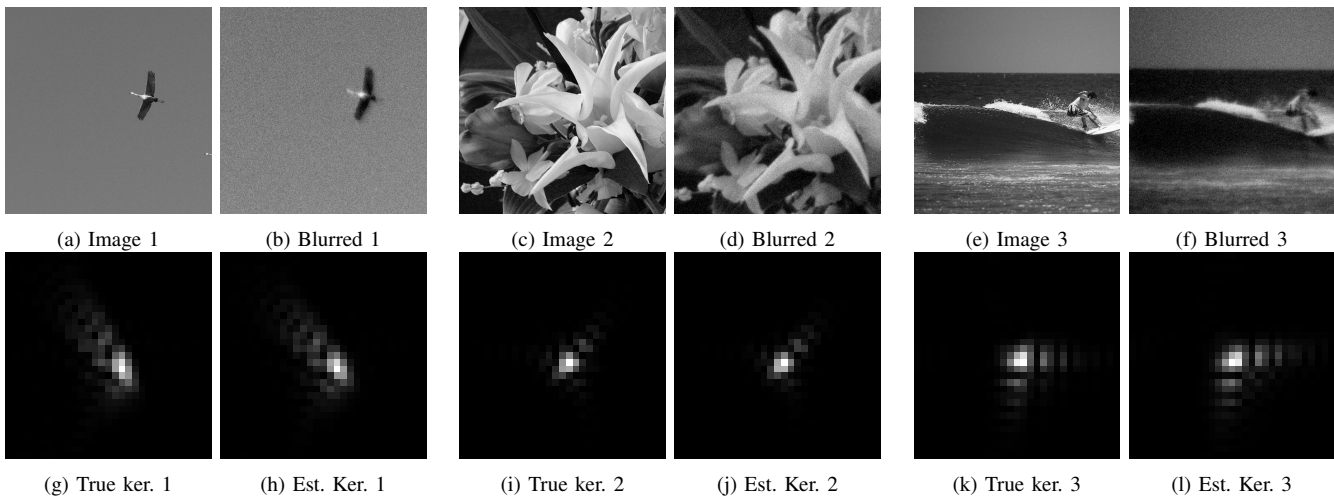


Fig. 4: Examples of results for the identification network with convolution kernels defined through Fresnel approximation. Top: the original and blurred and noisy 400x400 images. Bottom: the true 31x31 kernel used to generate the blurry image and the corresponding estimation by the neural network. Notice that there is a significant amount of white Gaussian noise added to the blurred image. The image boundaries have been discarded from the estimation process to prevent the neural network from using information that would not be present in real images.

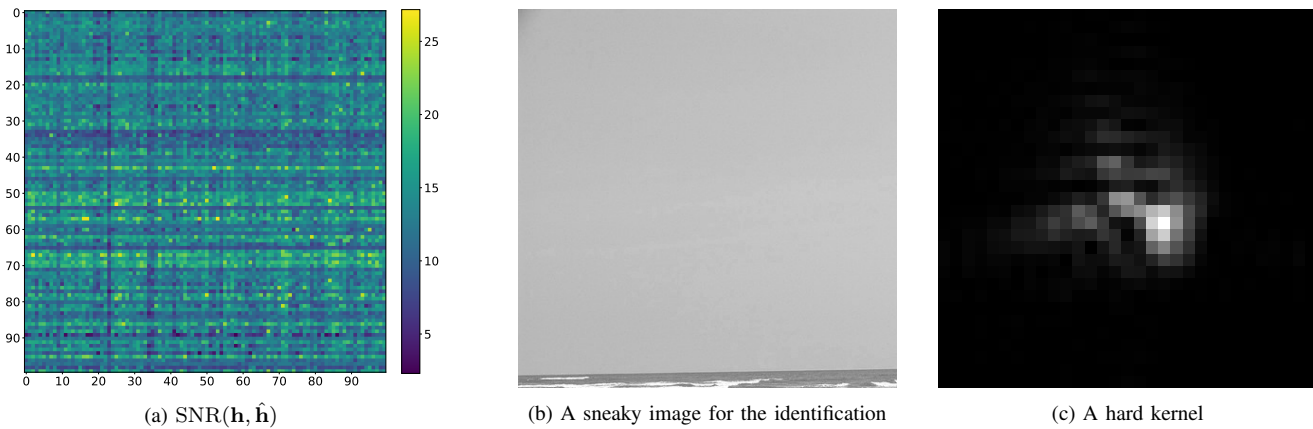


Fig. 5: On the left: a 100x100 table representing the SNR of the PSF. In this table we evaluated the identification network for 100 images (left to right) and 100 kernels (top to bottom) with no noise $\sigma = 0$. As can be seen, there are horizontal and vertical stripes. This means that some images and some kernels make the identification problem easier or harder. In the middle: an image making the identification problem hard (column 23). On the right: a kernel making the identification harder (row 65).

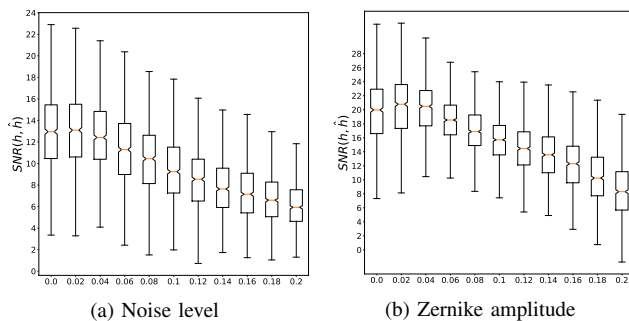


Fig. 6: Stability of the kernel estimation with respect to noise level (left) and amplitude of the Zernike coefficients in the noiseless regime (right).

- We also tested two state-of-the-art neural network approaches. The first one is the current leader (as of May 15, 2022) of the Go-Pro deblurring challenge called

NAFNET [80].

- The second one is the Deblur-Gan V2 [84], [81].

As can be seen from the tables and the perceptual results, our method (Deep-Blur) outperforms the other ones by a really significant margin. This is particularly true for the deep learning approaches, which are unable to improve the sharpness significantly and even fail completely for high noise levels. The comparisons are somewhat unfair since we test our method for the conditions it was trained for (diffraction blurs and grayscale images), while the two other deep learning methods were trained on motion blurs and color images. This probably illustrates the lack of robustness of neural network when using conditions escaping the ones used during the training approach. The handcrafted methods (Goldstein-Fattal and ℓ^0 gradient prior) significantly improve the image quality in the low noise regime, but fail to do so for high noise levels.

		im 1	im 2	im 3	im 4	im 5	im 6
SSIM \hat{x}	y	0.49	0.46	0.41	0.52	0.47	0.50
	ours	0.73	0.87	0.83	0.86	0.87	0.90
	[77]	0.51	0.52	0.56	0.58	0.58	0.55
	[78]	0.45	0.38	0.57	0.66	0.53	0.54
	[79]	0.57	0.46	0.44	0.54	0.51	0.48
	[80]	0.50	0.48	0.44	0.54	0.48	0.51
[81]	0.52	0.44	0.37	0.41	0.44	0.60	
SNR \hat{h}	ours	8.99	14.05	13.46	11.67	11.35	14.79
	[77]	-1.85	1.38	-0.46	-1.99	2.26	2.57
	[78]	0.21	1.78	0.49	-0.25	2.44	4.50
	[79]	0.27	3.45	0.33	-0.31	3.62	3.06

TABLE II: Reconstruction results in the noiseless setting $\sigma = 0$.

		im 1	im 2	im 3	im 4	im 5	im 6
SSIM \hat{x}	y	0.47	0.43	0.40	0.50	0.46	0.49
	ours	0.58	0.84	0.67	0.81	0.79	0.81
	[77]	0.53	0.63	0.53	0.52	0.53	0.64
	[78]	0.55	0.54	0.53	0.71	0.55	0.58
	[79]	0.52	0.52	0.43	0.54	0.50	0.48
	[80]	0.50	0.48	0.45	0.54	0.49	0.52
[81]	0.52	0.42	0.38	0.39	0.43	0.56	
SNR \hat{h}	ours	8.66	14.48	13.38	11.62	12.19	15.80
	[77]	-5.28	-0.25	-1.08	-2.61	1.83	2.60
	[78]	0.94	1.19	0.59	0.20	0.83	2.25
	[79]	-0.28	1.37	0.57	-0.36	3.57	2.85

TABLE III: Reconstruction results $\sigma = 10^{-2}$.

		im 1	im 2	im 3	im 4	im 5	im 6
SSIM \hat{x}	y	0.29	0.25	0.25	0.29	0.31	0.36
	ours	0.52	0.50	0.52	0.72	0.60	0.58
	[77]	0.47	0.48	0.41	0.49	0.53	0.39
	[78]	0.36	0.36	0.32	0.41	0.40	0.36
	[79]	0.49	0.53	0.38	0.56	0.53	0.47
	[80]	0.0	0.32	0.0	0.0	0.0	0.0
[81]	0.32	0.27	0.25	0.30	0.31	0.38	
SNR \hat{h}	ours	7.05	15.00	12.88	11.03	13.01	19.63
	[77]	-7.66	-4.85	-3.01	-6.81	-0.21	-3.50
	[78]	0.32	0.42	0.38	0.50	0.71	0.62
	[79]	-1.63	-0.53	0.94	-2.83	2.50	2.83

TABLE IV: Reconstruction results $\sigma = 5 \cdot 10^{-2}$.

a) *Training on the true or estimated operators?:* At training time, we can feed the unrolled deblurring network with the operator that was used to synthesize the blurry image, or the one estimated using the identification network. The potential advantage of the second option is to train the proximal networks correcting model mismatches. We tested both solutions on two different operator families. It turns out that they led to near undistinguishable results overall in average. The most likely explanation for this phenomenon is that the model mismatches produced by the identification network cannot be corrected with the proximal networks.

b) *The interest of training on a family:* The proposed deblurring algorithm can be seen as an intermediate step between the plug&play priors which are designed to solve arbitrary inverse problems and the traditional unrolled networks, which are designed for a specific operator. In this paragraph, we investigate the merits of these different approaches empirically. This study is closely related to the one in the preprint [65] for a different application.

To this end, we construct a plug&play deblurring algorithm using the Douglas-Rachford structure described in Algorithm 1. The proximal networks are all identical and correspond to the state-of-the-art DRUNET denoisers used in plug&play approaches [52]. It was trained specifically for denoising white Gaussian noise. The parameters of the algorithm are the number of iterations, the scaling parameter λ and a standard deviation for the noise level σ . Those parameters were tuned by hand to obtain the best possible signal-to-noise ratio. This optimum is attained with 8 iterations, $\lambda = 10^{-1}$ and $\sigma = 10^{-1}$.

We also trained our deblurring network DN for a single convolution operator with a Airy pattern. This pattern is obtained by setting $\gamma = 0$ in Model III.3.

We then test each deblurring network for 100 different pairs of images and operators and display the average gain and standard deviation in Table V.

All 3 deblurring networks depend on the operator parameterization γ . We evaluate the quality of the networks independently of the possible model mismatches and therefore feed all the networks with the true parameter used to synthesize the blurry images.

As can be seen in Table V, the plug&play approach takes significantly more time than the lighter deblurring networks we chose for this paper. The reason is twofold: first we use 4 iterations in our deblurring networks instead of 8 for the plug&play approach. Second, the DRUNET used in [85] contains significantly more parameters than the one used in our implementation. In both cases, memory issues justify this choice. We could likely increase the image quality of Deep-Blur further using more iterations and more parameters in the DRUNET, but this would come at a higher memory/energy cost.

Second, the deblurring network trained specifically on a Airy pattern outperforms the two other methods when tested on a Airy pattern, which is to be expected. However, the gain is anecdotic compared to the same deblurring network trained on a family of operators. In contrast, the deblurring network trained on a family of operators significantly outperforms the one trained on the Airy disk when tested on multiple operators. This study therefore confirms the findings in [65], where it was shown that the price of adaptivity to different operators was affordable. The plug&play approach on its side performs remarkably well for a “universal” method. Its main downsides are the computing times as well as a slight performance decrease.

C. Product-convolution operators

To finish the numerical experiments, we illustrate how the proposed ideas perform on product-convolution operators.

	Plug&Play	DN trained family	DN trained Airy
Airy kernel	0.29 ± 0.05	0.45 ± 0.09	0.46 ± 0.08
Random kernel	0.39 ± 0.1	0.43 ± 0.1	0.26 ± 0.05
Time per image	2.21"	0.25"	0.25"

TABLE V: Average SSIM gain for the different networks. The gain was evaluated on 100 random test images.

We first illustrate the performance of the identification network. We trained the identification network on natural images from the MS Coco dataset, but evaluate it on biological images from microscopes. We selected 6 images: *Image 1* is an histopathology of angiolipoma [86], *Image 2* is an histopathology of reactive gastropathy [87], *Image 3&4* are actin filaments within a cell [88], *Image 5* is an slice of a spheroid from [89] and *Image 6* is a crop of a podosome obtain on a wide-field microscope [90].

The blur operators are generated by Model III.2 using $K = 16$ parameters. This model was estimated using a real optical system [47]. Fig. 10 shows the identification results. The blur coefficients predicted by the Deep-Blur identification are accurate estimates in all cases. In average, the SNR is much higher than in the previous experiment, which can likely be explained by a smaller dimensionality of the operators' family. In all cases, the image quality is significantly improved despite an additive white Gaussian noise with $\sigma = 1 \cdot 10^{-2}$. This is remarkable since this type of images is really different from the typical computer vision images found in the MS COCO dataset.

VI. CONCLUSION

We proposed a specific neural network architecture to solve arbitrary blind inverse problems. We evaluated its performance carefully on blind deblurring problems with space invariant and space varying operators. A key assumption is to have access to a forward model that depends on a set of parameters. The network first estimates the unknown parameters describing the forward model from the measurements with a ResNet architecture. In a second step, an unrolled algorithm solves the inverse problem with a forward model that was estimated at the previous step. After designing a careful training procedure, we showed a significant advantage of the proposed approach in terms of robustness to noise levels and adaptivity to a vast family of operators and conditions not seen during the training phase. The deblurring network can be seen as an intermediate step between a plug&play algorithm which solves any inverse problems and the traditional unrolled algorithms which solve a specific one.

REFERENCES

- [1] Valentin Debarnot and Pierre Weiss. Deepblur: Blind identification of space variant psf. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 1544–1547. IEEE, 2021.
- [2] Antonin Chambolle and Thomas Pock. An introduction to continuous optimization for imaging. *Acta Numerica*, 25:161–319, 2016.
- [3] Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.
- [4] Alex von Diezmann, Maurice Y Lee, Matthew D Lew, and WE Moerner. Correcting field-dependent aberrations with nanoscale accuracy in three-dimensional single-molecule localization microscopy. *Optica*, 2(11):985–993, 2015.
- [5] Ferréol Soulez and Michael Unser. Superresolution with optically-motivated blind deconvolution. In *Laser Applications to Chemical, Security and Environmental Analysis*, pages JT3A–38. Optical Society of America, 2016.
- [6] Thomas G Stockham, Thomas M Cannon, and Robert B Ingebreten. Blind deconvolution through digital signal processing. *Proceedings of the IEEE*, 63(4):678–692, 1975.
- [7] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1):459–494, 2014.
- [8] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman. Understanding and evaluating blind deconvolution algorithms. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1964–1971. IEEE, 2009.
- [9] Felix Kraher, Youzuo Lin, Bonnie McAdoo, Katharine Ott, Jiakou Wang, David Widemann, and Brendt Wohlberg. Blind image deconvolution: Motion blur estimation. 2006.
- [10] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman. Removing camera shake from a single photograph. In *ACM SIGGRAPH 2006 Papers*, pages 787–794. 2006.
- [11] Shengyang Dai and Ying Wu. Motion from blur. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
- [12] Libin Sun, Sunghyun Cho, Jue Wang, and James Hays. Edge-based blur kernel estimation using patch priors. In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–8. IEEE, 2013.
- [13] Daniele Perrone and Paolo Favaro. Total variation blind deconvolution: The devil is in the details. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2909–2916, 2014.
- [14] Francoise Peyrin, Alina Toma, Bruno Sixou, Loïc Denis, Andrew Burghardt, and Jean-Baptiste Pialat. Semi-blind joint super-resolution/segmentation of 3d trabecular bone images by a tv box approach. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 2811–2815. IEEE, 2015.
- [15] Ayan Chakrabarti, Todd Zickler, and William T Freeman. Analyzing spatially-varying blur. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2512–2519. IEEE, 2010.
- [16] Florent Couzinie-Devy, Jian Sun, Karteek Alahari, and Jean Ponce. Learning to estimate and remove non-uniform image blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1075–1082, 2013.
- [17] Tony F Chan and Chiu-Kwong Wong. Total variation blind deconvolution. *IEEE transactions on Image Processing*, 7(3):370–375, 1998.
- [18] Joseph W Goodman. *Introduction to Fourier optics*. Roberts and Company Publishers, 2005.
- [19] Margret Keuper, Thorsten Schmidt, Maja Temerinac-Ott, Jan Padenken, Patrick Heun, Olaf Ronneberger, and Thomas Brox. Blind deconvolution of widefield fluorescence microscopic data by regularization of the optical transfer function (otf). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2179–2186, 2013.
- [20] Andrey Aristov, Benoit Lelandaïs, Elena Rensen, and Christophe Zimmer. Zola-3d allows flexible 3d localization microscopy over an adjustable axial range. *Nature communications*, 9(1):1–8, 2018.
- [21] Ferréol Soulez, Loïc Denis, Yves Tourneur, and Éric Thiébaud. Blind deconvolution of 3d data in wide field fluorescence microscopy. In *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 1735–1738. IEEE, 2012.
- [22] Rahul Mourya, Loïc Denis, Jean-Marie Becker, and Eric Thiébaud. A blind deblurring and image decomposition approach for astronomical image restoration. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 1636–1640. IEEE, 2015.
- [23] Valentin Debarnot and Pierre Weiss. Blind inverse problems with isolated spikes. *ArXiv*, 2020.
- [24] Leah Bar, Nir Sochen, and Nahum Kiryati. Semi-blind image restoration via mumford-shah regularization. *IEEE Transactions on Image Processing*, 15(2):483–493, 2006.
- [25] Praveen Pankajakshan, Bo Zhang, Laure Blanc-Féraud, Zvi Kam, Jean-Christophe Olivo-Marin, and Josiane Zerubia. Blind deconvolution for thin-layered confocal imaging. *Applied optics*, 48(22):4437–4448, 2009.
- [26] Ido Zachevsky and Yehoshua Y Zeevi. Blind deblurring of natural stochastic textures using an anisotropic fractal model and phase retrieval

- algorithm. *IEEE Transactions on Image Processing*, 28(2):937–951, 2018.
- [27] Amit Goldstein and Raanan Fattal. Blur-kernel estimation from spectral irregularities. In *European Conference on Computer Vision*, pages 622–635. Springer, 2012.
- [28] Jinshan Pan, Zhe Hu, Zhixun Su, and Ming-Hsuan Yang. Deblurring text images via 10-regularized intensity and gradient prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2908, 2014.
- [29] Christian J Schuler, Michael Hirsch, Stefan Harmeling, and Bernhard Schölkopf. Learning to deblur. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1439–1451, 2015.
- [30] Mehdi Noroozi, Paramanand Chandramouli, and Paolo Favaro. Motion deblurring in the wild. In *German conference on pattern recognition*, pages 65–77. Springer, 2017.
- [31] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3883–3891, 2017.
- [32] Raied Aljadaany, Dipan K Pal, and Marios Savvides. Douglas-rachford networks: Learning both the image prior and data fidelity terms for blind image deconvolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10235–10244, 2019.
- [33] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4641–4650, 2021.
- [34] Armin Mehri, Parichehr B Ardakani, and Angel D Sappa. Mprnet: Multi-path residual network for lightweight image super resolution. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2704–2713, 2021.
- [35] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 769–777, 2015.
- [36] Dong Gong, Jie Yang, Lingqiao Liu, Yanning Zhang, Ian Reid, Chunhua Shen, Anton Van Den Hengel, and Qinfeng Shi. From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2319–2328, 2017.
- [37] Ayan Chakrabarti. A neural approach to blind motion deblurring. In *European conference on computer vision*, pages 221–235. Springer, 2016.
- [38] Yuelong Li, Mohammad Tofighi, Junyi Geng, Vishal Monga, and Yonina C Eldar. Deep algorithm unrolling for blind image deblurring. *arXiv preprint arXiv:1902.03493*, 2019.
- [39] Leonhard Möckl, Petar N Petrov, and WE Moerner. Accurate phase retrieval of complex 3d point spread functions with deep residual neural networks. *Applied physics letters*, 115(25):251106, 2019.
- [40] Adrian Shajkofci and Michael Lieblich. Spatially-variant cnn-based point spread function estimation for blind deconvolution and depth estimation in optical microscopy. *IEEE Transactions on Image Processing*, 29:5848–5861, 2020.
- [41] Adrian Shajkofci and Michael Lieblich. Deepfocus: a few-shot microscope slide auto-focus using a sample invariant cnn-based sharpness function. *arXiv preprint arXiv:2001.00667*, 2020.
- [42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [43] Jonas Adler and Ozan Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, 2017.
- [44] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.
- [45] Vishal Monga, Yuelong Li, and Yonina C Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44, 2021.
- [46] Valentin Debarnot, Paul Escande, and Pierre Weiss. A scalable estimator of sets of integral operators. *Inverse Problems*, 2019.
- [47] Valentin Debarnot, Paul Escande, Thomas Mangeat, and Pierre Weiss. Learning low-dimensional models of microscopes. *IEEE Transactions on Computational Imaging*, 2020.
- [48] Vegard Antun, Francesco Renna, Clarice Poon, Ben Adcock, and Anders C Hansen. On instabilities of deep learning in image reconstruction and the potential costs of ai. *Proceedings of the National Academy of Sciences*, 117(48):30088–30095, 2020.
- [49] Martin Genzel, Jan Macdonald, and Maximilian Marz. Solving inverse problems with deep neural networks-robustness included. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [50] Gregory Ongie, Ajil Jalal, Christopher A Metzler, Richard G Baraniuk, Alexandros G Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020.
- [51] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948. IEEE, 2013.
- [52] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [53] Sarah Frisken Gibson and Frederick Lanni. Diffraction by a circular aperture as a model for three-dimensional optical microscopy. *JOSA A*, 6(9):1357–1367, 1989.
- [54] Loïc Denis, Eric Thiébaud, Ferréol Soulez, Jean-Marie Becker, and Rahul Mourya. Fast approximations of shift-variant blur. *International Journal of Computer Vision*, 115(3):253–278, 2015.
- [55] Paul Escande and Pierre Weiss. Approximation of integral operators using product-convolution expansions. *Journal of Mathematical Imaging and Vision*, 58(3):333–348, 2017.
- [56] Jérémie Bigot, Paul Escande, and Pierre Weiss. Estimation of linear operators from scattered impulse responses. *Applied and Computational Harmonic Analysis*, 47(3):730–758, 2019.
- [57] Bridget M Hanser, Mats GL Gustafsson, DA Agard, and John W Sedat. Phase-retrieved pupil functions in wide-field fluorescence microscopy. *Journal of microscopy*, 216(1):32–48, 2004.
- [58] Daniel Sage, Lauréne Donati, Ferréol Soulez, Denis Fortun, Guillaume Schmit, Arne Seitz, Romain Guiet, Cédric Vonesch, and Michael Unser. Deconvolutionlab2: An open-source software for deconvolution microscopy. *Methods*, 115:28–41, 2017.
- [59] Elias Nehme, Daniel Freedman, Racheli Gordon, Boris Ferdman, Lucien E Weiss, Onit Alalouf, Tal Naor, Reut Orange, Tomer Michaeli, and Yoav Shechtman. Deepstorm3d: dense 3d localization microscopy and psf design by deep learning. *Nature methods*, 17(7):734–740, 2020.
- [60] Thanh-an Pham, Emmanuel Soubies, Ahmed Ayoub, Joowon Lim, Demetri Psaltis, and Michael Unser. Three-dimensional optical diffraction tomography with lippmann-schwinger model. *IEEE Transactions on Computational Imaging*, 6:727–738, 2020.
- [61] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [62] Patrick L Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.
- [63] Samuel Hurault, Arthur Leclaire, and Nicolas Papadakis. Gradient Step Denoiser for convergent Plug-and-Play. In *International Conference on Learning Representations (ICLR’22)*, International Conference on Learning Representations, Online, United States, April 2022.
- [64] Yuelong Li, Mohammad Tofighi, Vishal Monga, and Yonina C Eldar. An algorithm unrolling approach to deep image deblurring. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7675–7679. IEEE, 2019.
- [65] Alban Gossard and Pierre Weiss. Training adaptive reconstruction networks for inverse problems. *arXiv preprint arXiv:2202.11342*, 2022.
- [66] Kristian Bredies and Hanna Katriina Pikkarainen. Inverse problems in spaces of measures. *ESAIM: Control, Optimisation and Calculus of Variations*, 19(1):190–218, 2013.
- [67] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [68] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2737–2746, 2020.
- [69] John Horton Conway and Neil James Alexander Sloane. *Sphere packings, lattices and groups*, volume 290. Springer Science & Business Media, 2013.
- [70] Mark E Johnson, Leslie M Moore, and Donald Ylvisaker. Minimax and maximin distance designs. *Journal of statistical planning and inference*, 26(2):131–148, 1990.
- [71] Luc Pronzato. Minimax and maximin space-filling designs: some properties and methods for construction. *Journal de la Société Française de Statistique*, 158(1):7–36, 2017.

- [72] Boris T Polyak. Introduction to optimization. optimization software. Inc., *Publications Division, New York*, 1, 1987.
- [73] Laurent Condat. Fast projection onto the simplex and the l_1 ball. *Mathematical Programming*, 158(1):575–585, 2016.
- [74] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.
- [75] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [76] Robert J Noll. Zernike polynomials and atmospheric turbulence. *JOsA*, 66(3):207–211, 1976.
- [77] Jérémy Anger, Gabriele Facciolo, and Mauricio Delbracio. Blind Image Deblurring using the l_0 Gradient Prior. *Image Processing On Line*, 9:124–142, 2019. <https://doi.org/10.5201/ipol.2019.243>.
- [78] Jérémy Anger, Gabriele Facciolo, and Mauricio Delbracio. Estimating an Image’s Blur Kernel Using Natural Image Statistics, and Deblurring it: An Analysis of the Goldstein-Fattal Method. *Image Processing On Line*, 8:282–304, 2018. <https://doi.org/10.5201/ipol.2018.211>.
- [79] Jérémy Anger, Mauricio Delbracio, and Gabriele Facciolo. Efficient blind deblurring under high noise levels. In *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 123–128. IEEE, 2019.
- [80] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. *arXiv preprint arXiv:2204.04676*, 2022.
- [81] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8878–8887, 2019.
- [82] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1628–1636, 2016.
- [83] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018.
- [84] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8183–8192, 2018.
- [85] Bo Zhang, Josiane Zerubia, and Jean-Christophe Olivo-Marin. Gaussian approximations of fluorescence microscope point-spread function models. *Applied optics*, 46(10):1819–1829, 2007.
- [86] Wikimedia Commons. File:histopathology of angioliopoma.jpg — wikimedia commons, the free media repository, 2020. [Online; accessed 9-May-2022].
- [87] Wikimedia Commons. File:histopathology of reactive gastropathy.jpg — wikimedia commons, the free media repository, 2020. [Online; accessed 9-May-2022].
- [88] Wikimedia Commons. File:std depth coded stack phalloidin stained actin filaments.png — wikimedia commons, the free media repository, 2020. [Online; accessed 9-May-2022].
- [89] Corinne Lorenzo, Céline Frongia, Raphaël Jorand, Jérôme Fehrenbach, Pierre Weiss, Amina Maandhui, Guillaume Gay, Bernard Ducommun, and Valérie Lobjois. Live cell division dynamics monitoring in 3d large spheroid tumor models using light sheet microscopy. *Cell division*, 6(1):1–8, 2011.
- [90] Anaïs Bouissou, Amscha Proag, Nicolas Bourg, Karine Pingris, Clément Cabriel, Stéphanie Balor, Thomas Mangeat, Christophe Thibault, Christophe Vieu, Guillaume Dupuis, et al. Podosome force generation machinery: a local balance between protrusion at the core and traction at the ring. *ACS nano*, 11(4):4028–4040, 2017.

performed using HPC resources from GENCI-IDRIS (Grant 2021-AD011012210R1).

ACKNOWLEDGMENTS

The authors wish to thank Emmanuel Soubies and Thomas Mangeat for fruitful discussions. This work was supported by the ANR Micro-Blind and by the ANR LabEx CIMI (grant ANR-11-LABX-0040) within the French State Programme “Investissements d’Avenir”. The authors acknowledge the support of AI Interdisciplinary Institute ANITI funding, through the French “Investing for the Future—PIA3” program under the Grant Agreement ANR-19-PI3A-0004. This work was



Fig. 7: Deep-Blur in action in the noiseless setting. Quantitative evaluations are reported in Table II. When available, the estimated blur kernel is displayed at the bottom-right. *First row*: original images. *Second row*: blurry-noisy images. *Third row*: Deep-Blur. *Fourth row*: [77] *Fifth row*: [78] *Sixth row*: [79] *Seventh row*: [80] *Eighth row*: [81]



Fig. 8: Deep-Blur in action with additive white Gaussian noise and $\sigma = 10^{-2}$. Quantitative evaluations are reported in Table II. When available, the estimated blur kernel is displayed at the bottom-right. *First row*: original images. *Second row*: blurry-noisy images. *Third row*: Deep-Blur. *Fourth row*: [77] *Fifth row*: [78] *Sixth row*: [79] *Seventh row*: [80] *Eith row*: [81]



Fig. 9: Deep-Blur in action with additive white Gaussian noise and $\sigma = 5 \cdot 10^{-2}$. Quantitative evaluations are reported in Table II. When available, the estimated blur kernel is displayed at the bottom-right. *First row*: original images. *Second row*: blurry-noisy images. *Third row*: Deep-Blur. *Fourth row*: [77] *Fifth row*: [78] *Sixth row*: [79] *Seventh row*: [80] *Eith row*: [81]

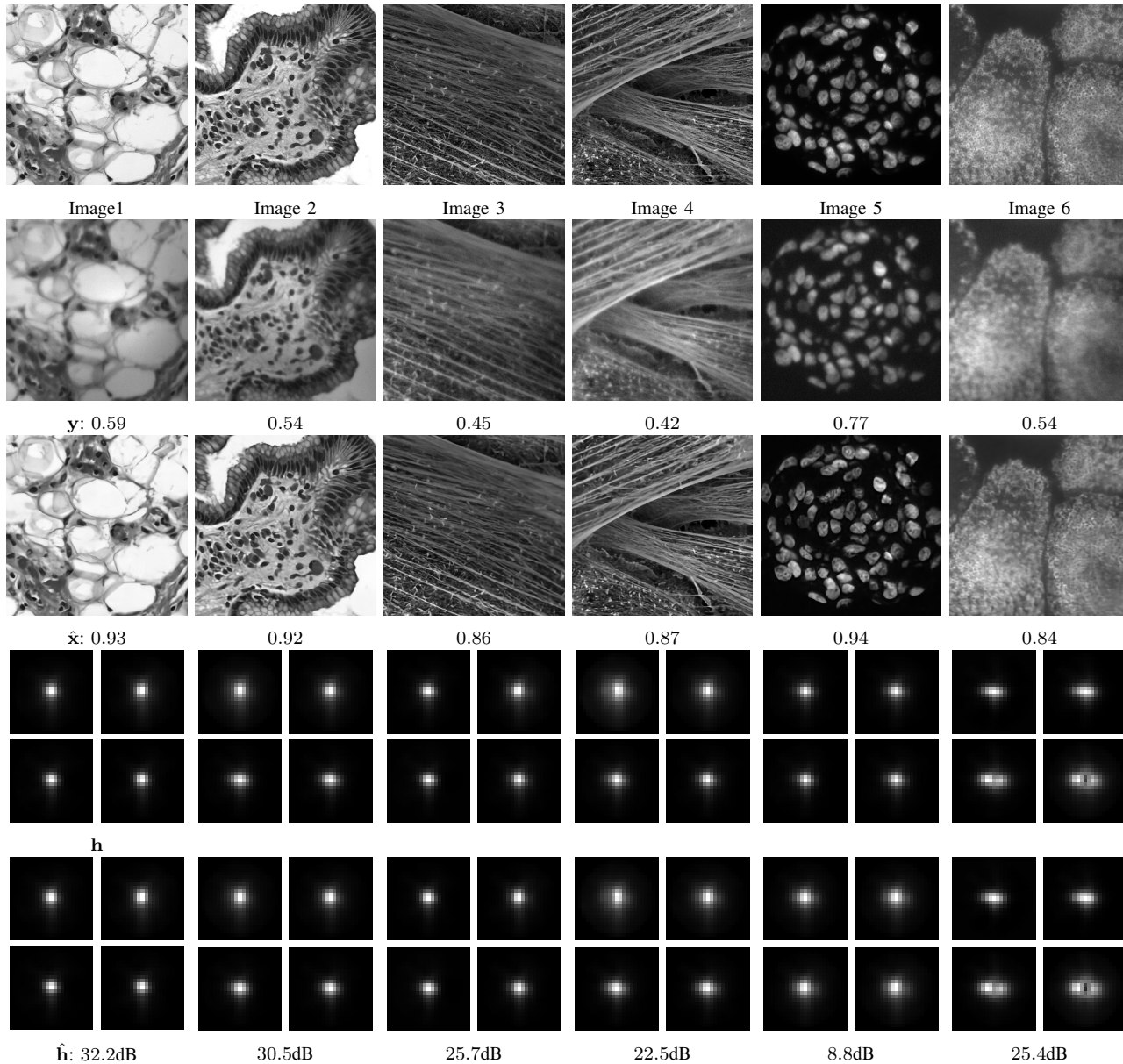


Fig. 10: Deep-Blur applied to spatially varying blur operators on microscopy images (not seen during training). The blur operators are sampled from a family estimated using a real wide-field microscope, as described in Paragraph V-A. *First row*: the original images. *Second row*: blurry-noisy images. *Third row*: the blind deblurring result with Deep-Blur. The SSIM of the resulting deblurred image is displayed below. *Fourth row*: The true blur operator. We display 4 evenly spaced impulse responses in the field of view. *Fifth row*: The estimated blur operator. The SNR of the estimated kernel is displayed in the caption in dB.