Digital Signal Processing 142 (2023) 104193

Contents lists available at ScienceDirect

Digital Signal Processing

journal homepage: www.elsevier.com/locate/dsp

Learning Moore-Penrose based residuals for robust non-blind image deconvolution ☆

Santiago López-Tapia [a], Javier Mateos [b,*], Rafael Molina [b], Aggelos K. Katsaggelos [a]

[a] Dept. of Electrical and Computer Engineering, Northwestern University, Evanston, IL, USA
[b] Dpto. de Ciencias de la Computación e I. A., Universidad de Granada, Spain

ARTICLE INFO

Article history:
Available online 18 August 2023

Keywords:
Robust non-blind image deconvolution
Deep learning
Convolutional neural network
Analytical methods
Moore-Penrose inverse

ABSTRACT

This paper proposes a deep learning-based method for image restoration given an inaccurate knowledge of the degradation. We first show how the impulse response of a Wiener filter can approximate the Moore-Penrose pseudo-inverse of the blur convolution operator. The deconvolution problem is then cast as the learning of a residual in the null space of the blur kernel, which, when added to the Wiener restoration, will satisfy the image formation model. This approach is expected to make the network capable of dealing with different blurs since only residuals associated with the Wiener filter have to be learned. Artifacts caused by inaccuracies in the blur estimation and other image formation model inconsistencies are removed by a Dynamic Filter Network. The extensive experiments carried out on several synthetic and real image datasets assert the proposed method's performance and robustness and demonstrate the advantage of the proposed method over existing ones.

1. Introduction

Image restoration is the problem of estimating an image $\mathbf{x}$ from its blurred and noisy version $\mathbf{y}$. Mathematically, the blurred image $\mathbf{y}$ is modeled as a convolution of the latent image $\mathbf{x}$ and the blur $\mathbf{H}$ as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}\,, \tag{1}$$

where $\mathbf{n}$ is the noise, usually additive white Gaussian noise (AWGN). Both $\mathbf{y}$ and $\mathbf{x}$, as well as $\mathbf{n}$, are $N \times 1$ vectors representing the lexicographically ordered corresponding images. In this paper, we assume that $\mathbf{H}$ the $N \times N$ matrix represents a spatially invariant two-dimensional convolution matrix of a point spread function (PSF), $\mathbf{h}$, of size $M \times 1$. If $\mathbf{H}$ is known, the problem is called Non-Blind Image Deconvolution (NBID). Otherwise, it is known as Blind Image Deconvolution (BID).

In the context of NBID, the presence of noise and the spectral properties of the operator $\mathbf{H}$ make it ill-posed. This means that even a small change in the data can cause a significant alteration in the solution. Additionally, the complexity of the problem increases in the blind case as it requires the estimation of $\mathbf{H}$.

Traditionally, image restoration algorithms have used the analytical approach that describes the forward model explicitly, decides the criteria for obtaining a solution, and chose an optimization procedure. These techniques need to optimize an energy function for each new image and blur. This provides them with high flexibility, allowing them to adapt to various blurs but at a high computational cost. Unfortunately, they are not amortized procedures. That is, there is no general fast procedure that, given an observed image, produces the corresponding restored image and blur.

In recent years, discriminative learning models, specifically those based on Deep Neural Networks (DNNs), have been used as a more efficient alternative to solve inverse problems [1]. Works on image denoising [2], inpainting [3] and superresolution [4] have shown that these methods can outperform analytical ones while being significantly faster. Unlike analytical methods, DNNs use large data sets to learn the restored image's mapping function. This makes them amortized procedures, thus significantly faster than approaches requiring iterative optimization for each new sample. Not only that but because they do not use the image formation model, they can be easily trained to be robust to inaccuracies caused by saturated pixels, underexposed areas, salt and pepper noise and, in the case of NBID, errors in the PSF estimation. However, they lack the flexibility of analytical models,

☆ This work was supported by grants P20_00286 and B-TIC-324-UGR20 funded by Consejería de Universidad, Investigación e Innovación (Junta de Andalucía) and by "ERDF A way of making Europe". Funding for open access charge: Universidad de Granada / CBUA.

* Corresponding author.
E-mail addresses: santiago.lopeztapia@northwestern.edu (S. López-Tapia), jmd@decsai.ugr.es (J. Mateos), rms@decsai.ugr.es (R. Molina), a-katsaggelos@northwestern.edu (A.K. Katsaggelos).

https://doi.org/10.1016/j.dsp.2023.104193

which significantly hampers their effectiveness in image restoration, where there is a high variation in the degradations caused by the combination of all the possible PSFs and noise. This is why, until recently, their application has been usually restricted to specific blur kernels and noise levels [5].

Recently, models that combine both approaches have been proposed to take advantage of the flexibility of analytical approaches and the effectiveness of DNNs. They use variable splitting techniques such as the alternating direction method of multipliers (ADMM) [6] and half-quadratic splitting (HQS) [7] to combine analytical and Deep Learning (DL) techniques. They split the recovery problem into two [8]: a regularized recovery one (*subproblem A*) which uses as penalty the squared Euclidean distance to an image. This image is estimated using an appropriate denoising technique (*subproblem B*). Subproblems A and B are easier to solve than the original ones. For shift-invariant degradation, subproblem A can easily be solved using the Discrete Fourier Transform (DFT), and subproblem B using any denoising algorithm. However, for each image, the solution must be found iteratively, which incurs a high computational cost.

To reduce the high computational cost of these approaches, several works [9–11] have proposed an alternative way to combine both analytical and DL models. It consists of two phases; First, an analytical model calculates one or several initial estimates of the reconstructed image. Then, these images are used by a DL model to produce the final estimation. The main idea behind these methods is to reduce the variability of the degradations that the DL model has to deal with. Inspired by these approaches, we propose a novel method to combine analytical and DL models for robust image restoration. Given an (inaccurate) estimation of the blur kernel, $\mathbf{H}$, we first approximate the Moore-Penrose pseudo-inverse of the blur convolution operator, $\mathbf{H}^+$, by a Wiener filter. Then, we cast the deconvolution problem as the learning of a residual in the null space of the kernel, that is, the recovering of the set of frequencies removed by the blur. This residual is added to the Wiener filter restoration to obtain the final estimation, which satisfies the image formation model. This strategy reduces the network's burden and makes it capable of dealing with different blurs and noise in the image formation model. To remove artifacts caused by inaccuracies in the estimated blur and other image formation model inconsistencies, we apply the kernels predicted by a Dynamic Filter Network (DFN) [12]. This results in a fast, accurate, and robust NBID model.

To summarize, the main contributions of this work are the followings:

1. We study the effect of using the Moore-Penrose pseudo-inverse, $\mathbf{H}^+$, and $\mathbf{H}^+\mathbf{y}$, which is approximated by the Wiener filter, for deconvolution problems. We show that $\mathbf{H}^+\mathbf{H}$ is not the identity. This product is one only for non-zero frequencies of the blur and zero for the rest. Therefore, the Wiener deconvolution has only information on those frequencies of the original image. This forces us to design a robust network capable of learning the frequency components of natural images from partial observations.
2. The initial Wiener deconvolution obtained using a possibly inaccurate blur estimate is utilized as input of a network with a novel architecture that uses filters predicted by a DFN to improve the quality of the deconvolved image. To the best of our knowledge, this is the first time DFNs have been used to remove spatially variant artifacts caused by inaccuracies in the estimated blur and other image formation model inconsistencies.
3. We show that the proposed approach outperforms similar CNN models twice as deep.

The rest of the paper is organized as follows: Section 2 presents a brief review of state-of-art approaches to image restoration. In section 3, the used notation and the proposed DL approach are presented. Section 4 describes the network training procedure. An ablation experiment is performed in Sec. 5 to determine the contribution of each component of the proposed CNN model. The performance of the proposed method is tested and compared to other classical and state-of-the-art deconvolution methods in Sec. 6. Finally, Sec. 7 concludes the paper.

2. Related works

Analytical techniques for image restoration have been studied for a long time. At a high level, one can group them into deterministic and stochastic ones. An optimization criterion is typically chosen within the first class, such as minimizing the l_2 error norm $\| \mathbf{y} - \mathbf{Hx} \|^2$. Then, prior (or domain) knowledge is incorporated into the solution process through regularization. With stochastic approaches, the unknowns are treated as stochastic quantities. Then a maximum likelihood, or a maximum *a posteriori* (MAP) or a fully (hierarchical) Bayesian approach is followed (see [13] for a review). Over the years, carefully designed image priors were based on constraints on the image gradients, while flat priors were used for the blur, see however [14]. Some examples of such image priors are hyper-Laplacian priors [15,16], log-TV priors [17], mixture of Gaussians (MoG) [18], Super Gaussian (SG) [14,19], Scale Mixture of Gaussian (SMG) [20], non-local adaptive tight frames [21] and generalized ℓ_p/ℓ_q norm-based priors [22]. Approximations of the l_0 function (see [23], extended with the use of a dark channel prior [24] and an extreme channel prior [25]) have also been successfully used.

As it can be deduced from the methods introduced previously, within the blind scenario, analytical techniques usually use an iterative approach to estimate the blur and the latent sharp image in an alternate way. The majority of these techniques concentrate first on obtaining a precise blur kernel estimation. In this phase, the extraction of a clear image is restricted to only the most prominent features that aid in estimating the blur. Later, a separate non-blind approach is required to compute the sharp image, leading to a dual-phase strategy (initially estimating the blur, followed by non-blind deconvolution) [13,19,26]. Errors made during the initial blur estimation phase can cause significant issues such as ringing and distortions [27], even when there are minor inaccuracies in the kernel. This highlights the need for developing NBID methods that can withstand kernel inaccuracies.

There is limited research available in the literature regarding NBID that specifically addresses blur estimation errors. Within the analytical approaches, kernel inaccuracy has been modeled as zero-mean white noise added to the kernel [28] or a residual term added to the degradation model [29,30]. Although artifacts are well suppressed by the method in [29], details in the image are usually over-smoothed. In more recent studies, [31] employed a partial map in the Fourier domain to model kernel estimation errors, [32] proposed a prior tailored to remove ringing artifacts, and [33] introduced a robust NBID method to handle kernel errors through bias correction based on the classic errors-in-variables (EIVs) model. It should also be noted that kernel estimation errors are not the sole outlier type in the model presented in (1). Local violations of the convolution model can occur due to saturated pixels, underexposed regions, or hot or dead pixels. To address these outlier types, various robust NBID methods have been suggested, as seen in [34–37].

Although DL techniques were easily applied to denoising [2], inpainting [3] and superresolution [4], the significant diversity of degradations caused by the combination of all possible PSFs and noise has hindered the implementation of DL models in image

restoration. The majority of research employing only DL models for image restoration limits them to specific blur types, with blind non-uniform motion blur removal being the most frequently addressed task. Refer to [38] for an up-to-date review. In [39], a multi-scale residual CNN is utilized to remove non-uniform motion blur from images. Adopting a similar strategy, the authors of [40] propose a multi-stage progressive image restoration network for retrieving blurred images. The application of GANs is introduced in [41] and further refined in [42].

Rather than restricting deep learning models to particular blurs, other studies have suggested combining them with analytical methods. In the case of BID, [26] introduced a deep CNN to learn a discriminative regularizer. This regularizer provides an output in the interval $[0, 1]$, which distinguishes between clear (0) and blurred images (1). In [43] a generalization of the traditional iterative total-variation regularization method in the gradient domain is unrolled to construct a neural network.

For the NBID problem, several solutions have been proposed using variable splitting techniques, introduced in Section 1. Such is the case for [44] and [45], where the proximal operator [46] associated with subproblem B is replaced by a denoising neural network. In [47], a combination of adversarial learning and a denoising autoencoder is chosen as the proximal operator to project the images into the natural images' space. In [48], the authors opted for a denoising autoencoder network inspired by the U-Net architecture. The authors of [49] leverage this approach to develop a method robust to kernel inaccuracies. In the same context, RGDN [50] integrates both subproblems into a recurrent convolutional network, where a common CNN block replaces the gradient of the image prior unit. Similarly, [51] makes use of unrolling techniques to leverage a new architecture for robust NBID. The authors of [52] extend this approximation to BID using a residual network to estimate the noise in subproblem B, subtracted from the currently estimated image to obtain the restored one.

Another approach consists of replacing the prior model by a deep CNN that generates the output image. Following this approach, [53] extends to image deconvolution the deep image prior in [54] and [55,56] further adapt it to BID. The authors of [57] further illustrate the power of untrained neural architectures applying them to NBID. In the same context, [58] explicit prior models have been replaced by a neural network trained for denoising, and in [59], the image prior is learned for robust deblurring.

Recently, several NBID works have proposed a different scheme to combine both analytical and DL models. The analytical model estimates an initial solution that is then used by the DL model to produce the final estimation. The authors of [9] use a U-net to remove the noise introduced by a Tikhonov deconvolution of galaxy surveys. In [10], the authors introduce a CNN model fed with several estimates of the latent image. These estimates are obtained using the method in [15] with different prior strengths and provide complementary information that the network combines into the restored image. Another approach, that explicitly handles outliers in the image formation model for deblurring, is proposed in [60] where outliers identification is performed by a CNN. This information is used by an analytical IR method. Assuming an exact blur knowledge, [11] integrates a classical Wiener deconvolution framework with learned deep features and a multi-scale feature refinement module to estimate the deblurred image from the deconvolved deep features. This results in a method empirically robust to some model inaccuracies.

3. Deep learning model

As previously stated, NBID is a very challenging task for analytical and DL models due to the variability of the degradations and the ill-posed nature of the problem. Even if the blur is exactly known, the data noise results in large perturbations in the solution. Moreover, the great variability of degradations caused by different combinations of blurring and noise makes it very difficult to train a DL model that can generalize well to blurs not seen during its training. To ease the problem, the image formation model has to be separated from the network parameters' learning.

In this section, we first introduce how we separate the degradation from the network parameters' learning in the simpler case of known blur and no added noise. Then, in Sect. 3.2, we extend it to the general model in (1) assuming a known (but possibly inaccurate) blur. A Dynamic Filter Network (DFN) [12] is proposed in Sect. 3.3 to remove spatially variant artifacts that may arise in the restoration process. Finally, the proposed network architecture is presented in Section 3.4.

3.1. Noise-free known blur scenario

Let us consider the much simpler scenario of image deconvolution with known blur $\mathbf{H}$ and no added noise; that is, we consider the degradation model $\mathbf{y} = \mathbf{H}\mathbf{x}$.

Ideally, to estimate the sharp image $\mathbf{x}$, we should calculate $\mathbf{x} = \mathbf{H}^{-1}\mathbf{y}$. However, the inverse of $\mathbf{H}$ may not exist. To tackle this problem, in this work we make use of the Moore-Penrose pseudo-inverse of the blur, $\mathbf{H}^{+}$. Let us first study its properties.

It can be shown [61] that

$$\mathbf{H}^{+} = \lim_{\delta \to 0^{+}} (\mathbf{H}^{\mathbf{T}}\mathbf{H} + \delta \mathbf{I})^{-1}\mathbf{H}^{\mathbf{T}} \tag{2}$$

and so, $\mathbf{H}^{+}\mathbf{y}$ recovers the frequencies of $\mathbf{x}$ but those for which the Fourier transform of the blurring filter is zero. This set of frequencies forms the null space of the kernel $\mathbf{H}$ [62]. Note that, for these frequencies, it is not possible to revert the blur effects by using the Moore-Penrose pseudo-inverse and artifacts arise. Hence, we need to recover (learn) those frequencies of the original image. The rest of the frequencies are in $\mathbf{H}^{+}\mathbf{y}$.

It is also easy to show that for the i-th discrete Fourier frequency of $\mathbf{H}\mathbf{H}^{+}\mathbf{y}$, $\mathcal{F}_{\mathbf{H}\mathbf{H}^{+}\mathbf{y}}(i)$, we have

$$\mathcal{F}_{\mathbf{H}\mathbf{H}^{+}\mathbf{y}}(i) = \begin{cases} \mathcal{F}_{\mathbf{y}}(i) & \text{if } \mathcal{F}_{\mathbf{H}}(i) \neq 0 \\ 0 & \text{if } \mathcal{F}_{\mathbf{H}}(i) = 0, \end{cases} \tag{3}$$

that is, in the Fourier domain, $\mathbf{H}\mathbf{H}^{+}\mathbf{y}$ is equal to $\mathbf{y}$, since $\mathcal{F}_{\mathbf{y}}$ is zero when is $\mathbf{H}$ zero.

To learn the frequencies in $\mathbf{x}$ not present in $\mathbf{H}^{+}\mathbf{y}$, we follow the approach proposed in [63] for image super resolution and define $\mathrm{g}_{\theta}(\mathbf{z})$, the solution to our deconvolution problem, as

$$\mathrm{g}_{\theta}(\mathbf{z}) = (\mathbf{I} - \mathbf{H}^{+}\mathbf{H})\mathrm{f}_{\theta}(\mathbf{z}) + \mathbf{H}^{+}\mathbf{y}, \tag{4}$$

where $\mathrm{f}_{\theta} : \mathcal{R}^{L} \to \mathcal{R}^{N}$ is a CNN with parameters θ, whose input $\mathbf{z}$ will be clearly specified later.

Note that, since $\mathbf{H}\mathbf{H}^{+}\mathbf{H} = \mathbf{H}$, we have

$$\mathbf{H}\mathrm{g}_{\theta}(\mathbf{z}) = \mathbf{H}(\mathbf{I} - \mathbf{H}^{+}\mathbf{H})\mathrm{f}_{\theta}(\mathbf{z}) + \mathbf{H}\mathbf{H}^{+}\mathbf{y} = \mathbf{H}\mathbf{H}^{+}\mathbf{y} = \mathbf{y}. \tag{5}$$

Using (3), this means that, although, in contrast to image and video super-resolution, $\mathbf{H}\mathbf{H}^{+} \neq \mathbf{I}$, $\mathrm{g}_{\theta}(\mathbf{z})$ is a solution to the problem $\mathbf{y} = \mathbf{H}\mathbf{x}$.

Our proposed restored image, $\mathrm{g}_{\theta}(\mathbf{z})$ satisfies

$$\mathbf{x} - g_{\theta}(\mathbf{z}) = (\mathbf{x} - \mathbf{H}^{+}\mathbf{y}) - (\mathbf{I} - \mathbf{H}^{+}\mathbf{H})\mathrm{f}_{\theta}(\mathbf{z}), \tag{6}$$

and neither $\mathbf{H}^{+}\mathbf{y}$ nor $(\mathbf{I} - \mathbf{H}^{+}\mathbf{H})$ have information on the values of $\mathbf{H}$, only on where its frequencies are (non)zero. Using this strategy, we dramatically reduce the dependency of the residual on the blur $\mathbf{H}$, thus separating the degradation from the network parameters' learning. Then, with the proper network design, we can deal with different degradations since all of them are expected to hide a similar set of high frequencies.

3.2. General model

In the previous subsection, we have shown how to solve the noise-free known-blur image deconvolution problem by separating the degradation and the learning of the parameters of a network that does not depend on the specific blur. Unfortunately, for the general scenario in (1), the presence of noise hampers the use of the model in (4) as is.

To ameliorate this problem and also to deal with the ill-posed nature of the image deconvolution, we approximate $\mathbf{H}^+$ defined in (2) by

$$\mathbf{H}_\epsilon^+ = (\mathbf{H}^\mathbf{T}\mathbf{H} + \epsilon\mathbf{I})^{-1}\mathbf{H}^\mathbf{T}, \tag{7}$$

where $\epsilon > 0$ is a regularization parameter. This approximation controls the noise and avoids the loss of those frequencies for which $\mathbf{H}$ is zero (see (3)). Note that (7) can be seen as an approximation to the classical Wiener filter [64], where ϵ represents the ratio between the power spectral density of the noise and the power spectral density of the image.

Then, we rewrite (4) as

$$\mathrm{g}_\theta(\mathbf{z}) = (\mathbf{I} - \mathbf{H}_\epsilon^+\mathbf{H})\mathrm{f}_\theta(\mathbf{z}) + \mathbf{H}_\epsilon^+\mathbf{y} = (\mathbf{I} - \mathbf{H}_\epsilon^+\mathbf{H})\mathrm{f}_\theta(\mathbf{z}) + \mathbf{x}_\mathbf{w}, \tag{8}$$

where $\mathbf{x}_\mathbf{w} = \mathbf{H}_\epsilon^+\mathbf{y}$ is the rough estimation of the sharp image $\mathbf{x}$ obtained by an approximation of the Wiener filter (for simplicity, we will use the term Wiener filter instead of approximation of the Wiener filter throughout the paper). It can be shown that by reducing $\| \mathbf{x} - \mathrm{g}_\theta(\mathbf{z}) \|^2$ we also reduce the data fidelity error since

$$\begin{aligned} \| \mathbf{y} - \mathbf{H}\mathrm{g}_\theta(\mathbf{z}) \|^2 &\leq \| \mathbf{H}\mathbf{x} - \mathbf{H}\mathrm{g}_\theta(\mathbf{z}) \|^2 + \| \mathbf{n} \|^2 \\ &\leq \| \mathbf{H} \|^2 \| \mathbf{x} - \mathrm{g}_\theta(\mathbf{z}) \|^2 + \| \mathbf{n} \|^2 \\ &\leq N \| \mathbf{x} - \mathrm{g}_\theta(\mathbf{z}) \|^2 + \| \mathbf{n} \|^2, \end{aligned} \tag{9}$$

which means that we only need to learn the network $\mathrm{f}_\theta(\mathbf{z})$ to solve the problem in (1). Notice that we are making implicit use of the observation model and that the proposed approach constrains the functions that the network learns to only those that produce images consistent with the observation $\mathbf{y}$.

We now have an interpretation of the goal of the network. As in the case of the noise-free known blur scenario presented in Section 3.1, we have cast the deconvolution problem as the learning of a residual in the null space of the kernel, which when added to the Wiener filter restoration will satisfy the image formation model. The network learns the "missing" frequencies in $\mathbf{x}_\mathbf{w}$ and to remove the noise and the artifacts introduced by the Wiener filter. These artifacts, although dependent on the image, blur, and noise combination, are all very similar in nature, consisting mainly of amplified noise and ringing. Learning the residual associated with the Wiener filter reduces the burden of the network and makes it capable of dealing with different blurs. Furthermore, note that $\mathbf{x}_\mathbf{w}$, although noisy, contains information on the high frequencies of the original image $\mathbf{x}$, which makes the recovery task easier.

Let us now see what the input to the network is. The function $\mathrm{f}_\theta(\cdot)$ (and consequently $\mathrm{g}_\theta(\cdot)$) takes as input both $\mathbf{x}_\mathbf{w}$ and $\mathbf{y}$ concatenated along the channel dimension, i.e., if both images are one-channel gray- scale images, the input to the network would be an image with two channels. In other words,

$$\mathbf{z} = (\mathbf{x}_\mathbf{w}, \mathbf{y}). \tag{10}$$

The use of the blurred image $\mathbf{y}$ is necessary since some artifacts that appear in the Wiener filter solution, $\mathbf{x}_\mathbf{w}$, are difficult to distinguish from real-world-like structures, making them difficult to remove. Fig. 1 presents an example of these artifacts. Incorporating $\mathbf{y}$ as an input helps the network distinguish between artifacts and

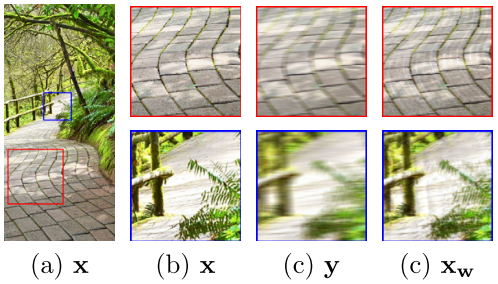


Fig. 1. An illustration of the artifacts caused by the Wiener filter. Ringing artifacts in the red box are compatible with natural images and are difficult to distinguish from real structures, unlike those in the blue box.

real scene objects, thus removing structures not present in $\mathbf{y}$ and preserving the consistency with the observation. To avoid boundary artifacts, before applying the Wiener filter, the blurred image is padded by repeating the last pixel of each image boundary and decaying their values linearly to zero.

3.3. Removal of deconvolution artifacts

Since the image formation model in (1) is only approximate because it does not include, for instance, saturated pixels or underexposed areas, and inaccurate blur estimations, the deconvolved image obtained by the operator $\mathrm{g}_\theta(\cdot)$ introduced in the previous subsection still exhibits some spatially variant artifacts. To remove them, we propose to use a DFN [12]. The DFN comprises a filter-generating network and a dynamic filtering layer. The former dynamically generates filters that depend on the network input values and pixel position. The latter then applies those filters to the input.

For a pixel (i, j), let $\mathbf{d}_\psi^{\mathbf{u}_{i,j}}$ be the filter of support $(2L+1) \times (2M+1)$ generated by the filter generating network with parameters ψ from the input image $\mathbf{u}$ (notice that we are using bidimensional notation and that the network generates the filter from a region centered around $u(i, j)$). The set of all filters generated from image u is denoted as $\mathbf{d}_\psi^\mathbf{u}$. Then, the pixel at position (i, j) of the filtered image $\mathbf{r}$ obtained by the dynamic filtering layer on the input image $\mathbf{q}$ with the set of filters $\mathbf{d}_\psi^\mathbf{u}$ (notice that $\mathbf{u}$ and $\mathbf{q}$ do not have to coincide) is defined as

$$r(i, j) = \sum_{l=-L}^{L} \sum_{m=-M}^{M} d_\psi^{\mathbf{u}_{i,j}}(l + L, m + M) q(i + l, j + m) \tag{11}$$

Hence, the application of the dynamic filtering in (11) to $\mathrm{g}_\theta(\cdot)$ is expected to be a sharp, artifact-free restored image given by

$$\hat{\mathbf{x}} = \mathrm{r}_{\theta,\psi}(\mathbf{x}_\mathbf{w}, \mathbf{y}) = \mathbf{D}_\psi^{\mathbf{x}_\mathbf{w},\mathbf{y}} \mathrm{g}_\theta(\mathbf{x}_\mathbf{w}, \mathbf{y}), \tag{12}$$

where $\mathbf{D}_\psi^{\mathbf{x}_\mathbf{w},\mathbf{y}}$ is the (spatially variant) convolution matrix associated with the set of kernels $\mathbf{d}_\psi^{\mathbf{x}_\mathbf{w},\mathbf{y}}$. Notice that a DFN is a perfect suit for this final refinement since it can produce spatially-variant filters. In contrast, normal CNN filters are spatially-invariant, requiring far more parameters to adapt to these artifacts' spatially variant nature. The whole model used to deconvolve blurred images, which corresponds to $\mathrm{r}_{\theta,\psi}(\mathbf{x}_\mathbf{w}, \mathbf{y})$, will be denoted by DDNet.

3.4. DDNet's architecture

The architecture of DDNet is shown in Fig. 2. The blue box represents the model $\mathrm{g}_\theta(\mathbf{x}_\mathbf{w}, \mathbf{y})$ defined in (8). To reduce the number of computations, its main branch, $\mathrm{e}_\phi(\cdot)$, is shared with the filter generating network of the DFN, $\mathrm{d}_\psi(\cdot)$. This main branch follows

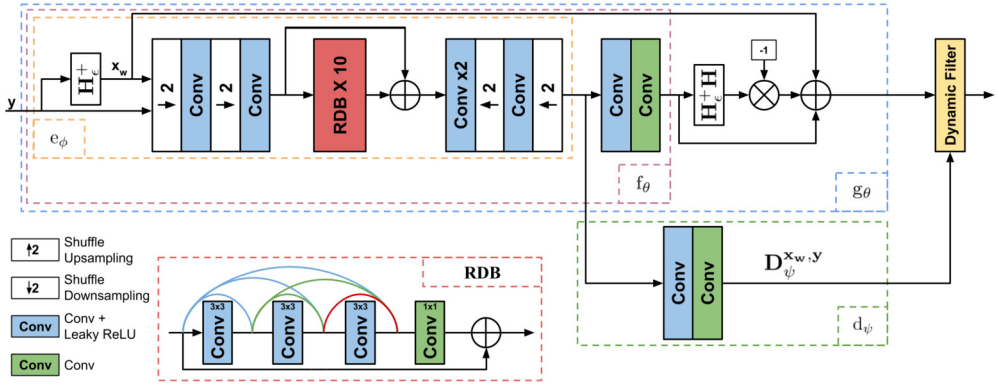


Fig. 2. The proposed DDNet architecture. RDB indicates a Residual Dense Block (RDB) and Dynamic Filter refers to the operation in (12). Each non-output convolution outside an RDB has $64 \times 3 \times 3$ filters. Except for the last one, which uses $64 \times 1 \times 1$ filters, convolutions in RDBs have $32 \times 3 \times 3$ filters. The size of the filters used during dynamic filtering is 5×5. See text for details.

an encoder-decoder structure. During the encoding phase, we extract features from the image. At the same time, we reduce its spatial resolution by a factor of 4 using two space-to-depth operations defined as $S_2 : [0,1]^{2H \times 2W \times C} \rightarrow [0,1]^{H \times W \times 4C}$. In this projected space, the features are transformed using 10 Residual Dense Blocks (RDB) [65]. An RDB is a modified Residual Block (RB) with dense connections between layers. These dense connections allow the reuse of features and provide better performance. The features are finally up-scaled to the original image size using two sub-pixel convolutions [66] of factor 2. Notice that the spatial size reduction allows the network to process the image much faster and increases the receptive field without increasing the CNN model's depth. The features obtained by the main branch are used to calculate the initial image estimation, $g_\theta(\mathbf{x_w}, \mathbf{y})$, and the filters $d_\psi(\mathbf{x_w}, \mathbf{y})$. The result of applying the filters, calculated via the dynamic filter module, produces the final estimation $\hat{\mathbf{x}}$ in (12) of our proposed $r_{\theta,\psi}(\mathbf{x_w}, \mathbf{y})$ model.

All of the non-output convolutions outside an RDB block use $64 \times 3 \times 3$ filters and are followed by a Leaky ReLU with a negative slope set to 0.2. Convolutions in each RDB only use $32 \times 3 \times 3$ filters, except for the last convolution that uses $64 \times 1 \times 1$ filters, to control the increment in the number of parameters due to the use of dense connections. The size of the filters predicted by $d_\psi(\cdot)$ is 5×5, $M = L = 2$.

4. Model training

To train the proposed model, we generated an image dataset based on the COCO 2017 Dataset.[1] It has been used for object segmentation and recognition. It comprises many sharp natural images divided into a train set of 118287 images, a validation set of 5000 images and a test set of 40670 images. To simulate the degraded images, we proceeded as follows. First, we generated 1024 PSFs using the method in [67] of size between 11×11 and 65×65 pixels, with $T = 0.8$ and anxiety $10^r/1000$, where r is a random number from a uniform $[0, 1]$ distribution. We used 736 kernels for training, 32 for validation and 256 for testing. This follows approximately the same proportion of the images in each subset of COCO 2017. Each image in the training and validation sets was blurred with exactly 3 PSFs on their own set: training images with training kernels and validation images with validation kernels, and Gaussian noise of standard deviation $\sigma = 0.01$ was added to the blurred image. Images with a size smaller than 320 pixels in the shorter dimension were discarded to avoid boundary artifacts. Finally, the 256×256 central part of each image was cropped. This process generated a training set of 347436 images and a validation set of 14637 images. For testing, we created a reduced set of 512 images, obtained by degrading two randomly chosen test images for each kernel in the test set and adding Gaussian noise with $\sigma = 0.01$.

Since, very frequently, the real blur kernel is not available, we use a variation of the fast Bayesian BID method in [19] to obtain sufficiently accurate estimations of the blur in an offline manner, i.e., before the training. The blur is estimated using a multiscale coarse-to-fine approach to avoid local minima. At each scale, the method enforces sparsity on high pass filtered reconstructions, that is, on the edges of the image, using a Huber Super Gaussian (HSG) prior. For the blur, no prior knowledge is assumed other than non-negativity and that the blur coefficients should add to one. In this paper, we modify the method in [19] by adding a heuristic kernel cleaning step at the end of each scale [25]. This step computes the 8-neighbors connected components of the kernel, treated as a binary image where nonzero pixels are considered to be on, those connected components whose sum is smaller than a given threshold, i.e., 0.1, are set to zero. This removes from the kernel isolated pixels or sets of small valued connected pixels that are not connected to a path with a high-value sum. Note that those small values are usually due to noise and setting them to zero contributes to PSF estimates comprised of relevant connected pixels. We will refer to this kernel estimation method as CHSG throughout the rest of the paper.

During training, data augmentation was performed by random vertically and horizontally flipping and rotating each instance by multiples of 90°. We trained our models for 35 epochs using Adam optimizer [68] with weight decay set to 10^{-4}. Each epoch of the training consisted of 5428 batches of 64 images. The learning rate was set to 5×10^{-4} for the first 5 epochs, to 10^{-4} for the next 20 epochs, and to 10^{-5} for the last 10 epochs. We set the regularization parameter for the Wiener filter, ϵ, equal to 0.01.

Finally, to train the DDNet we use the Charbonnier loss, defined as

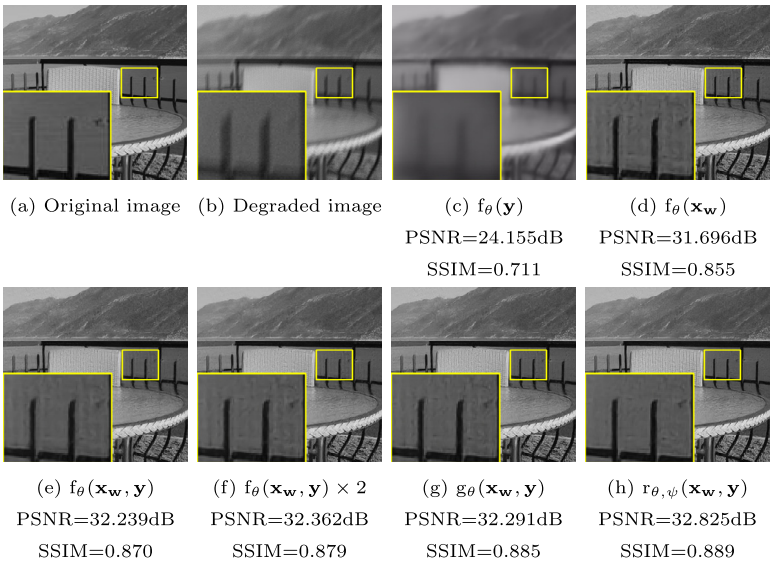
$$\mathcal{L}(\hat{\mathbf{x}}, \mathbf{x}) = \sum_{i=1}^{N} \sqrt{(\hat{\mathbf{x}}(i) - \mathbf{x}(i))^2 + \varepsilon^2}\,, \tag{13}$$

where $\hat{\mathbf{x}}$ and $\mathbf{x}$ are, respectively, the estimated and the real image and ε is a constant, which in our experiments was set equal to 10^{-3}. We use this loss instead of the Mean Squared Error (MSE) since it is more robust to outliers [69]. The implementation of the model and training was performed using Pytorch [70].

[1] http://cocodataset.org/.

Table 1
Ablation study of components of the proposed DDNet. The comparison is done using our validation dataset and in terms of PSNR and SSIM. The best result is marked in bold. The second-best result is underlined.

	$f_\theta(\mathbf{y})$	$f_\theta(\mathbf{x_w})$	$f_\theta(\mathbf{x_w}, \mathbf{y})$	$f_\theta(\mathbf{x_w}, \mathbf{y}) \times 2$	$g_\theta(\mathbf{x_w}, \mathbf{y})$	$r_{\theta,\psi}(\mathbf{x_w}, \mathbf{y})$
PSNR	20.12	25.93	26.15	<u>26.42</u>	26.37	**26.59**
SSIM	0.6053	0.7189	0.7267	<u>0.7355</u>	0.7341	**0.7516**



(a) Original image (b) Degraded image (c) $f_\theta(\mathbf{y})$ PSNR=24.155dB SSIM=0.711 (d) $f_\theta(\mathbf{x_w})$ PSNR=31.696dB SSIM=0.855

(e) $f_\theta(\mathbf{x_w}, \mathbf{y})$ PSNR=32.239dB SSIM=0.870 (f) $f_\theta(\mathbf{x_w}, \mathbf{y}) \times 2$ PSNR=32.362dB SSIM=0.879 (g) $g_\theta(\mathbf{x_w}, \mathbf{y})$ PSNR=32.291dB SSIM=0.885 (h) $r_{\theta,\psi}(\mathbf{x_w}, \mathbf{y})$ PSNR=32.825dB SSIM=0.889

Fig. 3. Visual comparison of components of the proposed DDNet.

5. Ablation experiments

To determine the contribution of each component of the proposed DDNet to the final solution, we performed an experimental ablation study where we added each main component one at a time. We consider the following models:

1. $f_\theta(\mathbf{y})$: The base CNN architecture. We use our proposed CNN architecture using only the blurred image as input without the Wiener filtered image; no approximation in (8) and no dynamic filtering.
2. $f_\theta(\mathbf{x_w})$: The base CNN architecture using only the Wiener-filtered image as network input.
3. $f_\theta(\mathbf{x_w}, \mathbf{y})$: The base CNN architecture using both the Wiener-filtered image and the blurred one as network inputs.
4. $f_\theta(\mathbf{x_w}, \mathbf{y}) \times 2$: The base CNN architecture using both the Wiener filtered image and the blurred one as inputs, but using 20 RD-Blocks instead of 10.
5. $g_\theta(\mathbf{x_w}, \mathbf{y})$: The approximation defined in (8) without dynamic filtering.
6. $r_{\theta,\psi}(\mathbf{x_w}, \mathbf{y})$: Our complete DDNet.

The results of testing each model on the validation dataset are shown in Table 1. We also include a visual comparison that can be seen in Fig. 3. From the table results, it is clear that each one of the proposed components significantly increases the model's performance when added. The most remarkable performance increase is due to the addition of the Wiener filtered image, $\mathbf{x_w}$, to the model inputs. The difference in PSNR/SSIM between both $f_\theta(\mathbf{y})$ and $f_\theta(\mathbf{x_w})$ can be explained by the fact that $f_\theta(\mathbf{y})$ is not able to remove the blur from the image, as can be seen in Fig. 3(c). However, this behavior was expected since $f_\theta(\mathbf{y})$ solves a much harder problem than the rest. By using $\mathbf{x_w}$, the problem is reduced to a denoising one. However, as can be seen by the increase in performance when using both $\mathbf{y}$ and $\mathbf{x_w}$ as input, $\mathbf{x_w}$ does not include enough information to easily detect and remove some of the noise and the artifacts present at $\mathbf{x_w}$. This difference in metrics is also apparent when comparing the results of $f_\theta(\mathbf{x_w})$ and $g_\theta(\mathbf{x_w}, \mathbf{y})$ shown in Fig. 3(d-e), respectively. Indeed, the result of $f_\theta(\mathbf{x_w})$ shows more ringing artifacts. Notice also that incorporating the approximation of the Wiener filter in (8) allows $g_\theta(\mathbf{x_w}, \mathbf{y})$ to obtain a performance similar to $f_\theta(\mathbf{x_w}, \mathbf{y}) \times 2$ with half the number of learnable parameters and being almost twice as fast. Despite the good PSNR/SSIM shown by $g_\theta(\mathbf{x_w}, \mathbf{y})$, some artifacts are still noticeable in Fig. 3(g). Finally, our complete DDNet, $r_{\theta,\psi}(\mathbf{x_w}, \mathbf{y})$, clearly outperforms all other models while keeping a complexity similar to $g_\theta(\mathbf{x_w}, \mathbf{y})$. Moreover, as shown in Fig. 3(h), $g_\theta(\mathbf{x_w}, \mathbf{y})$ is able to remove most artifacts thanks to the filters predicted by the DFN.

6. Experimental results

6.1. Materials and methods

To assess the performance and robustness of the proposed DDNet method, we have tested it on several image datasets, including synthetic datasets with spatially invariant blurs as well as real image datasets. More concretely, together with the test images from COCO 2017 dataset (see Section 4), we have used the datasets proposed by Levin et al. [71] and Lai et al. [72]. Details of each dataset are shown in Table 2. While those datasets are synthetically generated, we also tested the proposed method on the real images dataset proposed in Lai et al. [72], a set of 100 real images from multiple sources and different categories.

The proposed DDNet method was compared with classic and state-of-the-art deconvolution methods. These include a set of BID methods comprised by two analytical blind deconvolution methods, the Huber super Gaussian prior (HSG) method in [19] and the extreme channel prior (ECP) method in [25], the combined analytical and DL (Li) method in [26], the method in [55] (SelfDeblur), the GAN based (DeblurGAN-v2) method in [42] and CNN based

Table 2
Description of the synthetic datasets used to assess the performance and robustness of the proposed DDNet method and compare against other state-of-art methods.

Dataset	# Sharp images	# Blurred images	Image size	Type	# PSFs	PSF size	Noise
COCO 2017	512	512	256×256	Color	256	11×11 to 65×65	0.01 AWGN
Levin et al. [71]	4	32	255×255	Gray	8	13×13 to 27×27	0.01 AWGN
Lai et al. [72]	25	100	350×500 to 1024×768	Color	4	31×31 to 75×75	0.01 AWGN 0.01 AWGN

Table 3
Numerical comparison of the proposed DDNet and competing methods on different image datasets. The best result is highlighted in bold. The second-best result is underlined.

Dataset	Method	PSNR	SSIM	Time (Total/PSF + NBD)
COCO 2017	HSG [19]	29.018	0.832	24 / 23+1
	CHSG+Wiener [64]	27.497	0.743	23.2 / 23.1+0.1
	CHSG+Robust [29]	<u>29.063</u>	0.827	323.1 / 23.1+300
	ECP [25]	26.874	0.804	125
	Li [26]	26.196	0.785	1095
	SelfDeblur [55]	22.172	0.478	1494
	DeblurGAN-v2 [42]	26.668	0.730	0.26
	CHSG+DWDN [11]	27.830	<u>0.835</u>	24.2 / 23.1+1.1
	MPRNet [40]	26.534	0.743	0.24
	CHSG+MUNID [57]	28.520	0.810	1547.1 / 23.1+1524
	CHSG+KerUnc [49]	29.589	0.842	30.9 / 23.1+7.8
	CHSG+DDNet	**30.320**	**0.860**	23.12 / 23.1+0.02
	Real PSF+DDNet	31.119	0.880	0.02
Levin [71]	HSG [19]	31.800	<u>0.931</u>	11.2 / 10+1.2
	CHSG+Wiener [64]	27.180	0.751	10.08 / 10 + 0.08
	CHSG+Robust [29]	<u>31.995</u>	<u>0.931</u>	193 / 10+183
	ECP [25]	31.920	<u>0.931</u>	100
	Li [26]	30.304	0.896	149
	SelfDeblur [55]	29.673	0.860	399
	DeblurGAN-v2 [42]	26.239	0.800	0.2
	MPRNet [40]	28.095	0.842	0.3
	CHSG+MUNID [57]	31.039	0.898	170 / 10+160
	CHSG+KerUnc [49]	32.286	0.926	12 / 10+2
	CHSG+DDNet	**33.367**	**0.947**	10.02 / 10+0.02
	Real PSF+DDNet	35.202	0.962	0.02
Lai [72]	HSG [19]	24.605	0.783	169 / 156.3+12.7
	CHSG+Wiener [64]	23.247	0.682	157.7 / 156.5+1.2
	CHSG+Robust [29]	**24.996**	<u>0.790</u>	4235.4 / 156.5+4078.9
	ECP [25]	23.988	0.752	1011
	Li [26]	24.565	0.771	981
	SelfDeblur [55]	22.344	0.637	3407
	DeblurGAN-v2 [42]	19.644	0.581	0.4
	CHSG+DWDN [11]	23.182	0.754	158.5 / 156.5+2
	MPRNet [40]	18.781	0.567	0.3
	CHSG+MUNID [57]	24.034	0.749	3479.4 / 156.5+3322.9
	CHSG+KerUnc [49]	24.588	0.782	168.8 / 156.5+12.3
	CHSG+DDNet	<u>24.775</u>	**0.798**	156.58 / 156.53+0.05
	Real PSF+DDNet	27.218	0.887	0.05

method MPRNet [40]. As a baseline method, we used the Wiener [64] method with the kernels estimated by the CHSG method in Sect. 4, which we will refer to as CHSG + Wiener. Using these same kernels estimated by the CHSG method, we also compared with the non-blind deconvolution methods handling inaccurate kernels (CHSG + Robust) in [29], which uses an analytical approach, and (CHSG + KerUnc) in [49] which uses a DL approach. Also, the DL based Deep Wiener (CHSG + DWDN) method in [11] and the (CHSG + MUNID) method in [57] were used in this comparison. Since the Robust method in [29] is designed to handle only grayscale images, the method is applied to each one of the RGB channels independently. Notice that we did not include the results of CHSG + DWDN for Levin [71] gray scale image dataset because the authors only provide the model weights for color images. As an upper bound of the performance of the method, we also run the proposed DDNet method using the real kernel as a blur estimate. All methods were run using the default parameters or the parameters suggested by the authors in the corresponding paper. For DDNet, we used $\epsilon = 0.01$ for all the datasets except for the Lai dataset, where $\epsilon = 0.003$ was used since images degraded with large PSFs need a smaller regularization parameter. Note that, for large PSFs, the values of $\mathbf{H}$ are smaller than ϵ. To simulate a real environment, a fixed PSF support of 65×65 pixels was used for Levin dataset, while a support of 75×75 pixels was used for Lai and COCO 2017 datasets.

For quantitative comparison, PNSR and SSIM [73] quality measures were used. To avoid the inherent shifting ambiguity in blind deconvolution [17], we converted the images to the YCbCr color space and calculated both measures in the central part of the Y band of the deconvolved images, trimming 25 pixels from each side for the Levin and Sun datasets and 37 pixels for the rest of the datasets. This trimmed image was shifted in a window of size

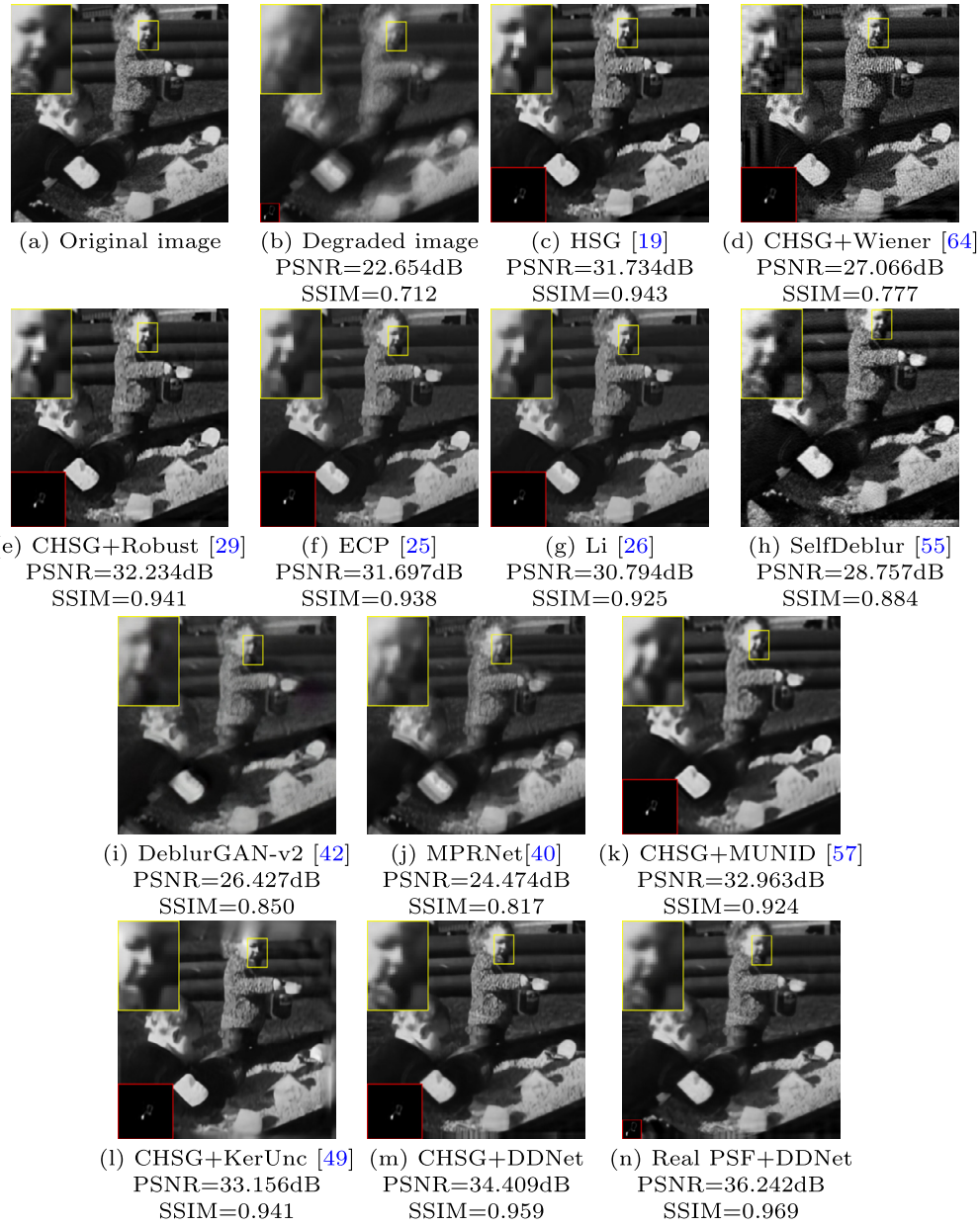


Fig. 4. Visual comparison of the proposed DDNet and competing methods on an image of the Levin [71] dataset.

25×25 or 37×37 with a precision of a quarter of pixel to find the position returning the best PSNR with respect to the original image. PSNR and SSIM results are reported for this position. The resulting figures of merit are summarized in Table 3.

6.2. Results

From Table 3, it is clear that the proposed DDNet method outperforms all competing methods. It is worth mentioning that CHSG + DDNet increases the PSNR up to 6.19 dB and the SSIM up to 0.196 with respect to the baseline method CHSG + Wiener. Also, notice that the results of pure DL models DeblurGAN-v2 [42] and MPRNet [40] are consistently worse than the other methods because these models were trained and designed for non-uniform motion blur. It is worth mentioning that the proposed CHSG + DDNet method scores are below the Real PSF + DDNet values, and that methods taking into account kernel inaccuracies usually obtain better results than those assuming a perfect kernel estimation. This demonstrates that the proposed method is robust to inaccuracies but that there is room for improvement both in kernel estimation and non-blind image deconvolution.

Fig. 4 shows the original, observed and restored images with the competing and the proposed method on an image of the Levin dataset. The insets show a detail of the area marked in yellow and the used PSF, when applicable. Notice that, due to the displacement introduced by the blur kernel, some border artifacts can be observed in most of the reconstructed images. We have opted not to crop the images to remove these artifacts to ease the comparison. Although all compared methods produce good results on this image, most of them present artifacts that reduce their quality. Those artifacts include ringing artifacts (see Fig. 4(d)), blurriness (Fig. 4(i) and (j)), excessive smoothness (Fig. 4(c), (g) and (k)), phantoms (see the area of the elbow patch in Fig. 4(f) and (g)), excessive contrast (Fig. 4(h), (i) and (l)) or washed out details (see, for instance, the jumper in Fig. 4(e), (f), and (g)). In contrast with competing methods, the proposed method is able to recover the small details removing all the ringing and most other artifacts

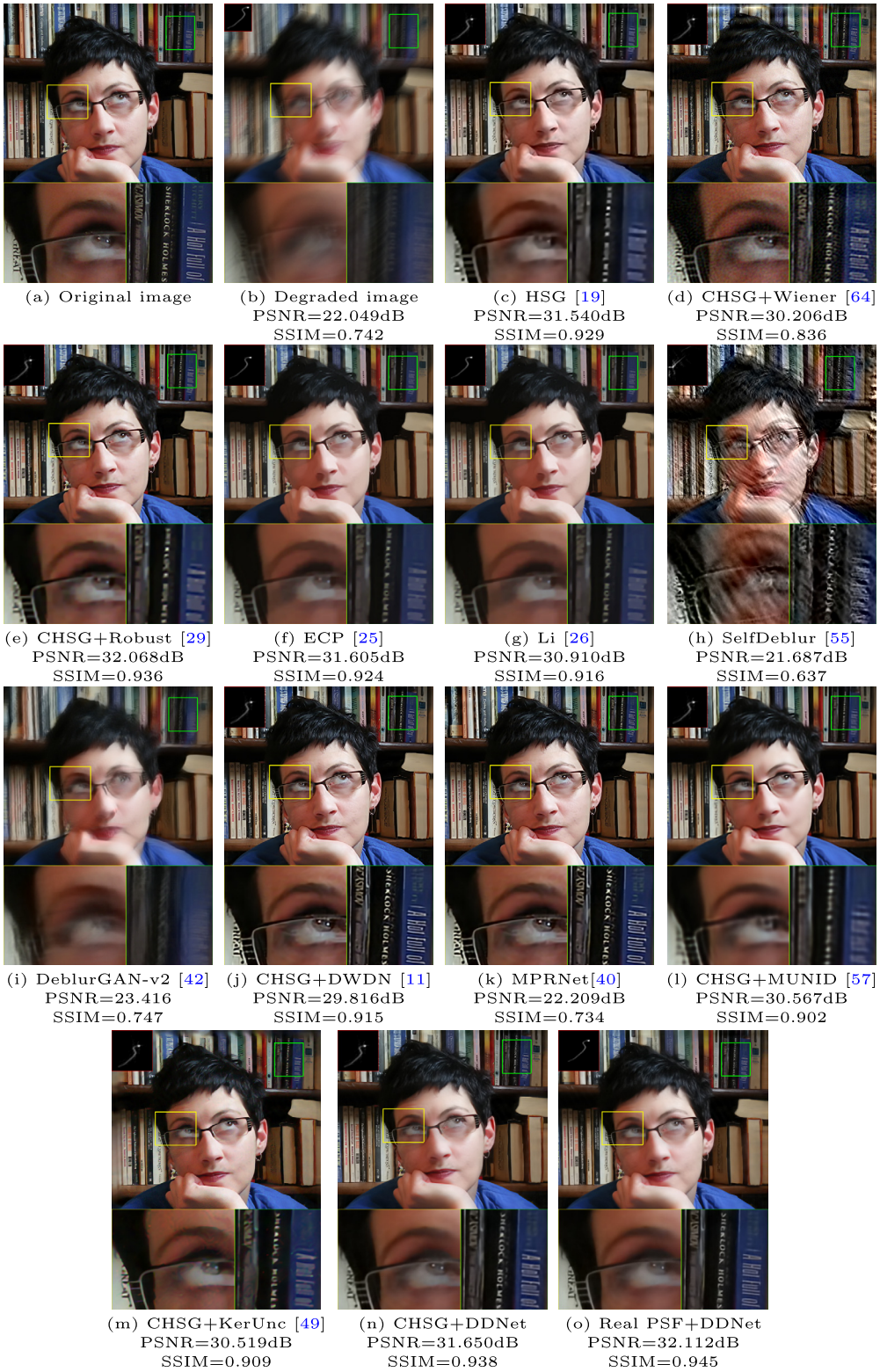
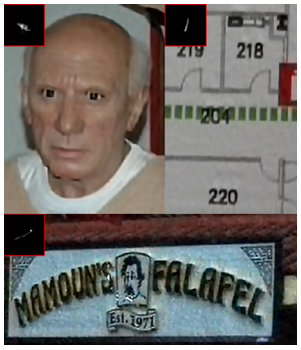


Fig. 5. Visual comparison of the proposed DDNet and competing methods on an image of the Lai [72] degraded dataset.

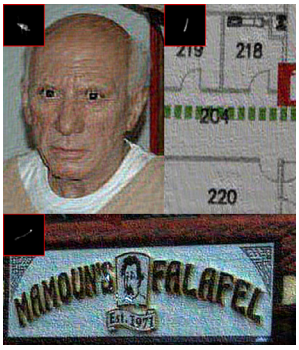
providing a result similar to the one obtained using the real blur kernel.

We also present results on an image of the Lai [72] spatially invariant degraded dataset. The original image is shown in Fig. 5(a), the degraded image in Fig. 5(b), and Fig. 5(c)-(o) depict the deconvolved images with the competing and DDNet method. From these images, it is clear that CHSG + DDNet produces better visual results than the competing ones. As for the Levin dataset, the methods which do not estimate the blur (DeblurGAN-v2 and MPRNet) obtain blurred images (see Fig. 5(i) and (k)). The baseline method CHSG + Wiener and SelfDeblur produce results, depicted in Fig. 5(d) and (h), respectively, with excessive ringing and other artifacts. The HSG (Fig. 5(c)), CHSG + Robust (Fig. 5(e)), CHSG + MUNID (Fig. 5(l)) and Li (Fig. 5(g)) methods produce good but

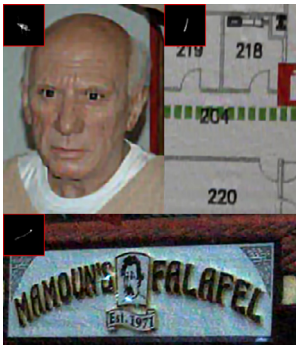
(a) Degraded image



(b) HSG [19]



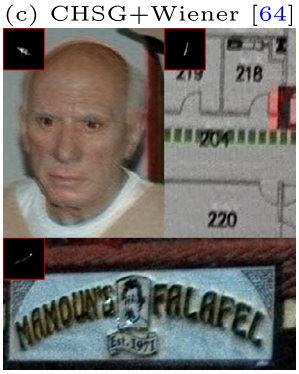
(c) CHSG+Wiener [64]



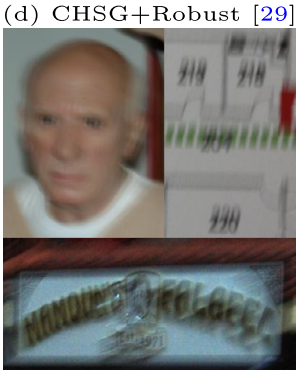
(d) CHSG+Robust [29]

(e) ECP [25]

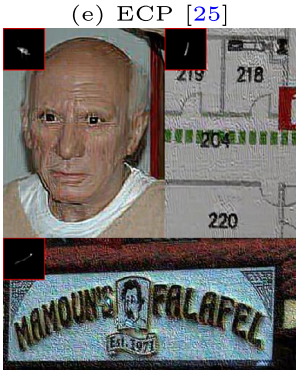
(f) Li [26]



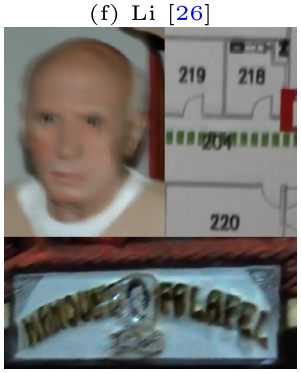
(g) SelfDeblur [55]



(h) DeblurGAN-v2 [42]



(i) CHSG+DWDN [11]



(j) MPRNet[40]

(k) CHSG+MUNID [57]

(l) CHSG+KerUnc [49]

(m) CHSG+DDNet

Fig. 6. Visual comparison of the proposed DDNet and competing methods on real images.

oversmoothed images, while CHSG + DWDN (Fig. 5(j)) oversharpens the resulting image. CHSG + NBD_KerUnc (Fig. 5(m)) produces a sharp image but with some color artifacts around edges. For this particular image, the Robust and ECP methods (see Fig. 5(e) and Fig. 5(f)) produce images close to the proposed one, in Fig. 5(n), but the small details (see, for instance, the title of the books) are not so well resolved. The DDNet method can recover small details without any of the artifacts of the CHSG + Wiener restoration. Note also that the result using CHSG + DDNet is very similar to the obtained using the real PSF.

Fig. 6 depicts three details of challenging real images as well as the deconvolutions with the proposed and competing methods. The DDNet method provides almost artifact-free images and, compared with other methods, more consistent results. When compared to CHSG + Wiener, it is clear that the proposed method can remove most restoration artifacts while maintaining the details in

the image. The CHSG + DDNet method produces sharper images than HSG, ECP, Li, DeblurGAN-v2, CHSG + MUNID, and MPRNet methods, and fewer artifacts than SelfDeblur, CHSG + DWDN and CHSG + Robust (see, the color artifacts in Picasso's face and hair, for instance) and it produces less noisy images (see the roof over the restaurant sign), without artifacts in flat regions and more natural looking image (see, Picasso's cheek and T-shirt). In general, the proposed DDNet method is robust to different categories of images and blurs and provides good, natural-looking, almost artifact-free images.

6.3. Computation cost and efficiency

All the experiments were run on an Intel Xeon E5-2630v@2.2 GHz CPU with a GeForce RTX 2080 Ti GPU. As shown in Table 3, the DDNet method achieves the third lowest total time, behind DeblurGAN-v2 [42] and MPRNet [40], but with a much higher restoration quality. Note that most of the total time is spent in the estimation of the blur kernel, which is carried out in the CPU, while the CNN cost is a few hundredths of a second. Compared with other non-blind image deconvolution methods, the DDNet method is the fastest one, even faster than Wiener method running on CPU. It is also much faster than other DL based methods such as MUNID [57] or KerUnc [49]. Note also that the proposed method is faster than DWDN [11], which applies the Wiener filter in the feature space and uses a significantly larger network (with almost twice the number of parameters than DDNet).

7. Conclusions

We have proposed a combination of an analytical and a DL method for robust non-blind image deconvolution. To improve the DL model's generalization to new blurs, we have separated the network parameters' learning from the image formation model as much as possible. This has allowed the deal with (possibly inaccurate) estimates of the blur. We have shown that the Wiener filter can be used to approximate the blur convolution operator's pseudo-inverse. Non-blind image deconvolution is cast as the learning of a residual in the null space of the kernel which, added to the Wiener restoration, will satisfy the image formation model. Spatially variant artifacts caused by an inexact PSF estimation or other degradation model inconsistencies are corrected using the spatially-variant filters produced by a DFN. As shown by our ablation experiments, this approach outperforms similar CNN models twice as deep. The proposed DDNet method outperforms the rest of the compared methods, generating good, natural-looking, nearly artifact-free images. It has proven to be robust, always providing one of the best results (on average) in a great variety of images. The DDNet method is faster than all other analytical methods and the combined analytical and DL method. Extension of the proposed method to deal with spatially-variant blur will be considered in future work.

CRediT authorship contribution statement

Santiago López-Tapia: Conceptualization, Formal analysis, Investigation, Methodology, Software, Writing – original draft. **Javier Mateos:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Writing – original draft. **Rafael Molina:** Conceptualization, Funding acquisition, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Aggelos K. Katsaggelos:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] A. Lucas, M. Iliadis, R. Molina, A.K. Katsaggelos, Using deep neural networks for inverse problems in imaging, IEEE Signal Process. Mag. 35 (1) (2018) 20–36.
[2] V. Jain, S. Sebastian, Natural image denoising with convolutional networks, in: D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (Eds.), Advances in Neural Information Processing Systems, vol. 21, Curran Associates, Inc., 2009, pp. 769–776.
[3] J. Xie, L. Xu, E. Chen, Image denoising and inpainting with deep neural networks, in: Conference on Neural Information Processing Systems, 2012, pp. 1–9.
[4] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2016) 295–307, https://doi.org/10.1109/TPAMI.2015.2439281.
[5] L. Xu, J.S. Ren, C. Liu, J. Jia, Deep convolutional neural network for image deconvolution, in: Advances in Neural Information Processing Systems, vol. 27, 2014, pp. 1790–1798.
[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (2011) 1–122.
[7] D. Geman, C. Yang, Nonlinear image recovery with half-quadratic regularization, https://doi.org/10.1109/83.392335, 1995.
[8] M.V. Afonso, J.M. Bioucas-Dias, M.A.T. Figueiredo, Fast image recovery using variable splitting and constrained optimization, IEEE Trans. Image Process. 19 (9 (3)) (2010) 2345–2356, https://doi.org/10.1109/TIP.2010.2047910.
[9] F. Sureau, A. Lechat, J.-L. Starck, Deep learning for a space-variant deconvolution in galaxy surveys, Astron. Astrophys. 641 (2020) A67, https://doi.org/10.1051/0004-6361/201937039.
[10] S. Vasu, V.R. Maligireddy, A.N. Rajagopalan, Non-blind deblurring: handling kernel uncertainty with CNNs, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 3272–3281.
[11] J. Dong, S. Roth, B. Schiele, DWDN: deep Wiener deconvolution network for non-blind image deblurring, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2022) 9960–9976, https://doi.org/10.1109/TPAMI.2021.3138787.
[12] X. Jia, B. De Brabandere, T. Tuytelaars, L.V. Gool, Dynamic filter networks, in: D.D. Lee, M. Sugiyama, U.V. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 29, 2016, pp. 667–675.
[13] P. Ruiz, X. Zhou, J. Mateos, R. Molina, A.K. Katsaggelos, Variational Bayesian blind image deconvolution: a review, Digit. Signal Process. 47 (2015) 116–127, https://doi.org/10.1016/j.dsp.2015.04.012.
[14] X. Zhou, J. Mateos, F. Zhou, R. Molina, A.K. Katsaggelos, Variational Dirichlet blur kernel estimation, IEEE Trans. Image Process. 24 (2015) 5127–5139.
[15] D. Krishnan, R. Fergus, Fast image deconvolution using hyper-Laplacian priors, in: Conference on Neural Information Processing Systems, 2009.
[16] B. Zhang, R. Liu, H. Li, Q. Yuan, X. Fan, Z. Luo, Blind image deblurring using adaptive priors, in: Internet Multimedia Computing and Service, Communications in Computer and Information Science, Springer, Singapore, 2017, pp. 13–22.
[17] D. Perrone, P. Favaro, A logarithmic image prior for blind deconvolution, Int. J. Comput. Vis. 117 (2) (2016) 159–172.
[18] R. Fergus, B. Singh, A. Hertzmann, S.T. Roweis, W.T. Freeman, Removing camera shake from a single photograph, ACM Trans. Graph. 25 (3) (2006).
[19] X. Zhou, M. Vega, F. Zhou, R. Molina, A.K. Katsaggelos, Fast Bayesian blind deconvolution with Huber super Gaussian priors, Digit. Signal Process. 60 (2017) 122–133.
[20] S.D. Babacan, R. Molina, M.N. Do, A.K. Katsaggelos, Bayesian blind deconvolution with general sparse image priors, in: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (Eds.), Computer Vision – European Conference on Computer Vision 2012, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 341–355.
[21] Y. Quan, H. Ji, Z. Shen, Data-driven multi-scale non-local wavelet frame construction and image recovery, J. Sci. Comput. 63 (2015) 307–329.
[22] L. Chen, Q. Sun, F. Wang, Adaptive blind deconvolution using generalized cross-validation with generalized lp/lq norm regularization, Neurocomputing 399 (2020) 75–85.
[23] L. Xu, S. Zheng, J. Jia, Unnatural L0 sparse representation for natural image deblurring, in: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, CVPR'13, IEEE Computer Society, Washington, DC, USA, 2013, pp. 1107–1114.

[24] J. Pan, D. Sun, H. Pfister, M. Yang, Blind image deblurring using dark channel prior, in: 2016 IEEE Conf. Comput. Vision and Pattern Recognition, 2016, pp. 1628–1636.
[25] Y. Yan, W. Ren, Y. Guo, R. Wang, X. Cao, Image deblurring via extreme channels prior, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6978–6986.
[26] L. Li, J. Pan, W. Lai, C. Gao, N. Sang, M. Yang, Learning a discriminative prior for blind image deblurring, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 6616–6625.
[27] Q. Shan, J. Jia, A. Agarwala, High-quality motion deblurring from a single image, ACM Trans. Graph. 27 (3) (2008).
[28] N. Galatsanos, V. Mesarovic, R. Molina, A. Katsaggelos, J. Mateos, Hyperparameter estimation in image restoration problems with partially-known blurs, Opt. Eng. 41 (8) (2002) 1845–1854.
[29] H. Ji, K. Wang, Robust image deblurring with an inaccurate blur kernel, IEEE Trans. Image Process. 21 (4) (2012) 1624–1634.
[30] D. Ren, W. Zuo, D. Zhang, L. Zhang, M.-H. Yang, Simultaneous fidelity and regularization learning for image restoration, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2021) 284–299, https://doi.org/10.1109/TPAMI.2019.2926357.
[31] D. Ren, W. Zuo, D. Zhang, J. Xu, L. Zhang, Partial deconvolution with inaccurate blur kernel, IEEE Trans. Image Process. 27 (1) (2018) 511–524, https://doi.org/10.1109/TIP.2017.2764261.
[32] J. Han, S. Zhang, Z. Ye, A non-blind deconvolution method by bias-correction for inaccurate blur kernel estimation in image deblurring, IEEE Trans. Geosci. Remote Sens. (2022), https://doi.org/10.1109/TGRS.2022.3207828.
[33] J. Han, S. Zhang, Z. Ye, A nonblind deconvolution method by bias correction for inaccurate blur kernel estimation in image deblurring, IEEE Trans. Geosci. Remote Sens. 61 (2023) 1–14, https://doi.org/10.1109/TGRS.2022.3207828.
[34] J. Kotera, V. Šmídl, F. Šroubek, Blind deconvolution with model discrepancies, IEEE Trans. Image Process. 26 (5) (2017) 2533–2544, https://doi.org/10.1109/TIP.2017.2676981.
[35] X. Chen, Y. Zhu, J. Sun, Y. Zhang, Robust motion blur kernel estimation by kernel continuity prior, IEEE Access 8 (2020) 46162–46175, https://doi.org/10.1109/ACCESS.2019.2961597.
[36] M. Chang, C. Yang, H. Feng, Z. Xu, Q. Li, Beyond camera motion blur removing: how to handle outliers in deblurring, IEEE Trans. Comput. Imaging 7 (2021) 463–474, https://doi.org/10.1109/TCI.2021.3076886.
[37] L. Chen, J. Zhang, S. Lin, F. Fang, J.S. Ren, Blind deblurring for saturated images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 6308–6316.
[38] J. Koh, J. Lee, S. Yoon, Single-image deblurring with neural networks: a comparative survey, Comput. Vis. Image Underst. 203 (2021) 103134, https://doi.org/10.1016/j.cviu.2020.103134.
[39] S. Nah, T.H. Kim, K.M. Lee, Deep multi-scale convolutional neural network for dynamic scene deblurring, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 257–265.
[40] S.W. Zamir, A. Arora, S. Khan, M. Hayat, F.S. Khan, M.-H. Yang, L. Shao, Multi-stage progressive image restoration, in: CVPR, 2021.
[41] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, J. Matas, DeblurGAN: blind motion deblurring using conditional adversarial networks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 8183–8192.
[42] O. Kupyn, T. Martyniuk, J. Wu, Z. Wang, DeblurGAN-v2: deblurring (orders-of-magnitude) faster and better, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 8877–8886.
[43] Y. Li, M. Tofighi, J. Geng, V. Monga, Y.C. Eldar, Efficient and interpretable deep blind image deblurring via algorithm unrolling, IEEE Trans. Comput. Imaging 6 (2020) 666–681.
[44] T. Meinhardt, M. Möller, C. Hazirbas, D. Cremers, Learning proximal operators: using denoising networks for regularizing inverse imaging problems, in: IEEE International Conference on Computer Vision, ICCV, 2017, pp. 1799–1808.
[45] K. Zhang, W. Zuo, S. Gu, L. Zhang, Learning deep CNN denoiser prior for image restoration, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3929–3938.
[46] N. Parikh, S. Boyd, Proximal algorithms, Found. Trends Optim. 1 (3) (2014) 127–239, https://doi.org/10.1561/2400000003.
[47] J.H.R. Chang, C. Li, B. Póczos, B.V.K.V. Kumar, One network to solve them all — solving linear inverse problems using deep projection models, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 5889–5898.
[48] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, X. Lu, Denoising prior driven deep neural network for image restoration, Computing Research Repository, arXiv: 1801.06756, 2018.
[49] Y. Nan, H. Ji, Deep learning for handling kernel/model uncertainty in image deconvolution, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
[50] D. Gong, Z. Zhang, Q. Shi, A. van den Hengel, C. Shen, Y. Zhang, Learning deep gradient descent optimization for image deconvolution, IEEE Trans. Neural Netw. Learn. Syst. (2020), https://doi.org/10.1109/TNNLS.2020.2968289, in press.
[51] Y. Quan, P. Lin, Y. Xu, Y. Nan, H. Ji, Nonblind image deblurring via deep learning in complex field, IEEE Trans. Neural Netw. Learn. Syst. (2022) 5387–5400, https://doi.org/10.1109/TNNLS.2021.3070596.
[52] L. Huang, Y. Xia, Joint blur kernel estimation and CNN for blind image restoration, Neurocomputing 396 (2020) 324–345.
[53] Z. Wang, Z. Wang, Q. Li, H. Bilen, Image deconvolution with deep image and kernel priors, in: The IEEE International Conference on Computer Vision (ICCV) Workshops, 2019.
[54] D. Ulyanov, A. Vedaldi, V. Lempitsky, Deep image prior, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 9446–9454.
[55] D. Ren, K. Zhang, Q. Wang, Q. Hu, W. Zuo, Neural blind deconvolution using deep priors, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 3338–3347.
[56] J. Kotera, F. Šroubek, V. Šmídl, Improving neural blind deconvolution, in: 2021 IEEE International Conference on Image Processing (ICIP), 2021, pp. 1954–1958.
[57] M. Chen, Y. Quan, T. Pang, E. Ji, Nonblind image deconvolution via leveraging model uncertainty in an untrained deep neural network, Int. J. Comput. Vis. 130 (2022) 1770–1789.
[58] S. Bigdeli, D. Honzátko, S. Süsstrunk, L.A. Dunbar, Image restoration using plug-and-play CNN MAP denoisers, in: Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020), vol. 4, 2020, pp. 85–92.
[59] W. Niu, K. Zhang, W. Luo, Y. Zhong, H. Li, Deep robust image deblurring via blur distilling and information comparison in latent space, Neurocomputing 466 (2021) 69–79, https://doi.org/10.1016/j.neucom.2021.09.019.
[60] J. Dong, J. Pan, Deep outlier handling for image deblurring, IEEE Trans. Image Process. 30 (2021) 1799–1811, https://doi.org/10.1109/TIP.2020.3048679.
[61] Chapter III: Geometric and Analytic Properties of the Moore-Penrose Pseudoinverse, A. Albert (Ed.), Regression and the Moore-Penrose Pseudoinverse, Mathematics in Science and Engineering, vol. 94, Elsevier, 1972, pp. 15–42.
[62] C.D. Meyer, Matrix Analysis and Applied Linear Algebra, 2010.
[63] C. Sonderby, J. Caballero, L. Theis, W. Shi, F. Huszar, Amortised MAP inference for image super-resolution, in: International Conference on Learning Representations, 2017.
[64] R. Gonzalez, R. Woods, S. Eddins, Digital Image Processing Using Matlab, Prentice Hall, 2003.
[65] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, C.C. Loy, Esrgan: enhanced super-resolution generative adversarial networks, in: The European Conference on Computer Vision Workshops (ECCVW), 2018.
[66] W. Shi, J. Caballero, F. Huszár, J. Totz, A.P. Aitken, R. Bishop, D. Rueckert, Z. Wang, Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1874–1883.
[67] G. Boracchi, A. Foi, Modeling the performance of image restoration from motion blur, IEEE Trans. Image Process. 21 (8) (2012) 3502–3517.
[68] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv:1412.6980, 2014.
[69] W.-S. Lai, J.-B. Huang, N. Ahuja, M.-H. Yang, Deep Laplacian pyramid networks for fast and accurate super-resolution, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[70] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: an imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., 2019, pp. 8024–8035.
[71] A. Levin, Y. Weiss, F. Durand, W.T. Freeman, Understanding and evaluating blind deconvolution algorithms, in: Computer Vision and Pattern Recognition, 2009.
[72] W.-S. Lai, J.-B. Huang, Z. Hu, N. Ahuja, M.-H. Yang, A comparative study for single image blind deblurring, in: 2016 IEEE Conf. Comput. Vision and Pattern Recognition (CVPR), 2016, pp. 1701–1709.
[73] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612, https://doi.org/10.1109/TIP.2003.819861.

Santiago López-Tapia received the bachelor's, master's, and Ph.D.'s degrees in computer science from the University of Granada in 2014, 2015 and 2021, respectively. He is a postdoc in the Department of Electrical Engineering and Computer Science at Northwestern University. His research mainly focuses on using deep learning models for image restoration and classification.

Javier Mateos received the degree in computer science in 1991 and the Ph.D. degree in computer science in 1998, both from the University of Granada. He joined the Department of Computer Science and Artificial Intelligence, University of Granada, in 1992 where he is currently a Professor. He is conducting research on image and video processing, including image restoration, image and video recovery, super-resolution from (compressed) stills and video sequences, pansharpening and image classification. He served as an editor of "Digital Signal Processing" (2011–2021)

and as an Associate Editor for the "IEEE Transactions on Image Processing" (2014–2017).

Rafael Molina received the degree in mathematics and the Ph.D. degree in optimal design in linear models from the University of Granada, Granada, Spain, in 1979 and 1983, respectively. He was the Dean of the Computer Engineering School, University of Granada, from 1992 to 2002. In 2000, he joined the University of Granada, as a Professor of computer science and artificial intelligence. He was the Head of the Computer Science and Artificial Intelligence Department, University of Granada, from 2005 to 2007. His research focuses on using Bayesian modeling and inference in problems like image restoration, active learning, and machine learning.

Aggelos K. Katsaggelos received the Diploma degree in electrical and mechanical engineering from the Aristotelian University of Thessaloniki, Greece, in 1979, and the M.S. and Ph.D. degrees in electrical engineering from the Georgia Institute of Technology in 1981 and 1985, respectively. In 1985, he joined the Department of Electrical Engineering and Computer Science, Northwestern University, where he is currently a professor. He was the Ameritech Chair of information technology and the AT&T Chair, and he is the Joseph Cummings Chair. He has published extensively in the areas of signal processing and communications, computational imaging, and machine learning (over 300 journal papers).