

Microscopy Image Restoration with Deep Wiener-Kolmogorov Filters

Valeriya Pronina¹, Filippos Kokkinos², Dmitry V. Dylov¹, and Stamatios Lefkimmiatis³

¹ Skolkovo Institute of Science and Technology, Moscow, Russia

² University College London

³ Q.bio Inc.

Abstract. Microscopy is a powerful visualization tool in biology, enabling the study of cells, tissues, and the fundamental biological processes; yet, the observed images typically suffer from blur and background noise. In this work, we propose a unifying framework of algorithms for Gaussian image deblurring and denoising. These algorithms are based on deep learning techniques for the design of learnable regularizers integrated into the Wiener-Kolmogorov filter. Our extensive experimentation line showcases that the proposed approach achieves a superior quality of image reconstruction and surpasses the solutions that rely either on deep learning or on optimization schemes alone. Augmented with the variance stabilizing transformation, the proposed reconstruction pipeline can also be successfully applied to the problem of Poisson image deblurring, surpassing the state-of-the-art methods. Moreover, several variants of the proposed framework demonstrate competitive performance at low computational complexity, which is of high importance for real-time imaging applications.

Keywords: deblurring, denoising, learnable regularizers, microscopy deblurring, Wiener filter.

1 Introduction

Microscopes are widely used in biological and medical research, allowing the study of organic and inorganic substances at the minuscule scale. The observed microscopy images, however, suffer from the following two inherent distortions: a blur of detail caused by the resolution limit of a microscope, and a background noise introduced by the imperfections of the imaging system as a whole and by the image-recording sensor in particular. Both of these traits not only distort the perception of the detail in the image but also influence the quantitative analysis of its content [14,33].

Mathematically, the image formation process can be described by the operation of convolution, where the underlying image is convolved with the point spread function of the microscope [17,37]. Two fluorescence microscopy imaging systems – widefield and confocal – are very popular among biologists. In the

case of widefield microscopy, the entire specimen is exposed to a uniform light source. Here, the intensity of the illuminating light is high and the noise statistics of the recorded image can be approximated by the Gaussian distribution [45]. On the contrary, the confocal microscopes rely on point-by-point imaging of the specimen thanks to an aperture (pin-hole) installed in the microscope’s optical system to block the out-of-focus signal. This improves the resolution but limits the numerical aperture of the microscope, effectively reducing the number of photons captured by the imaging detector. A more appropriate approximation for the noise in such low-photon images is the Poisson distribution [45].

Restoration of microscopy images is an ill-posed inverse problem where a unique solution does not exist [17]. One, therefore, needs to constrain the space of solutions in order to obtain a statistically or a physically meaningful one. A popular approach for doing that follows the variational formulation of the problem, where the restored image is obtained as the minimizer of an objective function [3]. This function comprises two terms: the data fidelity term that measures the proximity between the obtained measurements and the solution, and the regularization term that integrates prior information about the expected solution. There are several methods focused on the development of an effective regularization scheme for the image restoration, e.g., the Hyper-Laplacian priors [22], the non-local means [11], the shrinkage fields [36], and others.

With the advent of deep learning, many inverse problems have been successfully approached with Fully Convolutional Neural Networks (FCNNs) [32,39] which can learn a mapping between the measured image and its expected reconstruction. Furthermore, a series of works have incorporated deep learning for the regularization purposes in a wide range of image restoration problems [47,24,25,48,23,21]. Such a regularization paradigm, in turn, has allowed to surpass the performance of the one-shot methods relying on FCNN; however, none of these techniques have been tested on microscopy related deblurring problems.

In this work, we present a joint denoising and deblurring framework for microscopy images, which comprises a collection of methods that leverage the advantages of both the classical schemes for optimization, and the deep learning approaches for regularization. We develop an extensive set of techniques for handling the image prior information, by using both shallow and deep learning for parametrization. The proposed framework entails the following steps:

- First, a regularizer is formed as a group of learnable kernels and is deployed to every image identically.
- Second, an intuitive extension is further examined with the group of kernels being predicted *per image* using a compact FCNN as a Kernel Prediction Network (KPN).
- The same kernel-predicting approach is then probed in *per pixel* evaluation, with the KPN predicting the appropriate regularizer for each spatial location in the image.
- The last step consists of approximating the entire regularization function with a neural network, which is then employed in an iterative manner.

The developed methods outperform solutions based either on optimization schemes alone or the solutions based merely on deep learning techniques. Ex-

exploiting both the classical optimization and the deep learning methodologies, the proposed approaches are intrinsically ready for fine-tuning the trade-off between the computational efficiency and the accuracy of image reconstruction.

2 Related Work

Deconvolution in the presence of noise is generally considered a challenging task, and it has attracted significant attention by the research community. The simplest deconvolution approach consists in estimation of the maximum likelihood under the assumption of the statistical model of the observed data. The main drawback of this approach is the amplification of the measurement noise. One common way to avoid this hurdle is to add the regularization term to penalize the values of the solution.

One of the most well-known classical methods for image deblurring and denoising, the Wiener-Kolmogorov filter, or Wiener filter [44,41], is derived as a maximum likelihood estimator under the assumption of Gaussian noise. It uses the Tikhonov regularization functional to incorporate prior knowledge about the expected solution. Wiener filter also retrieves the minimum mean squared error estimate and it has established itself as a fast deconvolution algorithm [5,10,49] due to the closed-form solution in the Fourier domain. More recently, the Wiener filter became a part of methods such as PURE-LET [27] and SURE-LET [49], performing deconvolution with a piece-wise thresholding under wavelet coefficients regularization.

Another well-known method for image deblurring is the Richardson-Lucy algorithm, which is derived as a maximum-likelihood estimator under the assumption of Poisson noise [34,29]. Richardson-Lucy algorithm is a common method for tackling problems of image deblurring and denoising in microscopy and it is mostly used with Tikhonov regularization and total-variation regularization [15].

In order to drive the solution of the optimization problem towards a subset of physically plausible image reconstructions, many researchers have resorted to hand-crafted regularization schemes. For example, Krishnan *et al.* in [22] propose to use hyper-Laplacian penalty functions, while in [7] a method using non-local regularization constraint is proposed for image deblurring and denoising.

With the advent of deep learning, nearly all image restoration methods were revisited from a learnable perspective with great success. Being widely used in many research areas, including the biomedical field, a plethora of neural network approaches have been proposed for denoising [32], demosaicking [20], and super-resolution [39]. Deep learning methods are also applied in image deblurring, e.g., in [16], where a CNN is used for restoring images corrupted with various visual artifacts, and in [13], where a parametric CNN model was used to enhance a shape-based artifact elimination. Xu *et al.* in [46] proposed a CNN that learns the deconvolution operation for natural images in a supervised manner. A step into combining traditional optimization schemes with deep learning is made in [36] by using iterative FFT-based deconvolution with learnable regularization filters, weights and shrinkage functions. A similar approach was introduced

by [47], where the authors proposed to learn horizontal and vertical image gradient filters. Zhang *et al.* in [48] developed a trainable denoiser prior that is then integrated into a model-based optimization method. Finally, in [9] the authors propose to learn a prior that represents a Gaussian-smoothed version of the natural image distribution.

3 Problem Formulation

3.1 Image Formation in Microscopy

The image formation process can be described by the observation model

$$\mathbf{y} = \mathbf{K}\mathbf{x} + \mathbf{n}, \quad (1)$$

where $\mathbf{y} \in \mathbb{R}^N$ corresponds to the observed image, $\mathbf{K} \in \mathbb{R}^{N \times N}$ is the matrix corresponding to the point spread function (PSF), $\mathbf{x} \in \mathbb{R}^N$ is the underlying image that we aim to restore and $\mathbf{n} \sim \mathcal{N}(0, \sigma^2)$ denotes noise, which is assumed to follow i.i.d Gaussian distribution. While \mathbf{x} and \mathbf{y} are two dimensional images, for the sake of mathematical derivations, we assume that they have been raster scanned using a lexicographical order, and they correspond to vectors of N dimensions.

3.2 Regularization

Deconvolution being an image restoration task is an ill-posed inverse problem, which implies that a unique solution does not exist. In general, such problems can be addressed following a variational approach whose solution $\hat{\mathbf{x}}$ can be obtained by minimizing an objective function of the form

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \underbrace{\frac{1}{2} \|\mathbf{y} - \mathbf{K}\mathbf{x}\|_2^2 + \lambda r(\mathbf{x})}_{\mathbf{J}(\mathbf{x})}. \quad (2)$$

Here the first term corresponds to the data fidelity term, which measures the proximity of the solution to the observation, while the second one corresponds to the regularizer that models any prior knowledge one might have about the ground-truth image. The parameter λ is a trade-off coefficient that determines the contribution of the regularizer into the estimation of the solution. A list of popular regularizers contains Tikhonov [40] and TV functionals [35] which have been widely used in a plethora of image restoration tasks, including deconvolution problems [15]. However, in this work we attempt to incorporate prior information learned directly from available training data in a supervised manner with the inclusion of deep learning strategies.

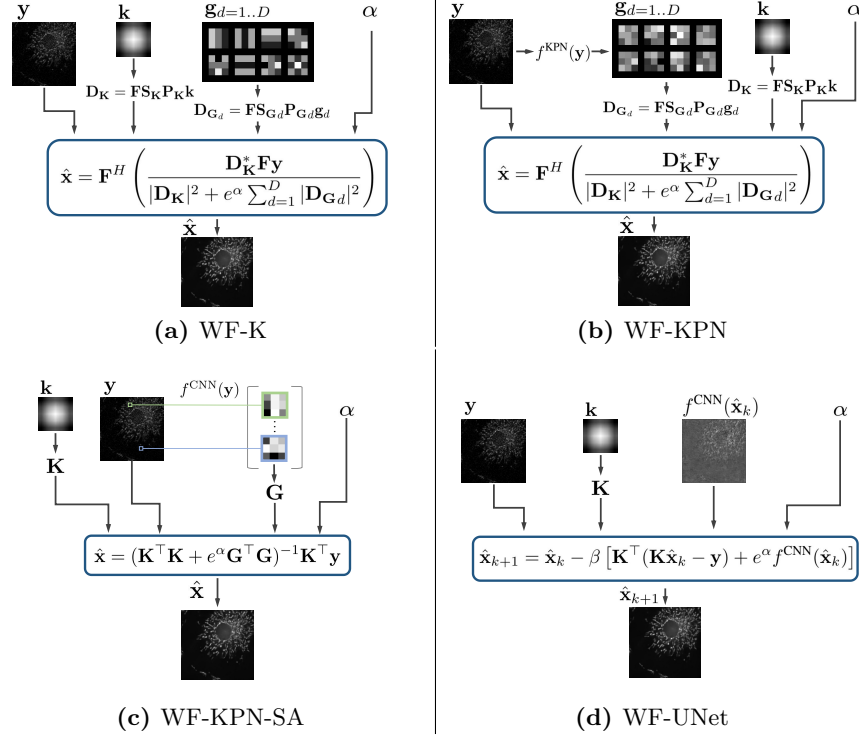


Fig. 1. Overview of the four proposed prior parametrization models. In Fig. 1a the one-shot Wiener filter with learnable kernels is depicted, while in 1b the same filter with predictable kernels is presented. In 1c the scheme incorporating prediction of per-pixel regularization kernels into Wiener filter is shown. The proposed iterative scheme which combines Wiener filtering with the approximation of regularization with a CNN is visualized in 1d

4 Proposed Approach

4.1 Learnable Regularization Kernels

First, we start from the observation that in most of the modern methods for image restoration the regularization term is formed as

$$r(\mathbf{x}) = \sum_{d=1}^D \rho_d(\mathbf{g}_d * \mathbf{x}), \quad (3)$$

where \mathbf{g}_d are typically linear filters and $\rho_d(\cdot)$ is a set of penalty functions, acting on the filters outputs [22]. In this work, we explicitly set the penalty function to be the squared ℓ_2 norm, leading to a Tikhonov regularizer [40]

$$r(\mathbf{x}) = \sum_{d=1}^D \|\mathbf{g}_d * \mathbf{x}\|_2^2 = \sum_{d=1}^D \|\mathbf{G}_d \mathbf{x}\|_2^2, \quad (4)$$

where \mathbf{g}_d are learnable convolution kernels and $\mathbf{G}_d \in \mathbb{R}^{N \times N}$ are convolution matrices, corresponding to these kernels. This specific choice of the penalty function allows us to obtain the solution of (2) in the closed-form

$$\hat{\mathbf{x}} = (\mathbf{K}^\top \mathbf{K} + \lambda \sum_{d=1}^D \mathbf{G}_d^\top \mathbf{G}_d)^{-1} \mathbf{K}^\top \mathbf{y}, \quad (5)$$

which corresponds to the Wiener-Kolmogorov filter. Here \mathbf{K}^\top and \mathbf{G}_d^\top are the adjoint matrices of \mathbf{K} and \mathbf{G}_d , respectively. In order to obtain the solution of Eq. (21) one has first to perform the inversion of a huge matrix which can be very slow in practice or even intractable. However, the supremacy of the Wiener filter lies in its FFT-based inference, which renders it a fast and efficient method and allows to restore the underlying signal at a low computational complexity. Specifically, under the assumption of periodic image boundary conditions, the degradation matrix \mathbf{K} and convolution matrix \mathbf{G}_d are circulant real matrices and thus, they can be diagonalized in the Fourier domain as

$$\mathbf{K} = \mathbf{F}^H \mathbf{D}_\mathbf{K} \mathbf{F}, \quad \mathbf{G}_d = \mathbf{F}^H \mathbf{D}_{\mathbf{G}_d} \mathbf{F} \quad (6)$$

$$\mathbf{D}_\mathbf{K} = \mathbf{F} \mathbf{S}_\mathbf{K} \mathbf{P}_\mathbf{K} \mathbf{k}, \quad \mathbf{D}_{\mathbf{G}_d} = \mathbf{F} \mathbf{S}_{\mathbf{G}_d} \mathbf{P}_{\mathbf{G}_d} \mathbf{g}_d. \quad (7)$$

Here $\mathbf{F} \in \mathbb{C}^{N \times N}$ is the Fourier (DFT) matrix, $\mathbf{F}^H \in \mathbb{C}^{N \times N}$ is its inverse, $\mathbf{D}_\mathbf{K}, \mathbf{D}_{\mathbf{G}_d} \in \mathbb{C}^{N \times N}$ are diagonal matrices, $\mathbf{S}_\mathbf{K}, \mathbf{S}_{\mathbf{G}_d} \in \mathbb{R}^{N \times N}$ are the corresponding circulant shift operators, $\mathbf{P}_\mathbf{K} \in \mathbb{R}^{N \times M}, \mathbf{P}_{\mathbf{G}_d} \in \mathbb{R}^{N \times Ld}$ are the corresponding zero-padding operators, $\mathbf{k} \in \mathbb{R}^M$ is the blurring kernel and $\mathbf{g}_d \in \mathbb{R}^{Ld}$ is a regularization convolution kernel. We also consider the trade-off coefficient λ to be equal to e^α in all experiments to ensure the positivity of the regularization weight. Based on the above, the one-shot solution of Eq. (21) is

$$\hat{\mathbf{x}} = \mathbf{F}^H \left(\frac{\mathbf{D}_\mathbf{K}^* \mathbf{F} \mathbf{y}}{|\mathbf{D}_\mathbf{K}|^2 + e^\alpha \sum_{d=1}^D |\mathbf{D}_{\mathbf{G}_d}|^2} \right), \quad (8)$$

where $\mathbf{D}_\mathbf{K}^*$ is the Hermitian transpose of $\mathbf{D}_\mathbf{K}$ and the division is applied in an element-wise fashion. The method is depicted in Fig. 1a and hereafter we refer to it as WF-K. To obtain the solution in this form, we firstly employ a group of D learnable kernels \mathbf{g}_d of size $K \times K$ which are initialized using a two-dimensional discrete cosine transform (DCT) frequency basis; a common choice in image processing since it has shown the ability to extract useful prior image information. We also consider α to be a learnable parameter in all proposed regularization schemes in order for the trade-off coefficient value to be tuned alongside the learnable kernels during the training process.

4.2 Prediction of Regularization Kernels

A critical drawback of the standard Wiener method lies in the way the group of kernels is formulated. In detail, the learnable group is global which means that

all kernels are applied on every image identically without any dependency on the underlying content. To remedy this situation, we propose the content-driven WF-KPN method that predicts per image the group of kernels that need to be used for regularization. In this framework, the solution of a Wiener filter yields the same form as presented in Eq. (1), but unlike the previous approach, the convolution kernels \mathbf{g}_d , that form the diagonal matrix $\mathbf{D}_{\mathbf{G}_d}$ in Eq. (7), are now predicted from a Kernel Prediction Network (KPN) [32] as shown in Fig. 1b.

For the Kernel Prediction Network we select a compact customized UNet architecture with nearly 470k parameters that receives the distorted input \mathbf{y} and produces an output with $K^2 D$ channels and the same spatial resolution as the input. Unlike the traditional UNet, we perform global average pooling in the spatial dimension on the output of the last layer, which is then reshaped into a stack of $D \times K \times K$ regularization kernels. Here D is the number of kernels and $K \times K$ is the support size. In all our experiments, we set $D = 8$ and $K = 3$. We also use instance normalization [42] after each convolutional layer to normalize weights for each image independently. In this way, the KPN predicts D content dependent regularization kernels of size $K \times K$ for each image, that then under the assumption of periodic boundary conditions form the diagonal matrix $\mathbf{D}_{\mathbf{G}_d}$ in Eq. (7). After that, the resulting diagonal matrices $\mathbf{D}_{\mathbf{G}_d}$ are incorporated into the solution of the Wiener filter according to Eq. (1).

The proposed models WF-K and WF-KPN involve learning of the trade-off coefficient α and regularization kernels \mathbf{g}_d in a supervised manner via means of back-propagation of a loss function. While the loss function we are minimizing during training is real-valued, the solution in Eq. (1) involves complex quantities and therefore we cannot perform back-propagation by solely relying on layers currently available in existing deep-learning libraries. For this reason we have implemented our own customized layers which depend on the analytical derivations of the gradients of the solution $\hat{\mathbf{x}}$ w.r.t. the trainable parameters. All analytical derivations of the necessary gradients are provided in the supplementary material.

4.3 Prediction of Spatially Adaptive Regularization Kernels

While the prediction of global regularization kernels per image provides the appealing property of content adaptation, we further extend WF-KPN to be both spatially and content adaptive. This is achieved by predicting for each image a different regularization kernel per-pixel and hence the method is dubbed as WF-KPN-SA. For this extension, we modify the UNet of Section 4.2 to predict a kernel per spatial location of an image. Unlike WF-KPN, the output of the network has K^2 channels and the same size as the input. We empirically found that normalization of the network hinders the performance and therefore it was removed in all WF-KPN-SA related experiments. Furthermore, the output of the network is reshaped into spatially adaptive regularization kernels of size $K \times K$ for each pixel of an input image. These kernels are then unfolded into a matrix \mathbf{G} which now does not correspond to a circulant matrix, and, thus, the FFT-based inference of the Wiener-Kolmogorov filter is not feasible. The solution of

the restoration problem has now the form of

$$\hat{\mathbf{x}} = (\mathbf{K}^\top \mathbf{K} + e^\alpha \mathbf{G}^\top \mathbf{G})^{-1} \mathbf{K}^\top \mathbf{y} \quad (9)$$

and the calculation of $\hat{\mathbf{x}}$ is done using the conjugate gradient algorithm [38]. To the best of our knowledge, WF-KPN-SA is the first method that predicts a spatially varying regularizer per image for inverse imaging related tasks.

4.4 Prediction of the Gradient of the Regularizer

Finally, with the desire to fully exploit deep learning capabilities, we employ a CNN for parametrizing a prior that would be specific for each image and, thus, completely content adaptive. This way we do not make assumptions about the form of the regularization term as we did above, and thus, the solution of Eq. (2) cannot be derived anymore in a closed-form expression. One common way to solve Eq. (2) in this case is to apply a gradient descent [8] optimization algorithm to find a minimizer of the objective function $\mathbf{J}(\mathbf{x})$,

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_k - \beta \nabla \mathbf{J}(\mathbf{x}). \quad (10)$$

Here $\hat{\mathbf{x}}_{k+1}$ is the solution of Eq. (2) that is updated after each iteration k of the gradient descent scheme and β is the learning rate defining the speed of the algorithm. Introducing the objective function $\mathbf{J}(\mathbf{x})$, defined according to Eq. (2), into Eq. (10), we obtain the solution of the gradient descent scheme,

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_k - \beta [\mathbf{K}^\top (\mathbf{K} \hat{\mathbf{x}}_k - \mathbf{y}) + e^\alpha f^{\text{CNN}}(\hat{\mathbf{x}}_k)]. \quad (11)$$

Here we parametrize the gradient of the regularizer with the CNN, $\nabla \mathbf{r}(\hat{\mathbf{x}}_k) = f^{\text{CNN}}(\hat{\mathbf{x}}_k)$. For the prediction of the gradient of a regularizer we employ the UNet architecture that was defined in Section 4.2. Unlike WF-KPN and WF-KPN-SA, we do not modify the original UNet architecture, and in this case the network receives a distorted image \mathbf{y} as an input and maps it to an output with the same resolution and number of channels as the input. We also use instance normalization after each convolutional layer to normalize weights for each image independently. With a slight abuse of notation, the method is hereafter referred as WF-UNet and it is depicted in Fig. 1d. The number of gradient descent iterations for WF-UNet is set to be equal to 10. We also consider the step size β to be a learnable parameter in order for the gradient descent speed to be tuned during the training process.

We stress that the proposed methods, WF-KPN, WF-KPN-SA and WF-UNet, share a UNet with almost identical architecture in order to have approximately the same number of trainable parameters that allows for fair comparison between the models. Furthermore, the selected network architecture is relatively compact, which in return allows the development of methods capable of deblurring an image in milliseconds. In detail, the architecture of the implemented UNet consists of a contracting path and an expansive path. Each step of the contracting path involves application of 2 convolutional layers of 3×3 kernels each followed by a 2×2 max pooling operation for downsampling. The number

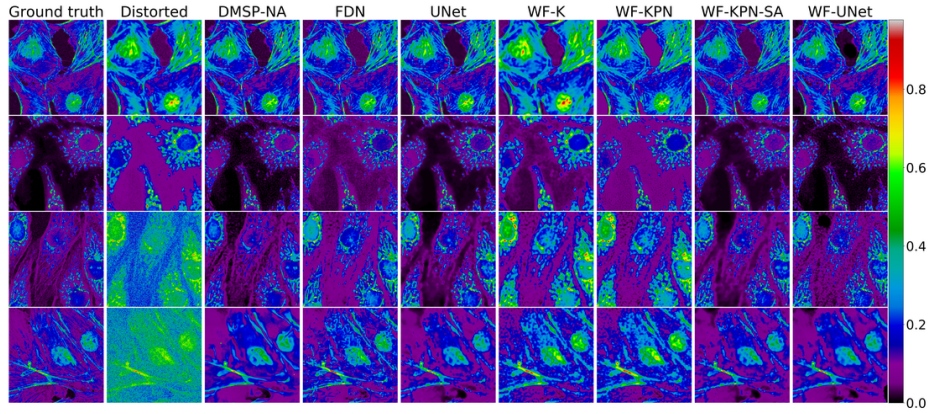


Fig. 2. Restoration of microscopy images degraded by PSF and Gaussian noise with standard deviation equal to 0.005, 0.01, 0.05, 0.1, respectively, from top to bottom. All images are originally grayscale but a different colormap is used to better highlight the differences among the various reconstructions

of feature channels starts from 12 and increases at each downsampling except the last one by a factor of 2 reaching a value of 96. Each step of the expanding path of the network consists of an upsampling, concatenation with the feature map from the contracting path and 2 convolutional layers of 3×3 kernels each. Respectively, the number of channels decreases at each upsampling except the last one by a factor of 2. The rectified linear unit is applied after each convolutional layer except the final convolutional layer that produces the output of the network.

5 Network Training

5.1 Dataset

Since there is a lack of a deconvolution dataset, comprising of microscopy images perturbed by Gaussian noise, we create training pairs of ground-truth and distorted with blur and noise images using the Fluorescence Microscopy Denoising (FMD) dataset [50] and a dataset that is used in cell segmentation of microscopy images [4]. To create a large set of reference images, we take the ground-truth images from both datasets and crop each image into patches of size 256×256 . Consequently, all patches that contain no information about cells are discarded by comparing the mean value of a patch with the mean value of the original image. Note that all ground-truth images as well as resulting distorted images are grayscale.

To produce blurred versions of the ground-truth images we create 35 different 2D PSFs of sizes 7×7 , 9×9 , 11×11 and 13×13 by altering the index of refraction of the media, the numerical aperture, the pixel size and the excitation wavelength using the ImageJ plugins Diffraction PSF 3D [1] and PSF Generator [2] with the

Richards & Wolf optical model. In detail, 25 PSFs are reserved for training, 5 for validation and the rest for testing purposes. Notably, all PSFs \mathbf{k} are normalized such that $\sum_i k_i = 1$, which is a realistic assumption in optics [15].

To assess the performance of our algorithms on a wide range of Gaussian noise levels, all images are scaled in the range of $[0, 1]$, then blurred with a PSF and finally perturbed with i.i.d. Gaussian noise with standard deviation from the set $(0.001, 0.005, 0.01, 0.05, 0.1)$.

The resulting dataset, which contains 1405 pairs of ground-truth and distorted images, is split into 975 training, 200 validation and 230 testing samples. During training, a ground-truth sample is convolved with a randomly chosen blur kernel from the 25 training PSFs, and subsequently Gaussian noise with a randomly selected standard deviation from the set presented above is added to the blurred image.

5.2 Training Details

All proposed methods are trained in an end-to-end fashion to minimize the ℓ_1 loss function between the output and the ground-truth image as well as the gradients of the aforementioned entities

$$\mathcal{L} = \|\hat{\mathbf{x}} - \mathbf{x}\|_1 + \|\nabla \hat{\mathbf{x}} - \nabla \mathbf{x}\|_1. \quad (12)$$

It has been reported that training with the ℓ_1 loss function yields images with sharper edges [51]. Also, by incorporating a gradient based loss alongside the pixel-wise loss, we obtain models that are capable of reconstructing the sharp details of the underlying images. All proposed methods are optimized using Adam [19] with a learning rate 10^{-3} . For all methods, except the WF-KPN-SA, we use batch size equal to 25 and train the models for 300 epochs. For the WF-KPN-SA network, due to memory-related limitations, we set the batch size to be equal to 3.

5.3 Evaluation

To evaluate the performance of all algorithms on each noise level, we use 230 ground-truth images that are convolved with the 5 PSFs reserved for testing purposes. Note that the held-out PSFs were not seen during training in order to explore the generalization ability of the methods to unknown blurring kernels. Finally, the noisy observation of a blurry image is produced by adding Gaussian noise with each standard deviation from the set $(0.001, 0.005, 0.01, 0.05, 0.1)$. In detail, we get 1150 test samples in total, combined into 5 test sets of different noise levels which allows the thorough comparison of the performance of our algorithms on each noise level separately.

We compare our proposed algorithms with three state-of-the-art algorithms for deblurring images distorted by Gaussian noise: IRCNN [48], FDN [23] and DMSP [9]. The latter was considered in two implementations – noise-blind, or noise-adaptive (NA), and noise-aware, or non-blind (NB). We also provide a CNN

Table 1. PSNR and SSIM comparisons on Gaussian image deblurring for five noise levels. In all tables, we highlight the best result in bold and underline the second best result

	STD									
	0.001		0.005		0.01		0.05		0.1	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Input	36.23	.8955	35.37	.8791	33.93	.8339	26.03	.3858	21.14	.1718
IRCNN [48]	33.33	.8604	36.88	.8972	36.80	.9013	32.44	.7932	28.95	.6835
FDN [23]	<u>40.31</u>	.9424	38.61	.9239	37.33	.9086	33.50	.8406	30.49	.7550
DMSP-NA [9]	40.44	.9402	39.16	.9290	<u>37.73</u>	<u>.9123</u>	<u>34.31</u>	<u>.8589</u>	32.19	.8312
DMSP-NB [9]	40.27	<u>.9411</u>	37.23	.9027	36.59	.8930	34.29	.8582	32.20	<u>.8313</u>
UNet	37.73	.9177	37.27	.9136	36.64	.9064	34.04	.8556	<u>32.40</u>	.8187
WF-K	35.66	.8849	35.61	.8834	35.45	.8787	32.74	.7950	29.27	.6835
WF-KPN	38.72	.9253	37.98	.9176	36.80	.9028	32.33	.8022	29.20	.7259
WF-KPN-SA	39.86	.9390	<u>38.76</u>	<u>.9275</u>	37.81	.9157	34.58	.8688	32.60	.8363
WF-UNet	38.08	.9102	37.55	.9053	36.77	.8966	33.89	.8442	32.01	.8096

baseline which is a UNet with the same architecture and number of parameters as the one used for the parametrization of the regularization gradient. This way we get a valid comparison and we are able to investigate if the combination of a standard optimization scheme and a deep learning approach will help to achieve good restoration results and surpass those obtained by employing solely a deep learning based algorithm. For assessing the methods performance we use the standard peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM [43]) metrics. Comparisons of all the methods runtime was conducted on a computer with an Intel Core i7-8750H CPU and a NVIDIA GeForce GTX 1080Ti GPU.

6 Results

We evaluate all methods on the developed test set across all different noise ranges, as described in Section 5.3. The results presented in Table 1 show that our proposed spatially adaptive method WF-KPN-SA surpasses in performance all competing state-of-the-art methods except in the very low noise cases. The performance over the deep learning based competing methods FDN [23], IRCNN [48] and DMSP [9] ranges from nearly 0.1 in the low noise regime to 0.4 dB for high noise. Only in the case of very low noise the previous state-of-the-art methods provide better reconstruction quality than WF-KPN-SA. However, we stress that, as well as DMSP-NA, our methods are noise-blind, while IRCNN, FDN and DMSP-NB are noise-aware, with the oracle noise standard deviation being explicitly provided as an input.

Furthermore, we find that Wiener Filter (WF-K) with the learnable regularization filters yields satisfactory reconstruction quality and perform only slightly worse than several deep learning methods for a fraction of its computational complexity. A detailed benchmark of computational time for all methods presented

Table 2. Runtime of several deconvolution algorithms for processing an image with the spatial dimensions of 256×256 . All times were calculated using the publicly available implementations and the reported benchmarks are the average of 10 runs

	CPU, ms GPU, ms			CPU, ms GPU, ms	
GILAM [12]	4155.9	–	UNet	32.4	2.8
HSPiRAL [26]	5828.8	–	WF-K	5.6	4.6
PURE-LET [27]	211.3	–	WF-KPN	45.4	7.7
FDN [23]	34542.1	1095.0	WF-KPN-SA	968.2	341.4
DMSP [9]	79656.0	19707.7	WF-UNet	354.1	39.8
IRCNN [48]	7890.7	–			

in our work can be found in Table 2. As presented in Fig. 2, WF-K and WF-KPN methods tend to miss sharp details in the restored images, while WF-KPN-SA and WF-UNet allow the restoration of fine image details.

7 Poisson Image Deblurring

As an extension, all developed methods are applied in the case of confocal microscopy, where as it was mentioned in Section 1 the intensity of the light illuminating the sample is very small and thus the noise statistics obey the Poisson distribution. One notable property of the Poisson distribution is that the mean and variance of the random variable are not independent, i.e. $\text{mean}(\mathbf{y}) = \text{var}(\mathbf{y})$. As such, the image formation model is formulated as $\mathbf{y} = \mathcal{P}(\mathbf{K}\mathbf{x})$ where \mathcal{P} denotes the Poisson noise distorting the image. One successful way to perform deconvolution and denoising in the presence of Poisson noise is to first apply a variance stabilizing transformation (VST) [31] which transforms a variable from the Poisson distribution into one from the Gaussian. This allows to borrow the ample apparatus of the well-studied methods derived for Gaussian statistics. This approach has been successfully applied in the literature [31,18,50,28] to solve the Poissonian restoration problem using Gaussian denoising algorithms.

In our work we explore a similar approach and make use of the widely known VST, the Anscombe transform [6], which is applied on the distorted observation

$$\mathbf{y} \rightarrow 2\sqrt{\mathbf{y} + \frac{3}{8}}. \quad (13)$$

The Anscombe transform aims to stabilize the data variance to be approximately unity. After the transformation, the data can be viewed as a signal-independent Gaussian process with unit variance, and therefore the Wiener filter, which is derived from Gaussian statistics, can be directly applied. Once the deconvolved solution, denoted as $\hat{\mathbf{x}}$, is obtained, an inverse Anscombe transform is applied to return the data to its original domain. Applying the simple algebraic inverse usually results in a biased estimate of the output. To mitigate the bias in the case of photon-limited imaging the exact unbiased inverse transformation should

Table 3. PSNR and SSIM comparisons on Poisson image deblurring for six different noise levels. Methods that were not able to produce meaningful results due to numerical instability issues are marked with N/A

	PEAK											
	1		2		5		10		25		50	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Input	10.93	.0981	13.22	.1104	16.67	.1639	19.45	.2425	23.21	.3930	26.00	.5247
GILAM [12]	24.57	.4977	25.66	.6350	26.64	.5451	28.24	.6304	29.92	.6975	30.97	.7463
HSPiRAL [26]	23.18	.4101	26.54	.5809	30.18	.7576	31.94	.8229	33.39	.8578	34.38	.8729
PURE-LET [27]	26.18	.7318	26.44	.7494	26.77	.7612	27.67	.7799	28.77	.8065	28.77	.8114
FDN [23]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	31.16	.8176
DMSP-NA [9]	16.07	0.4243	18.86	.5648	23.80	.7126	27.76	.7880	31.74	.8490	33.65	.8770
DMSP-NB [9]	10.86	.2079	12.97	.2761	20.49	.5691	26.12	.7356	30.25	.8136	33.01	.8633
IRCNN [48]	6.81	.0902	11.67	.1809	17.37	.3695	22.14	.5323	29.46	.8038	33.10	.8660
UNet	28.50	.7916	29.79	.8143	31.47	.8428	32.72	.8633	34.20	.8871	35.17	.9008
WF-K	25.81	.5821	27.85	.6947	29.94	.7888	31.13	.8288	32.06	.8557	32.43	.8654
WF-KPN	27.09	.7503	28.19	.7811	29.89	.8177	31.34	.8460	33.22	.8770	34.50	.8945
WF-KPN-SA	28.80	.7949	30.12	.8185	31.72	.8468	32.92	.8672	34.48	.8909	35.60	.9056
WF-UNet	29.04	.8005	30.27	.8220	31.83	.8492	33.06	.8694	34.59	.8928	35.67	.9069

be used, whose closed-form approximation [30] is

$$\hat{\mathbf{x}} \rightarrow \left(\frac{\hat{\mathbf{x}}}{2}\right)^2 - \frac{1}{8} + \frac{1}{4}\sqrt{\frac{3}{2}}\hat{\mathbf{x}}^{-1} - \frac{11}{8}\hat{\mathbf{x}}^{-2} + \frac{5}{8}\sqrt{\frac{3}{2}}\hat{\mathbf{x}}^{-3}. \quad (14)$$

Data Preparation and Results To assess the performance of our algorithms on the task of Poisson image deblurring, we retrain our models using the VST and the exact unbiased inverse transformation included into the pipeline. The Anscombe transform is applied on the input image \mathbf{y} as described in Eq. (13) before being fed into any of the proposed models of Section 4. Accordingly, the restored signal $\hat{\mathbf{x}}$ from any of the proposed models is transformed back using the exact unbiased version of the inverse transformation, presented in Eq. (14), in order to recover the final solution. For this task we use the same dataset as the one used for the evaluation of the Gaussian deblurring. We create 35 2D PSFs of size 5×5 to simulate confocal microscope pinhole with the ImageJ Diffraction PSF 3D plugin [2]. We again use 25 PSFs for training purposes, 5 for validation and 5 for test. To simulate various SNR values of Poisson noise, we use the prior art approach [12,26] and scale the ground-truth images to have a maximum intensity of (1, 2, 5, 10, 25, 50). Poisson noise is signal dependent with local $\text{SNR} = \sqrt{y_i}$, where y_i denotes the underlying image intensity at position i , therefore by increasing the maximum intensity of an image, the amount of noise decreases and vice versa. Furthermore, it is long known fact that images with large mean value and perturbed with Poisson noise follow approximately a normal, or Gaussian distribution, and therefore methods based on Gaussian statistics might work equally well. Scaling the ground-truth images to have various ranges of maximum intensities aims to cover a wide gamut of noise levels, including the ones that belong to approximately a Gaussian distribution.

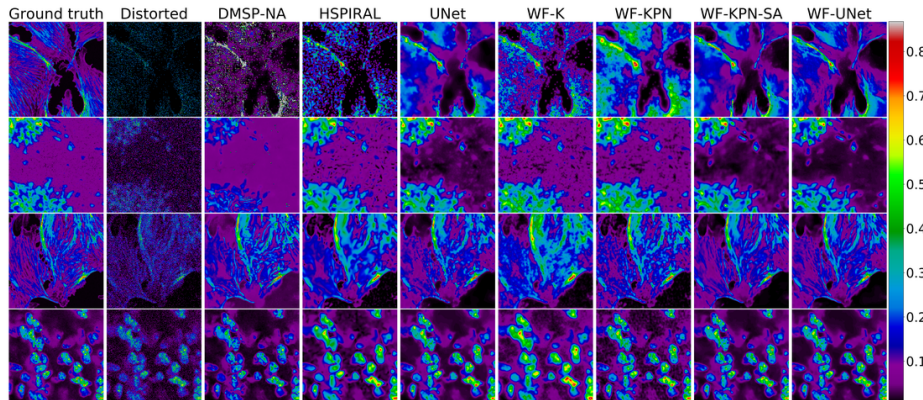


Fig. 3. Restoration of microscopy images scaled to have peak intensities equal to 1, 5, 10, 25, respectively from top to bottom and degraded by PSF and Poisson noise. All images are originally grayscale, but a different colormap is used to better highlight the differences among the various reconstructions

During training, a ground-truth sample is rescaled to a randomly chosen maximum intensity from the range mentioned above. Then the sample is convolved with a randomly chosen blur kernel from the 25 training PSFs, and after that the noisy observation is produced from the blurred image. We follow the same strategy as in Section 5.3 for producing blurred and noisy observations for the test set.

We compare the proposed algorithms with the non-blind deblurring algorithms, developed for Poisson distribution: GILAM [12], HSPiRAL [26] and PURE-LET [27]. We also include the aforementioned state-of-the-art methods for Gaussian image deblurring into the comparison by applying the VST to the distorted data and the exact unbiased transformation to the result of the restorations. Table 3 and Fig. 3 clearly show that WF-UNet outperforms the other approaches at most of the noise levels, which is especially important on the low intensity peak values where the Poisson noise is stronger. WF-KPN-SA shows comparable to WF-UNet performance being only marginally inferior. Overall, WF-UNet and WF-KPN-SA show good quantitative improvement from nearly 2.6dB to 5.8dB over the previous state-of-the-art methods on the low intensity peaks and from nearly 1.2dB to 7dB on the high intensity peaks. Moreover, Fig. 3 demonstrates that, despite the good quantitative improvement, WF-K and WF-KPN struggle to reconstruct the fine details in the images, while the reconstructions generated by WF-KPN-SA and WF-UNet methods are very accurate.

8 Conclusion

In this work, we proposed a series of methods based on the Wiener-Kolmogorov filtering technique for dealing with the problem of Gaussian and Poisson image

deblurring. We introduced three novel ways to parametrize the image priors, including a new approach of regularization with kernel predictions obtained by a neural network. Our extensive experimentation line showcased that our proposed framework based on the prediction of a regularizer achieves superior quality of image reconstruction and surpasses the solutions that rely either on deep learning or on optimization schemes alone. Finally, several of our proposed algorithms demonstrate low computational complexity, without sacrificing the accuracy of image restoration. Being fast and accurate, the proposed framework paves the way towards real-time microscopy image restoration.

References

1. Diffraction psf 3d. <https://www.optinav.info/Diffraction-PSF-3D.htm>, accessed: 2019-05-30
2. Psf generator. <http://bigwww.epfl.ch/algorithms/psfgenerator/#ref>, accessed: 2020-02-25
3. Introduction to variational image-processing models and applications. *Int. J. Comput. Math.* **90**, 1–8 (2013)
4. Al-Kofahi, Y., Zaltsman, A.B., Graves, R., Marshall, W., Rusu, M.: A deep learning-based algorithm for 2-d cell segmentation in microscopy images. In: *BMC Bioinformatics* (2018)
5. AL-Qinani, I.H.: Deblurring image and removing noise from medical images for cancerous diseases using a wiener filter. *IRJET* **8**(4), 2354–2365 (2017)
6. Anscombe, F.J.: The transformation of poisson, binomial and negative-binomial data (1948)
7. van Beek, P., Yang, J., Yamamoto, S., Ueda, Y.: Image deblurring and denoising with non-local regularization constraint. In: *Electronic Imaging* (2010)
8. Bertsekas, D.P.: *Nonlinear programming: Second edition* (1999)
9. Bigdeli, S.A., Zwicker, M., Favaro, P., Jin, M.: Deep mean-shift priors for image restoration. In: *NIPS* (2017)
10. Boyat, A.K., Joshi, B.K.: Image denoising using wavelet transform and wiener filter based on log energy distribution over poisson-gaussian noise model. 2014 IEEE International Conference on Computational Intelligence and Computing Research pp. 1–6 (2014)
11. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) **2**, 60–65 vol. 2 (2005)
12. Chen, D.Q.: Regularized generalized inverse accelerating linearized alternating minimization algorithm for frame-based poissonian image deblurring. *SIAM J. Imaging Sciences* **7**, 716–739 (2014)
13. Chowdhury, A., Dylov, D.V., Li, Q., MacDonald, M., Meyer, D.E., Marino, M., Santamaria-Pang, A.: Blood vessel characterization using virtual 3D models and convolutional neural networks in fluorescence microscopy. In: 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017). pp. 629–632. IEEE (apr 2017). <https://doi.org/10.1109/ISBI.2017.7950599>, <http://ieeexplore.ieee.org/document/7950599/>
14. Conchello, J.A., Lichtman, J.W.: Fluorescence microscopy. *Nature Methods* **2**(12), 910–919 (2005)

15. Dey, N., Blanc-Féraud, L., Zimmer, C., Roux, P., Kam, Z., Olivo-Marin, J.C., Zerubia, J.: Richardson-lucy algorithm with total variation regularization for 3d confocal microscope deconvolution. *Microscopy research and technique* **69** 4, 260–6 (2006)
16. Eigen, D., Krishnan, D., Fergus, R.: Restoring an image taken through a window covered with dirt or rain. 2013 IEEE International Conference on Computer Vision pp. 633–640 (2013)
17. Evangelista, V., Barsanti, L., Passarelli, V., Gualtieri, P.: From cells to proteins: Imaging nature across dimensions (2005)
18. Foi, A., Trimeche, M., Katkovnik, V., Egiazarian, K.O.: Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing* **17**, 1737–1754 (2008)
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *CoRR abs/1412.6980* (2014)
20. Kokkinos, F., Lefkimmiatis, S.: Iterative joint image demosaicking and denoising using a residual denoising network. *IEEE Transactions on Image Processing* **28**(8), 4177–4188 (Aug 2019). <https://doi.org/10.1109/TIP.2019.2905991>
21. Kokkinos, F., Lefkimmiatis, S.: Deep image demosaicking using a cascade of convolutional residual denoising networks. In: ECCV (2018)
22. Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-laplacian priors. In: NIPS (2009)
23. Kruse, J., Rother, C., Schmidt, U.: Learning to push the limits of efficient fft-based image deconvolution. 2017 IEEE International Conference on Computer Vision (ICCV) pp. 4596–4604 (2017)
24. Lefkimmiatis, S.: Non-local color image denoising with convolutional neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
25. Lefkimmiatis, S.: Universal denoising networks: A novel cnn architecture for image denoising. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
26. Lefkimmiatis, S., Unser, M.: Poisson image reconstruction with Hessian Schatten-norm regularization. *IEEE Transactions on Image Processing* **22**, 4314–4327 (2013)
27. Li, J., Luisier, F., Blu, T.: Pure-let image deconvolution. *IEEE Transactions on Image Processing* **27**, 92–105 (2018)
28. Lu, H., Cheng, J.H., Han, G., Li, L., Liang, Z.: 3d distance-weighted wiener filter for poisson noise reduction in sinogram space for spect imaging (2001)
29. Lucy, L.B.: An iterative technique for the rectification of observed distributions (1974)
30. Mäkitalo, M., Foi, A.: A closed-form approximation of the exact unbiased inverse of the anscombe variance-stabilizing transformation. *IEEE Transactions on Image Processing* **20**, 2697–2698 (2011)
31. Mäkitalo, M., Foi, A.: Optimal inversion of the generalized anscombe transformation for poisson-gaussian noise. *IEEE Transactions on Image Processing* **22**, 91–103 (2013)
32. Mildenhall, B., Barron, J.T., Chen, J., Sharlet, D., Ng, R., Carroll, R.: Burst denoising with kernel prediction networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 2502–2510 (2017)
33. de Monvel, J.B., Calvez, S.L., Ulfendahl, M.: Image restoration for confocal microscopy: improving the limits of deconvolution, with application to the visualization of the mammalian hearing organ. *Biophysical journal* **80** 5, 2455–70 (2001)

34. Richardson, W.H.: Bayesian-based iterative method of image restoration* (1972)
35. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena* **60**(1-4), 259–268 (1992)
36. Schmidt, U., Roth, S.: Shrinkage fields for effective image restoration. 2014 IEEE Conference on Computer Vision and Pattern Recognition pp. 2774–2781 (2014)
37. Sheppard, C.J.R., Wilson, T.: Image formation in confocal scanning microscopes (1980)
38. Shewchuk, J.R.: An introduction to the conjugate gradient method without the agonizing pain. Tech. rep., Pittsburgh, PA, USA (1994)
39. Tao, X., Gao, H., Liao, R., Wang, J., Jia, J.: Detail-revealing deep video super-resolution. 2017 IEEE International Conference on Computer Vision (ICCV) pp. 4482–4490 (2017)
40. Tikhonov, A.N.: Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.* **4**, 1035–1038 (1963)
41. Tintner, G., Kailath, T.: Linear least-squares estimation (1980)
42. Ulyanov, D., Vedaldi, A., Lempitsky, V.S.: Instance normalization: The missing ingredient for fast stylization. *ArXiv abs/1607.08022* (2016)
43. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* **13**, 600–612 (2004)
44. Wiener, N.: The extrapolation, interpolation and smoothing of stationary time series, with engineering applications. (1949)
45. Wu, Q., Merchant, F., Castleman, K.: Microscope image processing. Elsevier (2010)
46. Xu, L., Ren, J.S.J., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. In: NIPS (2014)
47. Zhang, J., shan Pan, J., Lai, W.S., Lau, R.W.H., Yang, M.H.: Learning fully convolutional networks for iterative non-blind deconvolution. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 6969–6977 (2016)
48. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep cnn denoiser prior for image restoration. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2808–2817 (2017)
49. Zhang, X.: An effective sure-based wiener filter for image denoising. In: ICC 2016 (2016)
50. Zhang, Y., Zhu, Y., Nichols, E., Wang, Q., Zhang, S., Smith, C., Howard, S.: A poisson-gaussian denoising dataset with real fluorescence microscopy images. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
51. Zhao, H., Gallo, O., Frosio, I., Kautz, J.: Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging* **3**, 47–57 (2017)

Microscopy Image Restoration with Deep Wiener-Kolmogorov Filters – Supplementary Material

1 Derivation of expressions for backward pass for WF-K and WF-KPN

Under the assumption of periodic image boundary conditions, the degradation matrix \mathbf{K} and the convolution matrix \mathbf{G}_d are circulant and, therefore, they can be diagonalized in the Fourier domain. Hence, the solution of the Wiener filter for the proposed models WF-K and WF-KPN can be expressed in a closed form as

$$\hat{\mathbf{x}} = \mathbf{F}^H \left(\frac{\mathbf{D}_{\mathbf{K}}^* \mathbf{F} \mathbf{y}}{|\mathbf{D}_{\mathbf{K}}|^2 + e^\alpha \sum_{d=1}^D |\mathbf{D}_{\mathbf{G}_d}|^2} \right), \quad (1)$$

where the division of the vector in the numerator by the diagonal matrix in the denominator is applied in an element-by-element fashion.

The regularization kernels \mathbf{g}_d and the power of the trade-off α are learned during the training process using back-propagation. While the solution to Eq. 1 comprises complex quantities, the loss function used for the network training is real-valued, allowing the parameter update to be performed using back-propagation. For this reason, we have implemented our customized layers which depend on the analytical derivations of the gradients of the solution $\hat{\mathbf{x}}$ w.r.t. the trainable parameters.

Below, we present the derivation of the expressions of the gradients for the backward pass. Unless explicitly stated otherwise, we assume the use of denominator-layout notation in the calculations.

1.1 Gradient w.r.t. α

We start the derivation of the closed-form expressions for back-propagation with a calculation of the gradient of the Wiener filter solution $\hat{\mathbf{x}}$ w.r.t. the parameter α . Let us denote the solution in Eq. (1) as a function defined by the parameters,

$$f(\alpha, \mathbf{y}, \mathbf{g}_d) = \mathbf{F}^H \left(\frac{\mathbf{D}_{\mathbf{K}}^* \mathbf{F} \mathbf{y}}{|\mathbf{D}_{\mathbf{K}}|^2 + e^\alpha \sum_{d=1}^D |\mathbf{D}_{\mathbf{G}_d}|^2} \right). \quad (2)$$

Note that similarly to Eq. (1), the division operation in the Eq. (2) is meant to describe the element-wise division of the vector in the numerator by the corresponding diagonal element of the diagonal matrix in the denominator.

For simplicity and compactness of the calculations we denote the numerator and the denominator in Eq. (2) as

$$\mathbf{D}_{\mathbf{K}}^* \mathbf{F} \mathbf{y} = \mathbf{z}, \quad (3)$$

$$|\mathbf{D}_{\mathbf{K}}|^2 + e^\alpha \sum_{d=1}^D |\mathbf{D}_{\mathbf{G}_d}|^2 = \Omega. \quad (4)$$

With the notations defined in Eqs. (3) and (4), the function in Eq. (2) yields the form of

$$f(\alpha, \mathbf{y}, \mathbf{g}_d) = \mathbf{F}^H \Omega^{-1} \mathbf{z}. \quad (5)$$

Note that Ω^{-1} is a diagonal matrix. Assuming the numerator-layout notation, the gradient of $f(\alpha, \mathbf{y}, \mathbf{g}_d)$ w.r.t. the parameter α can be calculated as

$$\begin{aligned} \frac{\partial f(\alpha, \mathbf{y}, \mathbf{g}_d)}{\partial \alpha} &= \frac{\partial \mathbf{F}^H \Omega^{-1} \mathbf{z}}{\partial \alpha} = -\mathbf{F}^H \Omega^{-1} \frac{\partial \Omega}{\partial \alpha} \Omega^{-1} \mathbf{z} = \\ &= -\mathbf{F}^H \Omega^{-1} e^\alpha \sum_{d=1}^D |\mathbf{D}_{\mathbf{G}_d}|^2 \Omega^{-1} \mathbf{z} = -e^\alpha \mathbf{F}^H \Omega^{-2} \sum_{d=1}^D |\mathbf{D}_{\mathbf{G}_d}|^2 \mathbf{D}_{\mathbf{K}}^* \mathbf{F} \mathbf{y}. \end{aligned} \quad (6)$$

1.2 Gradient w.r.t. \mathbf{g}_d

Next, we derive the expression for the gradient of $f(\alpha, \mathbf{y}, \mathbf{g}_d)$ w.r.t. the regularization kernel \mathbf{g}_d . To do that, we rewrite Eq. (2) as $f(\alpha, \mathbf{y}, \mathbf{g}_d) = \mathbf{F}^H \mathbf{h}(\mathbf{g})$, where

$$\mathbf{h}(\mathbf{g}) = \frac{\lambda_{\mathbf{K}}^* \odot \mathbf{F} \mathbf{y}}{|\lambda_{\mathbf{K}}|^2 + e^\alpha \sum_{d=1}^D |\mathbf{T}_d \mathbf{g}_d|^2}. \quad (7)$$

This way, we can express the gradient of $f(\alpha, \mathbf{y}, \mathbf{g}_d)$ w.r.t. \mathbf{g}_d as

$$\frac{\partial f(\alpha, \mathbf{y}, \mathbf{g}_d)}{\partial \mathbf{g}_d} = \frac{\partial \mathbf{F}^H \mathbf{h}(\mathbf{g})}{\partial \mathbf{g}_d} = \frac{\partial \mathbf{h}(\mathbf{g})}{\partial \mathbf{g}_d} \mathbf{F}^*. \quad (8)$$

We emphasize that $\mathbf{D}_{\mathbf{K}}$ and $\mathbf{D}_{\mathbf{G}_d}$ are diagonal matrices and we use the following notation:

$$\begin{aligned} \lambda_{\mathbf{K}} &= \text{vec}(\mathbf{D}_{\mathbf{K}}) \\ \lambda_{\mathbf{G}_d} &= \text{vec}(\mathbf{D}_{\mathbf{G}_d}) = \mathbf{T}_d \mathbf{g}_d \\ \mathbf{T}_d &= \mathbf{F} \mathbf{S}_{\mathbf{G}_d} \mathbf{P}_{\mathbf{G}_d} \in \mathbb{C}^{N \times L_d}. \end{aligned} \quad (9)$$

Here $\text{vec}(\mathbf{D})$ is meant to define the vector which lies on the main diagonal of the diagonal matrix \mathbf{D} . Note that the operation \odot corresponds to element-wise multiplication, and the division in Eq. (7) applies element-wisely. Here $\mathbf{S}_{\mathbf{G}_d} \in \mathbb{R}^{N \times N}$ is a circulant shift operator and $\mathbf{P}_{\mathbf{G}_d} \in \mathbb{R}^{N \times L_d}$ is a zero-padding operator.

Taking into consideration Eq. (7) and the notations in Eq. (9), $\mathbf{h}(\mathbf{g})$ can be rewritten as

$$\mathbf{h}(\mathbf{g}) = \begin{bmatrix} \frac{\mathbf{M}_1 \boldsymbol{\lambda}_{\mathbf{K}}^* \odot \mathbf{F}\mathbf{y}}{|\mathbf{M}_1 \boldsymbol{\lambda}_{\mathbf{K}}|^2 + e^\alpha \sum_{d=1}^D |\mathbf{M}_1 \mathbf{T}_d \mathbf{g}_d|^2} \\ \dots \\ \frac{\mathbf{M}_N \boldsymbol{\lambda}_{\mathbf{K}}^* \odot \mathbf{F}\mathbf{y}}{|\mathbf{M}_N \boldsymbol{\lambda}_{\mathbf{K}}|^2 + e^\alpha \sum_{d=1}^D |\mathbf{M}_N \mathbf{T}_d \mathbf{g}_d|^2} \end{bmatrix}, \quad (10)$$

where $\mathbf{M}_i \in \mathbb{R}^{1 \times N}$ is a vector every element of which except i is equal to zero, and the i -th element is equal to 1. Therefore, the i -th element of the vector $\mathbf{h}(\mathbf{g})$ is

$$\mathbf{h}_i(\mathbf{g}) = \frac{\mathbf{M}_i \boldsymbol{\lambda}_{\mathbf{K}}^* \odot \mathbf{F}\mathbf{y}}{|\mathbf{M}_i \boldsymbol{\lambda}_{\mathbf{K}}|^2 + e^\alpha \sum_{d=1}^D |\mathbf{M}_i \mathbf{T}_d \mathbf{g}_d|^2} = \frac{\mathbf{a}_i}{\mathbf{b}_i + e^\alpha \mathbf{u}_i(\mathbf{g})}. \quad (11)$$

Here we denote

$$\begin{aligned} \mathbf{a}_i &= \mathbf{M}_i \boldsymbol{\lambda}_{\mathbf{K}}^* \odot \mathbf{F}\mathbf{y}, \\ \mathbf{b}_i &= |\mathbf{M}_i \boldsymbol{\lambda}_{\mathbf{K}}|^2 \\ \mathbf{u}_i &= \sum_{d=1}^D |\mathbf{M}_i \mathbf{T}_d \mathbf{g}_d|^2 = \sum_{d=1}^D \mathbf{g}_d^\top \mathbf{T}_d^{*\top} \mathbf{M}_i^\top \mathbf{M}_i \mathbf{T}_d \mathbf{g}_d. \end{aligned} \quad (12)$$

To calculate the gradient of $\mathbf{h}_i(\mathbf{g})$ w.r.t. the j -th regularization kernel \mathbf{g}_j one must perform indirect differentiation, starting with acquiring the gradient of $\mathbf{u}_i(\mathbf{g})$ w.r.t. \mathbf{g}_j ,

$$\begin{aligned} \frac{\partial \mathbf{u}_i(\mathbf{g})}{\partial \mathbf{g}_j} &= \sum_{d=1}^D \left(\frac{\partial \mathbf{g}_d}{\partial \mathbf{g}_j} \mathbf{T}_d^{*\top} \mathbf{M}_i^\top \mathbf{M}_i \mathbf{T}_d \mathbf{g}_d + \frac{\partial \mathbf{g}_d}{\partial \mathbf{g}_j} \mathbf{T}_d^\top \mathbf{M}_i^\top \mathbf{M}_i \mathbf{T}_d^* \mathbf{g}_d \right) = \\ &= (\mathbf{T}_j^{*\top} \mathbf{M}_i^\top \mathbf{M}_i \mathbf{T}_j + \mathbf{T}_j^\top \mathbf{M}_i^\top \mathbf{M}_i \mathbf{T}_j^*) \mathbf{g}_j. \end{aligned} \quad (13)$$

Incorporating the result obtained in Eq. (13) into $\frac{\partial \mathbf{h}_i(\mathbf{g})}{\partial \mathbf{g}_j}$, we derive

$$\begin{aligned} \frac{\partial \mathbf{h}_i(\mathbf{g})}{\partial \mathbf{g}_j} &= \mathbf{a}_i \frac{\partial (\mathbf{b}_i + e^\alpha \mathbf{u}_i(\mathbf{g}))^{-1}}{\partial \mathbf{g}_j} = -\mathbf{a}_i \frac{e^\alpha}{(\mathbf{b}_i + e^\alpha \mathbf{u}_i(\mathbf{g}))^2} \frac{\partial \mathbf{u}_i(\mathbf{g})}{\partial \mathbf{g}_j} = \\ &= -e^\alpha \frac{\mathbf{M}_i \boldsymbol{\lambda}_{\mathbf{K}}^* \odot \mathbf{F}\mathbf{y}}{(|\mathbf{M}_i \boldsymbol{\lambda}_{\mathbf{K}}|^2 + e^\alpha \sum_{d=1}^D |\mathbf{M}_i \mathbf{T}_d \mathbf{g}_d|^2)^2} \cdot (\mathbf{R}_{ij} + \overline{\mathbf{R}_{ij}}) \mathbf{g}_j, \end{aligned} \quad (14)$$

where $\mathbf{R}_{ij} = \mathbf{T}_j^H \mathbf{M}_i^\top \mathbf{M}_i \mathbf{T}_j$ and $\overline{\mathbf{R}_{ij}}$ is conjugate to \mathbf{R}_{ij} .

The gradient of $\frac{\partial \mathbf{h}(\mathbf{g})}{\partial \mathbf{g}_j}$ can be written as

$$\frac{\partial \mathbf{h}(\mathbf{g})}{\partial \mathbf{g}_j} = \left[\frac{\partial \mathbf{h}_1(\mathbf{g})}{\partial \mathbf{g}_j} \quad \frac{\partial \mathbf{h}_2(\mathbf{g})}{\partial \mathbf{g}_j} \quad \dots \quad \frac{\partial \mathbf{h}_N(\mathbf{g})}{\partial \mathbf{g}_j} \right] \in \mathbb{R}^{Ld \times N} \quad (15)$$

Taking into consideration Eq. (15) and applying the chain rule, we derive the expression for $\frac{\partial \mathbf{h}(\mathbf{g})}{\partial \mathbf{g}_j} \mathbf{F}^* \mathbf{q}$ from Eq. (8), where $\mathbf{q} \in \mathbb{R}^{N \times 1}$ is a real-valued vector, backpropagated from further layers. This way we can write

$$\frac{\partial \mathbf{h}(\mathbf{g})}{\partial \mathbf{g}_j} \mathbf{F}^* \mathbf{q} = \sum_{i=1}^D \frac{\partial \mathbf{h}_i(\mathbf{g})}{\partial \mathbf{g}_j} \mathbf{Q}_i, \quad \mathbf{Q}_i = \mathbf{M}_i \mathbf{F}^* \mathbf{q}. \quad (16)$$

Let us derive the expression under the summation in Eq. (16) as

$$\begin{aligned} \frac{\partial \mathbf{h}_i(\mathbf{g})}{\partial \mathbf{g}_j} \mathbf{Q}_i &= -e^\alpha \frac{\mathbf{M}_i \boldsymbol{\lambda}_{\mathbf{K}}^* \odot \mathbf{F} \mathbf{y}}{(|\mathbf{M}_i \boldsymbol{\lambda}_{\mathbf{K}}|^2 + e^\alpha \sum_{d=1}^D |\mathbf{M}_i \mathbf{T}_d \mathbf{g}_d|^2)^2} \cdot \\ &\cdot (\mathbf{R}_{ij} + \overline{\mathbf{R}_{ij}}) \mathbf{g}_j \mathbf{M}_i \mathbf{F}^* \mathbf{q} = -e^\alpha (\mathbf{R}_{ij} + \overline{\mathbf{R}_{ij}}) \mathbf{g}_j \mathbf{M}_i \mathbf{z}. \end{aligned} \quad (17)$$

Here we use the notation $\mathbf{D}_{\mathbf{K}} = \text{diag}(\boldsymbol{\lambda}_{\mathbf{K}})$, $\mathbf{D}_{\mathbf{G}_d} = \text{diag}(\boldsymbol{\lambda}_{\mathbf{G}_d})$, and therefore $\mathbf{z} = \frac{\mathbf{D}_{\mathbf{K}}^* \mathbf{F} \mathbf{y}}{(|\mathbf{D}_{\mathbf{K}}|^2 + e^\alpha \sum_{d=1}^D |\mathbf{D}_{\mathbf{G}_d}|^2)^2} \odot \mathbf{F}^* \mathbf{q}$.

Here, the division operation corresponds to the element-wise division of the vector in the numerator by the corresponding diagonal elements of the diagonal matrix in the denominator. Here $\text{diag}(\boldsymbol{\lambda})$ denotes a square matrix with vector $\boldsymbol{\lambda}$ on the main diagonal and zeros otherwise.

Incorporating Eq. (17) into Eq. (16) and considering $\mathbf{Q} = \mathbf{F}^* \mathbf{q}$ we can write

$$\begin{aligned} \frac{\partial \mathbf{h}(\mathbf{g})}{\partial \mathbf{g}_j} \mathbf{Q} &= -e^\alpha \sum_{i=1}^D (\mathbf{R}_{ij} + \overline{\mathbf{R}_{ij}}) \mathbf{g}_j \mathbf{M}_i \mathbf{z} = -e^\alpha \sum_{i=1}^D (\mathbf{T}_j^H \mathbf{M}_i^\top \mathbf{M}_i \mathbf{T}_j + \\ &+ \overline{\mathbf{T}_j^H \mathbf{M}_i^\top \mathbf{M}_i \mathbf{T}_j}) \mathbf{g}_j \mathbf{z}_i = -e^\alpha \mathbf{P}_{\mathbf{G}_j}^\top \mathbf{S}_{\mathbf{G}_j}^\top \left(\mathbf{F}^H \sum_{i=1}^D \mathbf{M}_i^\top \mathbf{M}_i \boldsymbol{\lambda}_{\mathbf{G}_j} \mathbf{z}_i + \right. \\ &+ \mathbf{F}^\top \sum_{i=1}^D \mathbf{M}_i^\top \mathbf{M}_i \boldsymbol{\lambda}_{\mathbf{G}_j}^* \mathbf{z}_i \left. \right) = -e^\alpha \mathbf{P}_{\mathbf{G}_j}^\top \mathbf{S}_{\mathbf{G}_j}^\top (\mathbf{F}^H (\boldsymbol{\lambda}_{\mathbf{G}_j} \odot \mathbf{z}) + \\ &+ \mathbf{F}^\top (\boldsymbol{\lambda}_{\mathbf{G}_j}^* \odot \mathbf{z})) = -e^\alpha \mathbf{P}_{\mathbf{G}_j}^\top \mathbf{S}_{\mathbf{G}_j}^\top (\mathbf{F}^H (\boldsymbol{\lambda}_{\mathbf{G}_j} \odot \mathbf{z}) + \overline{\mathbf{F}^H (\boldsymbol{\lambda}_{\mathbf{G}_j} \odot \mathbf{z}^*)}). \end{aligned} \quad (18)$$

We note that $\mathbf{F}^H (\boldsymbol{\lambda}_{\mathbf{G}_j} \odot \mathbf{z}^*)$ is real, therefore $\overline{\mathbf{F}^H (\boldsymbol{\lambda}_{\mathbf{G}_j} \odot \mathbf{z}^*)} = \mathbf{F}^H (\boldsymbol{\lambda}_{\mathbf{G}_j} \odot \mathbf{z}^*)$. This way Eq. (18) can be rewritten as

$$\begin{aligned} \frac{\partial \mathbf{h}(\mathbf{g})}{\partial \mathbf{g}_j} \mathbf{Q} &= -e^\alpha \mathbf{P}_{\mathbf{G}_j}^\top \mathbf{S}_{\mathbf{G}_j}^\top (\mathbf{F}^H (\boldsymbol{\lambda}_{\mathbf{G}_j} \odot \mathbf{z}) + \mathbf{F}^H (\boldsymbol{\lambda}_{\mathbf{G}_j} \odot \mathbf{z}^*)) = \\ &= -e^\alpha \mathbf{P}_{\mathbf{G}_j}^\top \mathbf{S}_{\mathbf{G}_j}^\top \mathbf{F}^H (\boldsymbol{\lambda}_{\mathbf{G}_j} \odot (\mathbf{z} + \mathbf{z}^*)) = \\ &= -2e^\alpha \mathbf{P}_{\mathbf{G}_j}^\top \mathbf{S}_{\mathbf{G}_j}^\top \mathbf{F}^H (\boldsymbol{\lambda}_{\mathbf{G}_j} \odot \text{Re}(\mathbf{z})). \end{aligned} \quad (19)$$

This way the expression for the gradient of $f(\alpha, \mathbf{y}, \mathbf{g}_d)$ with respect to the regularization kernel \mathbf{g}_d is

$$\begin{aligned} \frac{\partial f(\alpha, \mathbf{y}, \mathbf{g}_d)}{\partial \mathbf{g}_d} \mathbf{q} &= -2e^\alpha \mathbf{P}_{\mathbf{G}_d}^\top \mathbf{S}_{\mathbf{G}_d}^\top \mathbf{F}^H \cdot \\ &\cdot \left[\boldsymbol{\lambda}_{\mathbf{G}_d} \odot \text{Re} \left(\frac{\mathbf{D}_{\mathbf{K}}^* \mathbf{F} \mathbf{y}}{(|\mathbf{D}_{\mathbf{K}}|^2 + e^\alpha \sum_{d=1}^D |\mathbf{D}_{\mathbf{G}_d}|^2)^2} \odot \mathbf{F}^* \mathbf{q} \right) \right]. \end{aligned} \quad (20)$$

The division operation in Eq. (20) defines the element-wise division of the vector in the numerator by the corresponding diagonal elements of the diagonal matrix in the denominator.

1.3 Gradient w.r.t. \mathbf{y}

Finally, we derive the expression for the gradient of $f(\alpha, \mathbf{y}, \mathbf{g}_d)$ w.r.t. the input \mathbf{y} . Although this formula is not used in the training routine presented herein, we include the derivation to cover a general case of a larger pipeline which would comprise our network, e.g., when several Wiener filters are stacked into a sequence. For this derivation, we use the original non FFT-based formulation of the Wiener filter:

$$f(\alpha, \mathbf{y}, \mathbf{g}_d) = (\mathbf{K}^\top \mathbf{K} + e^\alpha \sum_{d=1}^D \mathbf{G}_d^\top \mathbf{G}_d)^{-1} \mathbf{K}^\top \mathbf{y}. \quad (21)$$

Here we denote $\mathbf{K}^\top \mathbf{K} + e^\alpha \sum_{d=1}^D \mathbf{G}_d^\top \mathbf{G}_d = \mathbf{B}$. This way, Eq. (21) can be rewritten as $(\mathbf{K}^\top \mathbf{K} + e^\alpha \sum_{d=1}^D \mathbf{G}_d^\top \mathbf{G}_d)^{-1} \mathbf{K}^\top \mathbf{y} = \mathbf{B}^{-1} \mathbf{K}^\top \mathbf{y}$, and the expression for the gradient of $f(\alpha, \mathbf{y}, \mathbf{g}_d)$ w.r.t. the input \mathbf{y} can be simplified as

$$\frac{\partial f(\alpha, \mathbf{y}, \mathbf{g}_d)}{\partial \mathbf{y}} = \frac{\partial \mathbf{B}^{-1} \mathbf{K}^\top \mathbf{y}}{\partial \mathbf{y}} = \mathbf{K} \mathbf{B}^{-\top} = \mathbf{K} \mathbf{B}^{-1}. \quad (22)$$

As it was stated above, the matrices \mathbf{K} and \mathbf{G}_d are decomposed in the Fourier domain as

$$\mathbf{K} = \mathbf{F}^H \mathbf{D}_\mathbf{K} \mathbf{F}, \quad \mathbf{G}_d = \mathbf{F}^H \mathbf{D}_{\mathbf{G}_d} \mathbf{F}, \quad (23)$$

where $\mathbf{F} \in \mathbb{C}^{N \times N}$ is the Fourier (DFT) matrix, $\mathbf{F}^H \in \mathbb{C}^{N \times N}$ is its inverse, $\mathbf{D}_\mathbf{K}, \mathbf{D}_{\mathbf{G}_d} \in \mathbb{C}^{N \times N}$ are diagonal matrices. Using the FFT-based inference of the Wiener filter and the notation defined in Eq. (4), the gradient of $f(\alpha, \mathbf{y}, \mathbf{g}_d)$ w.r.t. the input \mathbf{y} in Eq. (22) can be rewritten as

$$\frac{\partial f(\alpha, \mathbf{y}, \mathbf{g}_d)}{\partial \mathbf{y}} = \mathbf{F}^H \mathbf{D}_\mathbf{K} (|\mathbf{D}_\mathbf{K}|^2 + e^\alpha \sum_{d=1}^D |\mathbf{D}_{\mathbf{G}_d}|^2)^{-1} \mathbf{F} = \mathbf{F}^H \boldsymbol{\Omega}^{-1} \mathbf{D}_\mathbf{K} \mathbf{F}. \quad (24)$$

2 Variation of kernels number and size for WF-K and WF-KPN

The proposed models WF-K and WF-KPN employ a group of D regularization kernels of size $K \times K$ to obtain the solution in the form of

$$\hat{\mathbf{x}} = \mathbf{F}^H \left(\frac{\mathbf{D}_\mathbf{K}^* \mathbf{F} \mathbf{y}}{|\mathbf{D}_\mathbf{K}|^2 + e^\alpha \sum_{d=1}^D |\mathbf{D}_{\mathbf{G}_d}|^2} \right). \quad (25)$$

We performed an ablation study to understand the influence of the number and the size of the regularization kernels on the image restoration quality. Specifically, we trained both models, WF-K and WF-KPN, implementing $D = 8$, $K = 3$ and $D = 24$, $K = 5$ regularization kernels. For the ablation study, we used the same dataset and the training pipeline as in the main case. Namely, we implemented a Gaussian deblurring dataset, that was created by taking the

Table 1. PSNR and SSIM comparisons on Gaussian image deblurring for five different noise levels for different number and size of the regularization kernels

		STD									
		0.001		0.005		0.01		0.05		0.1	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
	Input	36.23	.8955	35.37	.8791	33.93	.8339	26.03	.3858	21.14	.1718
$D = 8, K = 3$	WF-K	35.66	.8849	35.61	.8834	35.45	.8787	32.74	.7950	29.27	.6835
	WF-KPN	38.72	.9253	37.98	.9176	36.80	.9028	32.33	.8022	29.20	.7259
$D = 24, K = 5$	WF-K	35.81	.8861	35.75	.8846	35.58	.8798	32.75	.7945	29.21	.6807
	WF-KPN	39.95	.9368	38.41	.9218	37.20	.9057	32.69	.8095	29.20	.7386

ground-truth samples from the FMD [50] and the dataset described in [4] and by cropping them into the tiles of size 256×256 . All images were rescaled to the range $[0,1]$. During the training process, a ground-truth sample from 975 training samples is convolved with a randomly chosen blur kernel from a set of 25 training PSFs, followed by a perturbation with i.i.d. Gaussian noise with standard deviation from the set (0.001, 0.005, 0.01, 0.05, 0.1). To evaluate the performance of the algorithms, we used 5 test sets of images distorted with the Gaussian noise of different levels. In particular, we used 230 ground-truth images and convolved them with a fixed blur kernel from a sub-set of 5 PSFs reserved for testing. Finally, the noisy observation of a blurry image is produced by adding the Gaussian noise, with standard deviations being taken from the set (0.001, 0.005, 0.01, 0.05, 0.1). Note that all ground-truth and the resulting distorted images are grayscale in all experiments (the false color Figures in the main paper were produced to emphasize fine details in the background).

Results in the Table 1 show that increasing the number and the size of kernels in WF-K method leads to improvement in PSNR of nearly 0.15dB (in the low noise regime). Yet, at high noise levels, an increase of the size and the number of regularization kernels results in a marginal PSNR drop of 0.06dB. For WF-KPN, increasing the number and the size of kernels improves the results for all noise levels, from nearly 0.4dB in the high noise regime to nearly 1.25dB at the low noise levels. Table 1 shows that the increase of the parameters of the network expectedly leads to improvement of the results. In particular, increasing the number and the size of the kernels yields better performance at the low noise levels.

Regularization kernels, predicted with the models WF-K and WF-KPN with $D = 8, K = 3$, are shown in Figure 1 along with the results of the restoration with both models. The images show that WF-K provides a group of learnable kernels which are identical for all images, whereas WF-KPN predicts a group of regularization kernels per image. Results of the restoration with both models, presented in the Figure 1, demonstrate that WF-KPN tends to restore images better than WF-K.

3 Additional Results

Additional results for deblurring of images corrupted with the Gaussian noise of different levels are presented in Fig. 2. We observe that our methods are marginally inferior to the DMSP-NA and FDN on the lowest noise levels, however, our method WF-KPN-SA outperforms the other methods in the higher noise regimes, allowing the restoration of the finest image details.

Additional study of the Poisson image deblurring, presented in Fig. 3, further proves that our methods WF-KPN-SA and WF-UNet allow to reconstruct the smallest image details with excellent values of the metrics.

Fig. 1. Predicted kernels in WF-K and WF-KPN methods for different images and the corresponding restorations.

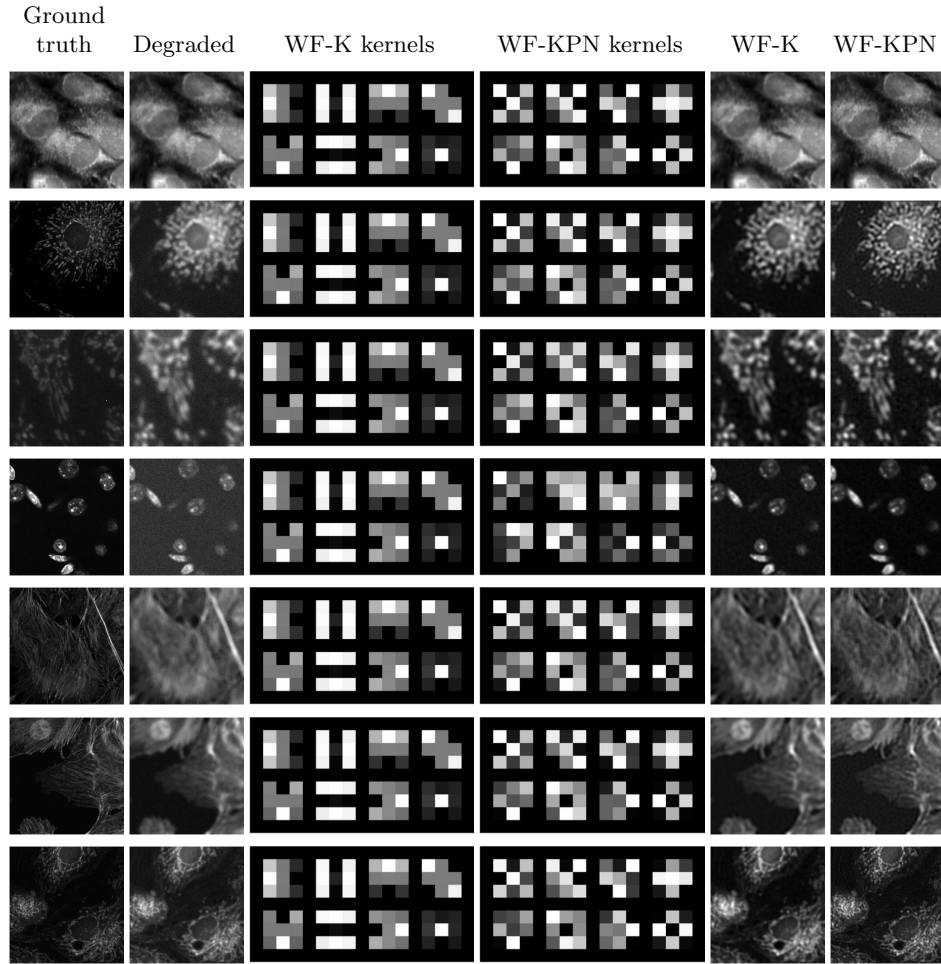


Fig. 2. Restoration of microscopy images degraded by PSF and Gaussian noise with the standard deviations (σ) from the set (0.001, 0.005, 0.01, 0.05, 0.1). The metrics shown beneath each image are PSNR/SSIM. All images are originally grayscale, but are shown in pseudo-color to stress the details in different structures, textures, and in the background.

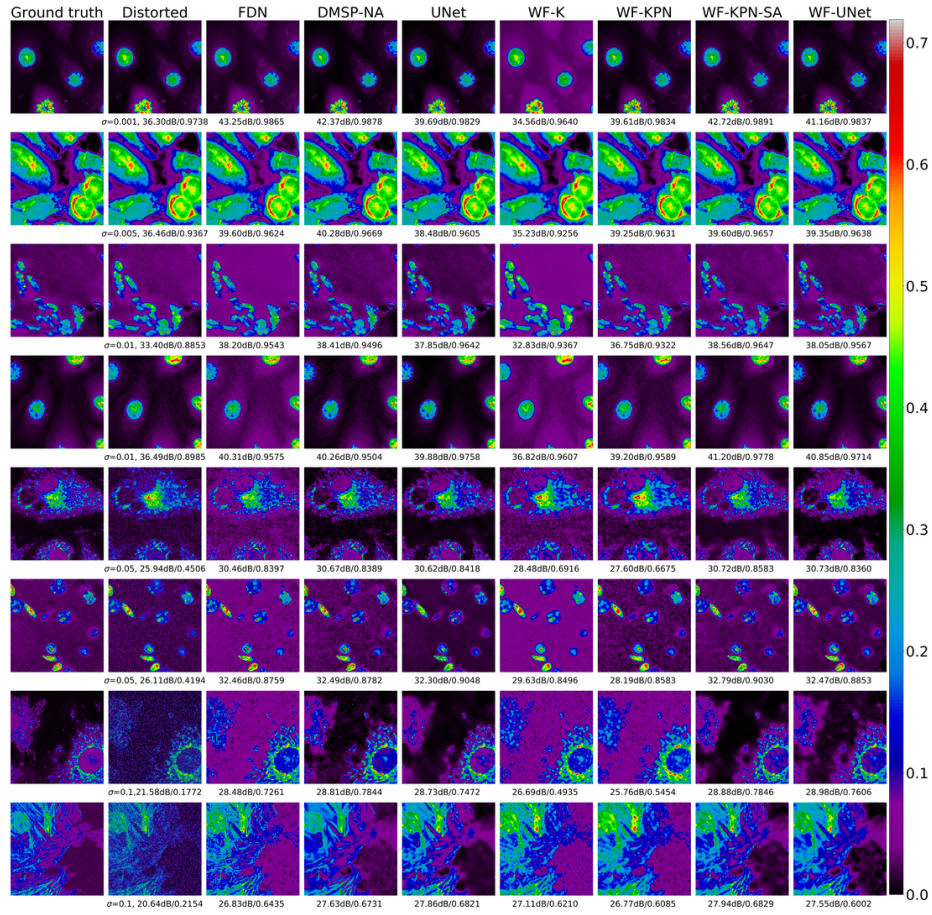


Fig. 3. Restoration of microscopy images scaled to the maximum intensity peaks (p) from the range (1, 2, 5, 10, 25, 50) and degraded by PSF and the Poisson noise. The metrics shown beneath each image are PSNR/SSIM. The images are originally grayscale, but are shown in a pseudo-color to stress the details in different structures, textures, and in the background.

