

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

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

¹ Skolkovo Institute of Science and Technology, Moscow, Russia

Valeriya.Pronina@skoltech.ru

² University College London

³ Q.bio Inc.

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

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

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

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

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

Note that $\boldsymbol{\Omega}^{-1}$ 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

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

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

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

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:

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

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 $\mathbf{P}_{\mathbf{G}_d} \in \mathbb{R}^{N \times Ld}$ 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

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

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

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

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 UNet architecture

UNet implemented in the proposed methods WF-KPN, WF-KPN-SA and WF-UNet, has almost identical architecture in all cases with some modifications for each proposed algorithm. Overall, the architecture of the 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 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.

3 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 ground-truth samples from the FMD [2] and the dataset described in [1] 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.

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
$D = 8, K = 3$	Input	36.23	.8955	35.37	.8791	33.93	.8339	26.03	.3858	21.14	.1718
	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

4 Visualization of PSFs

In microscopy, blurring kernels usually do not exhibit great deviations. Nevertheless, to ensure fair comparison we varied our kernels both in terms of generation of hyper-parameters for kernels implemented both in Poisson and Gaussian noise cases and the support sizes for the kernels implemented only in Gaussian noise case. PSFs used for image blurring during inference in cases of Poisson and Gaussian noise are shown in Fig. 4 along with the examples of the corresponding ground truth and distorted images. Fig. 5 and Fig. 6 show PSFs used for blurring during training and inference respectively in case of Gaussian noise; Fig. 8 and Fig. 7 show PSFs used for blurring during training and inference respectively in case of Poisson noise.

5 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.

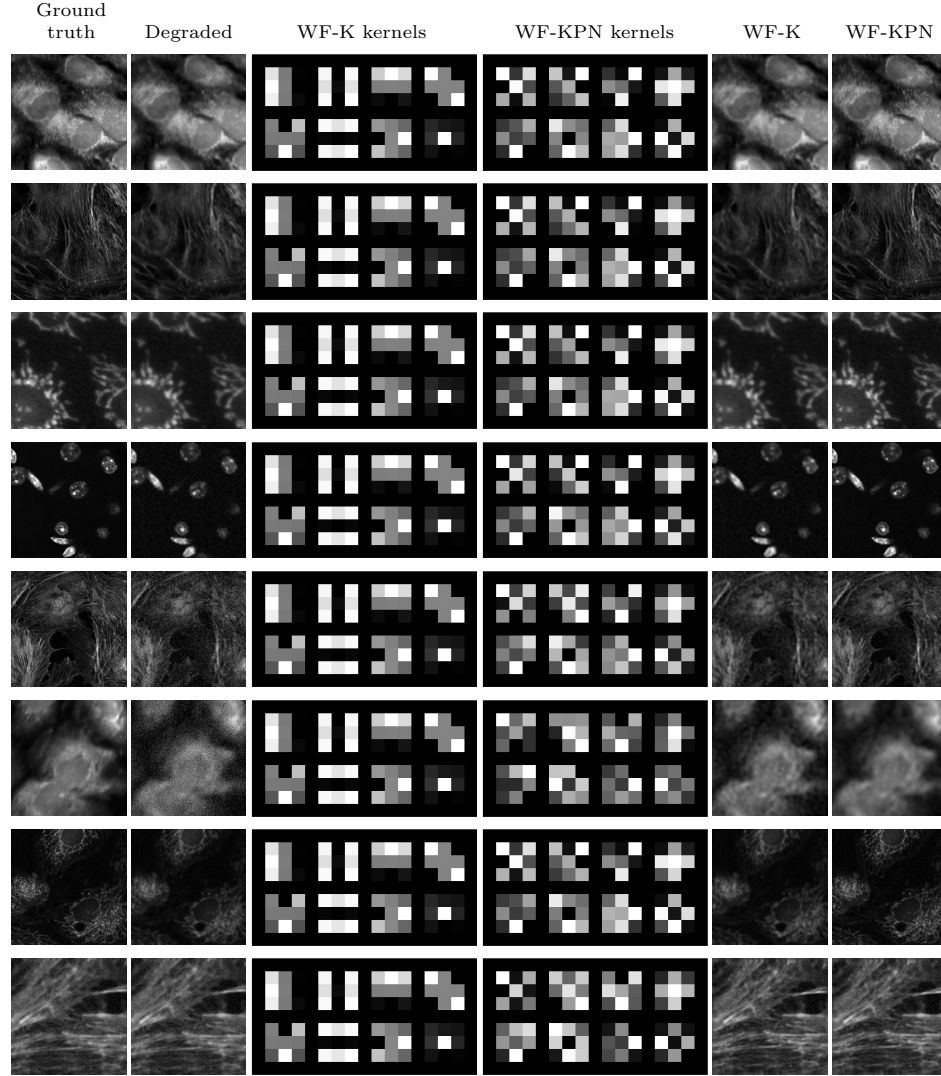


Fig. 1. Predicted kernels in WF-K and WF-KPN methods for different images and the corresponding restorations.

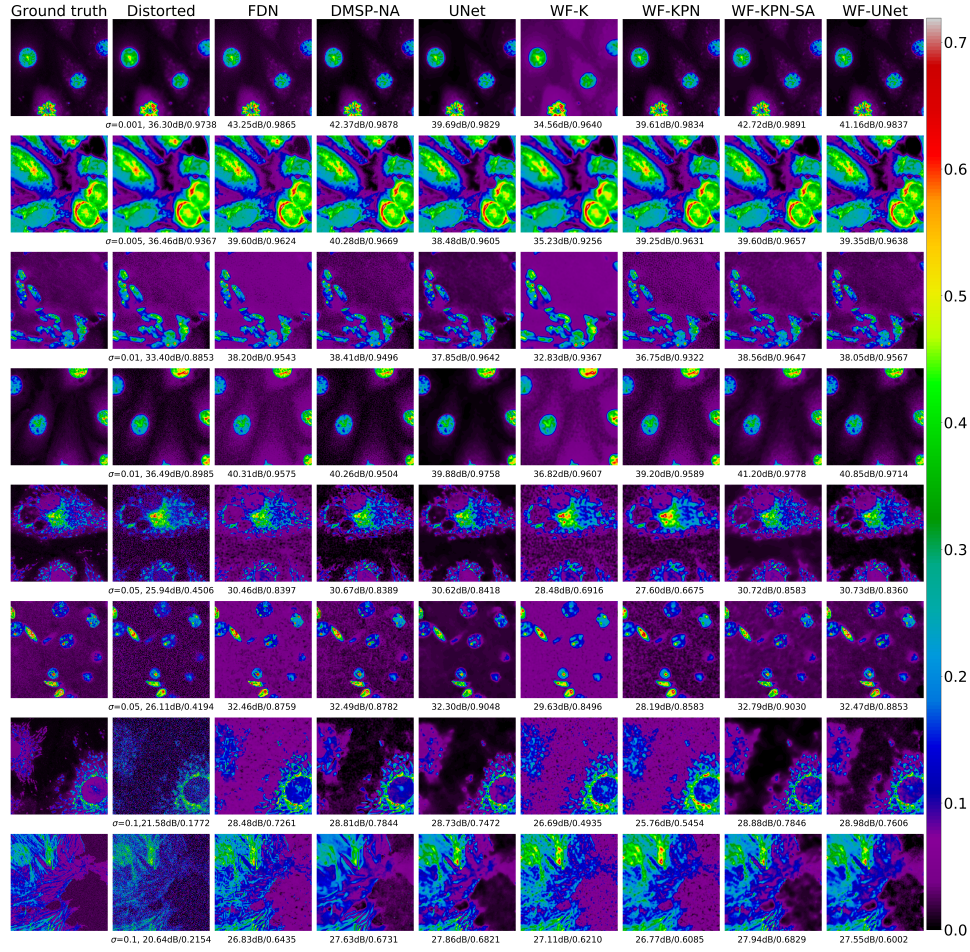


Fig. 2. Restoration of microscopy images degraded by PSF and Gaussian noise with the standard deviations (σ) from the set (0.001, 0.005, 0.01, 0.05, 0.1). The metrics shown beneath each image are PSNR/SSIM. All images are originally grayscale, but are shown in pseudo-color to stress the details in different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

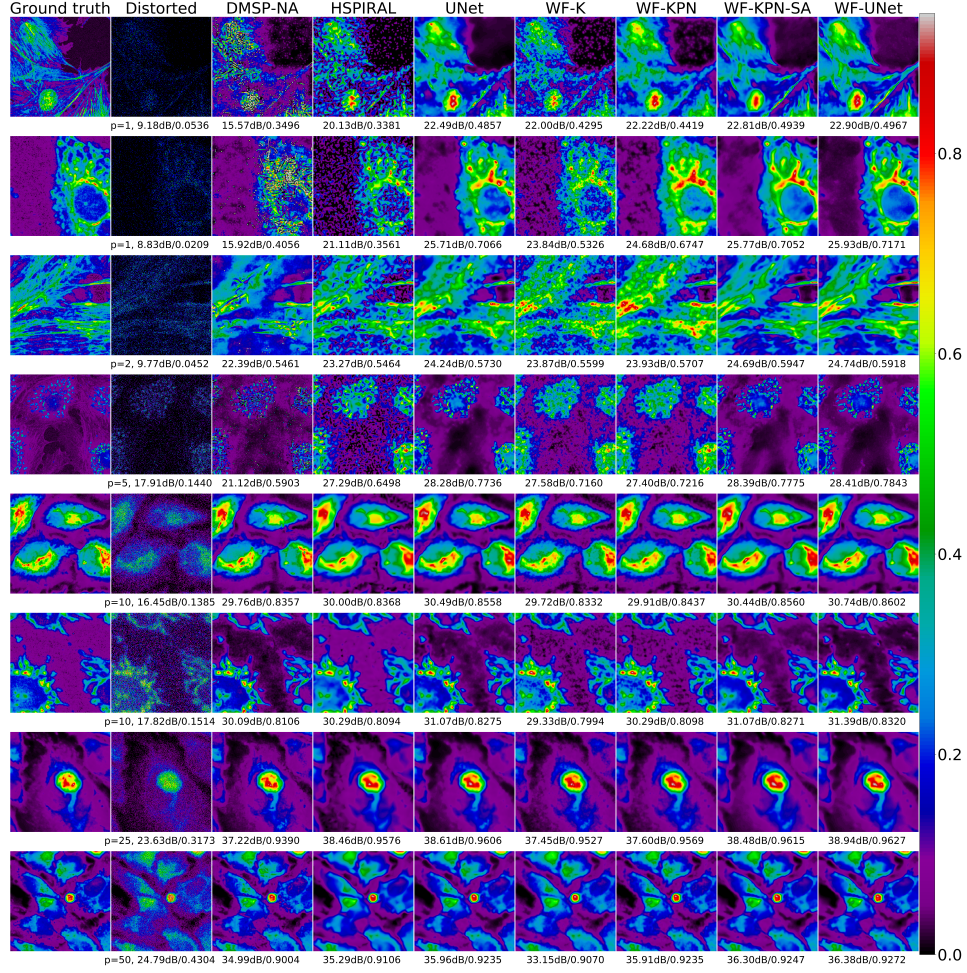


Fig. 3. Restoration of microscopy images scaled to the maximum intensity peaks (p) from the range (1, 2, 5, 10, 25, 50) and degraded by PSF and the Poisson noise. The metrics shown beneath each image are PSNR/SSIM. The images are originally grayscale, but are shown in a pseudo-color to stress the details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

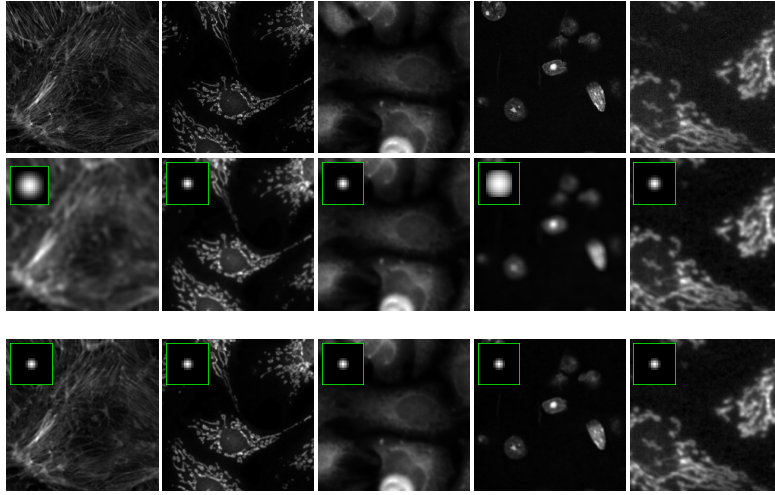


Fig. 4. **Top row.** Examples of the five ground truth images. **Middle row.** Five PSFs used for blurring in the Gaussian noise case during inference with the corresponding blurred images. **Bottom row.** Five PSFs used for blurring in the Poisson noise case during inference with the corresponding blurred images.

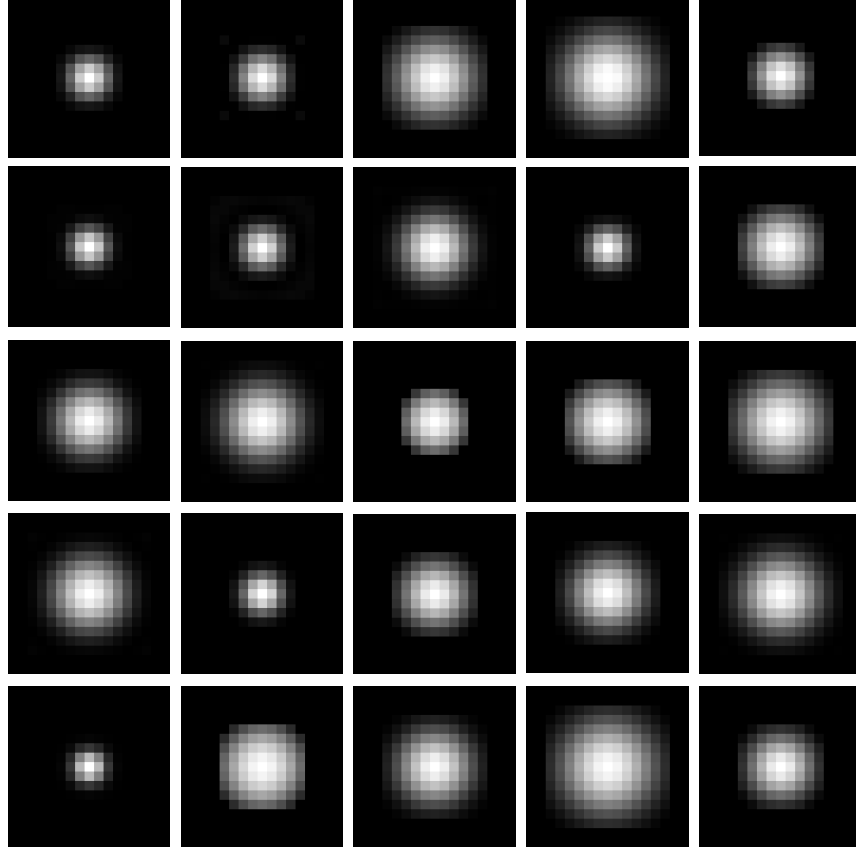


Fig. 5. Twenty five PSFs used for blurring in the Gaussian noise during training.

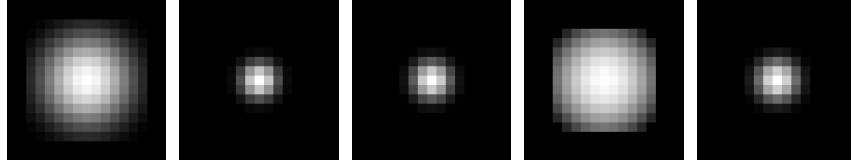


Fig. 6. Five PSFs used for blurring in the Gaussian noise case during inference.

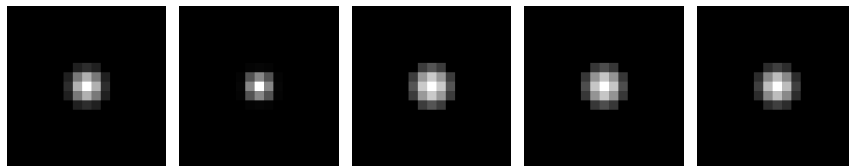


Fig. 7. Five PSF used for blurring in the Poisson noise case during inference.

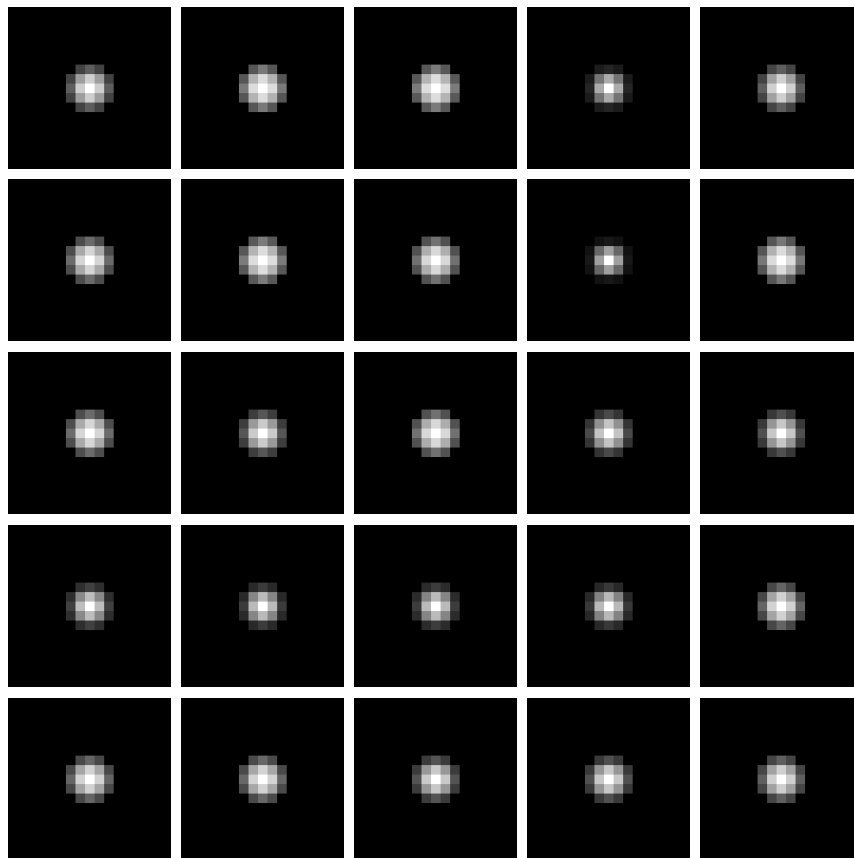


Fig. 8. Twenty five PSFs used for blurring in the Poisson noise during training.

References

1. Al-Kofahi, Y., Zaltsman, A.B., Graves, R.M., Marshall, W., Rusu, M.: A deep learning-based algorithm for 2-d cell segmentation in microscopy images. *BMC Bioinformatics* **19** (2018) [6](#)
2. Zhang, Y., Zhu, Y., Nichols, E., Wang, Q., Zhang, S., Smith, C., Howard, S.: A poisson-gaussian denoising dataset with real fluorescence microscopy images. 2019 IEEE Conference on Computer Vision and Pattern Recognition (2019) [6](#)