

Self-Supervised Single-Image Deconvolution with Siamese Neural Networks

Mikhail Papkov¹, Kaupo Palo², and Leopold Parts^{1,3}

¹ Institute of Computer Science, University of Tartu, Estonia

² Revvity Inc., Tallinn, Estonia

³ Wellcome Sanger Institute, Hinxton, United Kingdom

mikhail.papkov@ut.ee

Abstract. Inverse problems in image reconstruction are fundamentally complicated by unknown noise properties. Classical iterative deconvolution approaches amplify noise and require careful parameter selection for an optimal trade-off between sharpness and grain. Deep learning methods allow for flexible parametrization of the noise and learning its properties directly from the data. Recently, self-supervised blind-spot neural networks were successfully adopted for image deconvolution by including a known point-spread function in the end-to-end training. However, their practical application has been limited to 2D images in the biomedical domain because it implies large kernels that are poorly optimized. We tackle this problem with Fast Fourier Transform convolutions that provide training speed-up in 3D microscopy deconvolution tasks. Further, we propose to adopt a Siamese invariance loss for deconvolution and empirically identify its optimal position in the neural network between blind-spot and full image branches. The experimental results show that our improved framework outperforms the previous state-of-the-art deconvolution methods with a known point spread function.

Keywords: Deconvolution · Microscopy · Deep learning.

1 Introduction and related work

Quality enhancement is central for microscopy imaging [26,29]. Its two most important steps are noise removal and blur reduction that improve performance in downstream tasks for both humans and algorithms.

Denoising, approaches to which were developed decades ago [5], was given a boost by recent advances in deep learning methods [1,3,10,11,14,26,15,19,18], which substantially outperform classical algorithms [5,2]. A direct approach to train a denoising neural network is to provide it with pairs of low and high-quality images [26], but such data are rarely available in practice, which raises the need for alternative techniques. Lehtinen *et al.* [14] proposed to supervise the training with an independently acquired noisy copy of the input, while Batson and Royer [1] concurrently with Krull *et al.* [10] developed a theory of a J -invariant blind-spot denoising network that operates on a single image in a self-supervised manner.

The latter theory is based on the idea that a network cannot learn the noise that is conditionally pixel-wise independent given the signal, so learning to predict the values of masked pixels can improve the performance in restoration. Xie *et al.* [27] further questioned the necessity of blind spots for efficient denoising. They proposed Noise2Same with Siamese invariance loss between outputs of masked and unmasked inputs to prevent the network from learning an identity function. This advance pushed the performance of self-supervised denoising at a cost of doubled training time.

In addition to noise, equipment imperfection imposes blur on the images via a point spread function (PSF) [24]. The blur can be efficiently removed with classical iterative approaches such as Lucy-Richardson [21], but these algorithms tend to amplify the noise with each iteration and require careful regularization and stopping criteria [12]. As supervised methods are not applicable to deconvolution problem due to the ground truth inaccessibility, Lim *et al.* [16] proposed CycleGAN [30] with linear blur kernel, which performed well in both simulation and real microscopy deconvolution. However, the models of this class are notoriously hard to train [17] and tends to converge to a perceptually conceiving solution that does not necessarily reflect the true underlying structure [13]. Besides, it requires clean data examples, albeit unpaired. Deep self-supervised denoising systems were also adopted for deconvolution purposes. Kobayashi *et al.* [9] assumed an intermediate output of the network to be a deconvolved representation and proposed to train Noise2Self [1] as a pseudo-inverse model with a known PSF kernel. This method was only applied to 2D data.

Our contribution is three-fold. Firstly, we adopt the Siamese invariance loss [27] to the deconvolution task and identify the optimal neural network's outputs to apply it. Secondly, we train the first, to our knowledge, 3D self-supervised deconvolutional network without the adversarial component. Lastly, we propose to alleviate the computational costs of training a Siamese network by using the Fast Fourier Transform (FFT) for convolution with the PSF kernel.

2 Methods

We propose a generalized self-supervised deconvolution framework. We follow Kobayashi *et al.* [9] and define the framework as a composition of trainable model $f(\cdot)$ followed by the fixed PSF convolution $g(\cdot)$. The framework (Figure 1) allows for both one-pass blind-spot training[10,1,9] and two-pass training [27].

For the training, we use two inputs, unmasked \mathbf{x} and masked \mathbf{x}_{J^c} where the pixels at locations J are replaced with Gaussian noise. We optimize the trainable model using the composition loss $\mathcal{L}(f)$ of six terms in Eq. (1).

Blind-spot loss. \mathcal{L}_{bsp} is a mean-squared error (MSE) between the network's unmasked input \mathbf{x} and output from the masked forward pass $g(f(\mathbf{x}_{J^c}))_J$ measured at the locations of the masked pixels J [10,1].

Reconstruction loss. \mathcal{L}_{rec} is an MSE between the network's unmasked input \mathbf{x} and output from the unmasked forward pass $g(f(\mathbf{x}))$ [27].

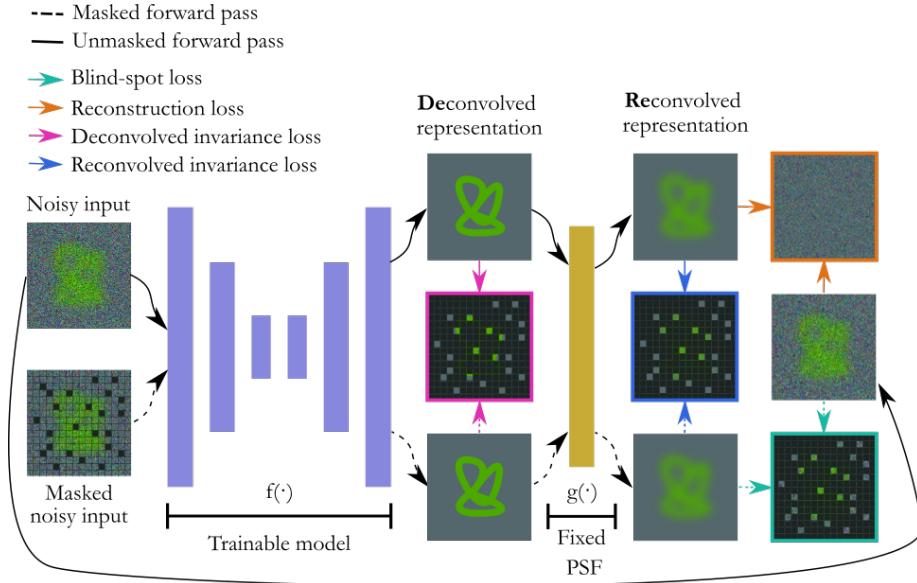


Fig. 1. The self-supervised deconvolution framework. The trainable model $f(\cdot)$ takes noisy input \mathbf{x} in unmasked forward pass (solid) and masked noisy input \mathbf{x}_{J^c} in masked forward pass (dashed). The model produces two intermediate outputs (deconvolved representations). These outputs are convolved with a fixed PSF convolution $g(\cdot)$ to obtain final outputs (reconvolved representations). The deconvolved and reconvolved invariance losses are computed between the respective outputs before and after the PSF. The reconstruction loss is computed between the noisy input and the unmasked forward pass output. The blind-spot loss is computed between the noisy input and masked forward pass output, it does not require an unmasked forward pass.

Reconvolved invariance loss. \mathcal{L}_{inv} is an MSE between the network’s output from the masked forward pass $g(f(\mathbf{x}_{J^c}))_J$ and output from the unmasked forward pass $g(f(\mathbf{x}))_J$.

Deconvolved invariance loss. $\mathcal{L}_{inv(d)}$ is an MSE between the network’s output from the masked forward pass $f(\mathbf{x}_{J^c})_J$ and output from the unmasked forward pass $f(\mathbf{x})_J$ before PSF convolution $g(\cdot)$. Both Siamese invariance losses are computed between the network’s outputs from altered (masked and unmasked) inputs at the location of the masked pixels J . They prevent the network from learning an identity function from \mathcal{L}_{rec} minimization, which is especially important in the single-image deconvolution task.

Boundary losses. $\mathcal{L}_{bound(d)}$ and \mathcal{L}_{bound} regularize the outputs to be within $[min, max]$ boundaries before or after the PSF convolution respectively [9]. These losses are measured after destandardization. We use $min = 0$ and $max = 1$.

In this work, we compare three practical cases. The first one, Eq. (2), considers only the blind-spot loss \mathcal{L}_{bsp} and establishes a self-supervised J -invariant baseline.

The second and the third ones are not J -invariant, both of them include reconstruction loss and invariance loss calculated before (Eq. (4)) or after (Eq. (3)) the PSF convolution. By default, we calculate boundary regularization loss and invariance loss from the same output. We set λ_{inv} and $\lambda_{inv(d)}$ to 2 [27], λ_{bound} and $\lambda_{bound(d)}$ to 0.1 [9] unless otherwise specified.

$$\begin{aligned} \mathcal{L}(f) = & \lambda_{bsp} \mathbb{E}_J \mathbb{E}_x \|g(f(\mathbf{x}_{J^c}))_J - \mathbf{x}_J\|^2 + \\ & \lambda_{rec} \mathbb{E}_x \|g(f(\mathbf{x})) - \mathbf{x}\|^2 / m + \\ & \lambda_{inv} \mathbb{E}_J [\mathbb{E}_x \|g(f(\mathbf{x}))_J - g(f(\mathbf{x}_{J^c}))_J\|^2 / |J|]^{1/2} + \\ & \lambda_{inv(d)} \mathbb{E}_J [\mathbb{E}_x \|f(\mathbf{x})_J - f(\mathbf{x}_{J^c})_J\|^2 / |J|]^{1/2} + \\ & \lambda_{bound} \mathbb{E}_x (|min - g(f(\mathbf{x}))| + |g(f(\mathbf{x})) - max|) + \\ & \lambda_{bound(d)} \mathbb{E}_x (|min - f(\mathbf{x})| + |f(\mathbf{x}) - max|) \end{aligned} \quad (1)$$

$$\mathcal{L}(f)_{\text{Noise2Self}} = \mathcal{L}_{bsp} \quad (2)$$

$$\mathcal{L}(f)_{\text{Noise2Same}} = \mathcal{L}_{rec} + \lambda_{inv} \mathcal{L}_{inv} + \lambda_{bound} \mathcal{L}_{bound} \quad (3)$$

$$\mathcal{L}(f)_{\text{Noise2Same(d)}} = \mathcal{L}_{rec} + \lambda_{inv(d)} \mathcal{L}_{inv(d)} + \lambda_{bound(d)} \mathcal{L}_{bound(d)} \quad (4)$$

Neural network architecture For 2D data, we use the previously proposed [10,27] variation of a U-Net [22] architecture without modifications. The fully-convolutional network of depth 3 consists of an encoder and a decoder with concatenating skip connections at corresponding levels. The first convolutional layer outputs 96 features, and this number doubles with every downsampling step. For 3D data, we modify the network to have 48 output features from the first convolution. We also replace the concatenation operation with an addition in skip connections. Such modification allows for a reduction in training time for 3D convolutional networks without substantial performance sacrifices. We implement the network using PyTorch [20].

Training During training, we sample a patch (128×128 pixels for 2D images, $64 \times 64 \times 64$ for 3D), randomly rotate and flip it. We use batch sizes of 16 for 2D images and 4 for 3D images. Each image is standardized individually to zero mean and unit variance. For the masked forward pass, we randomly replace 0.5% of pixels from each training image with Gaussian noise ($\sigma = 0.2$) [27]. We use Adam [8] optimizer and multiply its learning rate from $4 \cdot 10^{-4}$ by 0.5 every 500 steps out of 3k total for 2D data [9] and every 2k steps out of 15k total for 3D data. We train and evaluate all models on a single NVIDIA V100 16GB GPU.

Inference For inference, we do only unmasked forward pass and discard the PSF convolution during inference [9]. We use the last model checkpoint since we do not possess any reliable validation metric. For large images that do not fit in

GPU memory, we predict by overlapping patches and stitch the output together using pyramid weights [7] which allows for avoiding edge artifacts. In 3D data, we predict patches of $128 \times 128 \times 128$ with overlaps of 32 pixels.

Fast Fourier Transform Convolution We use FFT for the convolution with a fixed PSF during training. Convolution operation $x * k$ is equivalent to element-wise matrix multiplication in Fourier space $F^{-1}(F(x) \odot F(k))$. If image x is $M \times M$ pixels and kernel k is $N \times N$ and both of them have dimensionality of d , the complexity of an ordinary convolution is $O(M^d N^d)$, while the complexity of a Fourier convolution is $O(M^d d \log M)$ and does not depend on N . With large enough N , convolution in Fourier space becomes computationally cheaper, and large kernels (even of the size of the image [24]) are common for PSF. It is recommended to use FFT with 2D kernels for $N > 25$ and with 3D kernels for $N > 9$ [6].

3 Experiments

We validate our approach on 2D and 3D data in a single image deconvolution task. In all the experiments we train one model per image to obtain a lower bound for the extreme self-supervised case. Every image is normalized within $[0, 1]$. We generate blurry images by convolving them with a realistic Richards & Wolf PSF of a 0.8NA 16x microscope objective with 17×17 or $17 \times 17 \times 17$ 0.406-micron pixels. Then we add a mixture of Poisson ($\alpha = 0.001$), Gaussian ($\sigma = 0.1$), and salt-and-pepper (only for 2D images, $p = 0.01$) noise to these images and quantize them with 10 bit precision [9].

We compare the results against classical Lucy-Richardson (LR) deconvolution algorithm [21] and deep learning Self-Supervised Inversion (SSI) [9] algorithm. For LR evaluate the results after 2, 5, 10, and 20 iterations to observe the trade-off between deconvolution quality and noise amplification and report the best ones. We do not include in comparison other classical methods such as Conjugate Gradient optimization [4] and Chambolle-Pock primal-dual inversion [4], because they were shown inferior to both LR and SSI [9]. We rerun the baseline experiments ourselves wherever possible.

We evaluate the model performance against clean images using root mean squared error (RMSE), peak signal-to-noise ratio (PSNR) [28], and structural similarity index(SSIM) [25]. For PSNR and SSIM, we explicitly set the data range to the true values $[0, 1]$. Since it was not done in prior work [9], some values in Table 2 are missing to avoid confusion. Additionally, for 2D images, we report mutual information (MI) [23] and spectral mutual information (SMI) [9]. For all metrics except RMSE, higher is better.

3.1 2D dataset

First, we tested the deconvolution framework performance for the 2D case on a benchmark dataset of 22 single-channeled images [9] with size varying from

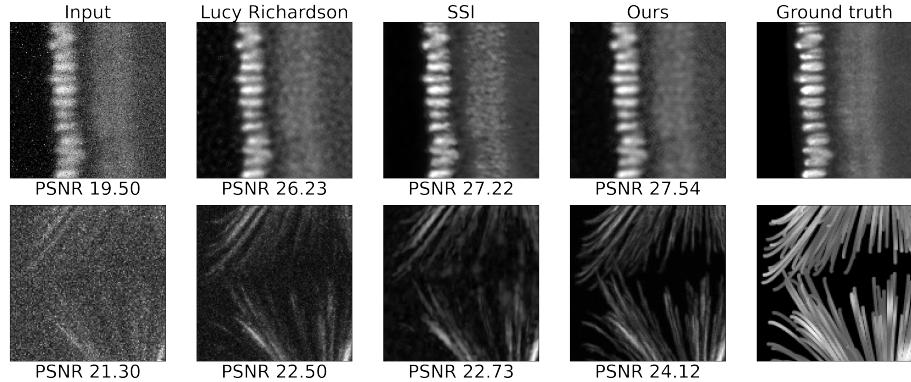


Fig. 2. Example deconvolution results on *Drosophila* 2D image (top), microtubules 3D image, frontal max projection (bottom), magnified. For Lucy Richardson, the result is shown after the best iteration. Our algorithm uses loss from Eq. (4) with $\lambda_{inv(d)} = 2$, $\lambda_{bound(d)} = 0.1$. PSNR metric is reported for the presented images.

512 × 512 to 2592 × 1728 pixels. We compared three different variants of loss function, Eqs. (2)–(4) (Table 1). Loss (4) performed best with $\lambda_{inv(d)} = 2$ and $\lambda_{bound(d)} = 0.1$, achieving PSNR = 22.79, SSIM = 0.46, and RMSE = 0.078.

We compared our model against the baseline LR and SSI methods (Table 2, Figure 2). It performed best by PSNR, SMI, and RMSE, and showed similar results by SSIM (0.01 less than LR) and MI (0.01 less than SSI).

3.2 3D dataset

We evaluated the framework performance for the 3D case on a single synthetic image of microtubules [24] of size 128 × 256 × 512. From the three variants of loss function, Eqs. (2)–(4), loss (3) performed best with $\lambda_{inv} = 2$ and $\lambda_{bound} = 0$ showing the results PSNR = 24.08, SSIM = 0.39, and RMSE = 0.063 (Table 3).

We also compared the best-performing model against the LR and SSI methods (Table 4, Figure 2). Our model surpassed baselines by a substantial margin (+1.5 PSNR against SSI and +1.7 against LR).

4 Discussion

Our Siamese neural network was superior in single-image deconvolution against both classical and deep learning baselines. The architecture is simple and did not require additional tricks such as masking schedule or adding noise to gradients [9] for training. Training converged to similar performance for all of the several tested random weight initializations.

Loss function performance was inconsistent between the 2D and 3D cases. Invariance loss applied to deconvolved representation proved best for 2D data but

Table 1. Denoising results for the 2D dataset with different coefficients for loss (1) components: λ_{bsp} — coefficient for the blind-spot loss, λ_{rec} — coefficient for the reconstruction loss, $\lambda_{inv(d)}$ — coefficient for deconvolved invariance loss, λ_{inv} — coefficient for reconvolved invariance loss, λ_{bound} and $\lambda_{bound(d)}$ — coefficients for boundary regularization loss for reconvolved and deconvolved images. For all metrics except RMSE, higher is better. The best values are highlighted in bold.

	λ_{bsp}	λ_{rec}	$\lambda_{inv(d)}$	λ_{inv}	$\lambda_{bound(d)}$	λ_{bound}	PSNR \uparrow	SSIM \uparrow	RMSE \downarrow
input							17.77	0.18	0.130
$\mathcal{L}(f)_{\text{Noise2Self}}$	1	0	0	0	0	0	22.49	0.27	0.079
	1	0	0	0	0	0.1	22.51	0.43	0.079
$\mathcal{L}(f)_{\text{Noise2Same}}$	0	1	0	2	0	0	21.49	0.23	0.089
	0	1	0	2	0	0.1	21.81	0.41	0.085
$\mathcal{L}(f)_{\text{Noise2Same (d)}}$	0	1	2	0	0	0	22.79	0.46	0.078
	0	1	2	0	0.1	0	22.67	0.46	0.078

Table 2. Our best-performing method for 2D data compared to Lucy Richardson (LR) [21] and Self-Supervised Inversion (SSI) [9] baselines by denoising metrics (see Section 3 for details), training and inference time (measured for *Drosophila* image 1352×532 pixels). For all metrics except RMSE, higher is better. The best values are highlighted in bold. Our algorithm uses loss (4) with $\lambda_{inv(d)} = 2$, $\lambda_{bound(d)} = 0.1$

	PSNR \uparrow	SSIM \uparrow	MI \uparrow	SMI \uparrow	RMSE \downarrow	Train t (s)	Inference t (ms)
input	17.8	0.18	0.07	0.18	0.131	-	-
LR $n = 2$	22.0	0.47	0.13	0.25	0.086	-	46
LR $n = 5$	22.2	0.44	0.12	0.25	0.082	-	48
SSI (reported [9])	22.5	-	0.14	0.27	-	238	26
SSI (reproduced)	22.0	0.46	0.14	0.26	-	442	26
ours	22.8	0.46	0.13	0.30	0.078	856	53

failed to give an advantage in 3D: despite the high SSIM, images appear noisy. We hypothesize that the optimal solution depends on the data; *e.g.* in microtubules, the 3D dataset signal is very sparse. Boundary loss did not drastically affect the training of Siamese networks.

Despite performance superiority, Siamese networks are twice more expensive in computation because they require two forward passes through the neural network. Additionally, Noise2Same U-Net architecture has 10x more trainable parameters than SSI (5.75M against 0.55M in 2D). This problem is exacerbated by the necessity of convolutions with large PSF kernels which are not optimized in modern GPUs. We propose to alleviate this problem by using FFT for convolution with PSF. It leads to 2.3x speed improvement for 3D deconvolution with PSF of size $17 \times 17 \times 17$, and this advantage will grow for larger kernels. For example, while training with a PSF of $31 \times 31 \times 31$ we observed a 14x speedup.

Table 3. Denoising results for the 3D dataset with different coefficients for loss (1) components: λ_{bsp} — coefficient for the blind-spot loss, λ_{rec} — coefficient for the reconstruction loss, $\lambda_{inv(d)}$ — coefficient for deconvolved invariance loss, λ_{inv} — coefficient for reconvolved invariance loss, λ_{bound} and $\lambda_{bound(d)}$ — coefficients for boundary regularization loss for reconvolved and deconvolved images. For all metrics except RMSE, higher is better. The best values are highlighted in bold.

	λ_{bsp}	λ_{rec}	$\lambda_{inv(d)}$	λ_{inv}	$\lambda_{bound(d)}$	λ_{bound}	PSNR \uparrow	SSIM \uparrow	RMSE \downarrow
input							21.21	0.64	0.087
$\mathcal{L}(f)_{\text{Noise2Self}}$	1	0	0	0	0	0	23.71	0.39	0.065
	1	0	0	0	0	0.1	23.71	0.39	0.065
$\mathcal{L}(f)_{\text{Noise2Same}}$	0	1	0	2	0	0	24.08	0.39	0.063
	0	1	0	2	0	0.1	24.02	0.39	0.063
$\mathcal{L}(f)_{\text{Noise2Same (d)}}$	0	1	2	0	0	0	21.05	0.82	0.089
	0	1	2	0	0.1	0	21.05	0.82	0.089

Table 4. Our best performing method for 3D data compared to Lucy Richardson (LR) [21] and Self-Supervised Inversion (SSI) [9] baselines by denoising metrics (see Section 3 for details), training and inference time. For all metrics except RMSE, higher is better. The best values are highlighted in bold. Our algorithm uses loss (3) with $\lambda_{inv} = 2$, $\lambda_{bound} = 0$. Training time is reported with and without FFT (in brackets)

	PSNR \uparrow	SSIM \uparrow	RMSE \downarrow	Train t (s)	Inference t (ms)
input	21.2	0.64	0.087	-	-
LR $n = 2$	22.4	0.32	0.076	-	69
SSI	22.6	0.39	0.074	2814 (6580)	578
ours	24.1	0.39	0.062	8160 (18900)	734

5 Conclusion

Blind-spot networks [10,1] were seminal in self-supervised denoising and performed comparably to supervised methods [14,26]. Their success was later translated to image deconvolution [9], a more complicated inverse problem. However, it was shown that J -invariance leads to suboptimal performance in denoising because masked pixels contain useful bits of information [27]. In this work, we presented a novel unified image deconvolution framework, which generalized the accumulated prior advances, and set a new standard for image quality enhancement performance. We investigated the contributions of various self-supervised loss components and empirically identified the optimal usage scenarios for 2D and 3D data. We also proposed using Fast Fourier Transform for the training of deconvolution neural networks, which drastically speeds up the computation, especially for large kernels. This advantage allows one to use our method for non-confocal microscopy with extremely big point spread functions.

Acknowledgements

This work was funded by Revvity Inc. (VLTAT19682) and Wellcome Trust (206194). We thank High Performance Computing Center of the Institute of Computer Science at the University of Tartu for the provided computing power.

References

1. Batson, J., Royer, L.: Noise2self: Blind denoising by self-supervision. In: International Conference on Machine Learning. pp. 524–533. PMLR (2019)
2. Buades, A., Coll, B., Morel, J.M.: Non-local means denoising. *Image Processing On Line* **1**, 208–212 (2011)
3. Buchholz, T.O., Jordan, M., Pigino, G., Jug, F.: Cryo-care: content-aware image restoration for cryo-transmission electron microscopy data. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). pp. 502–506. IEEE (2019)
4. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision* **40**(1), 120–145 (2011)
5. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing* **16**(8), 2080–2095 (2007)
6. Frank Odom, III, F.O.: fkodom/fft-conv-pytorch. <https://github.com/fkodom/fft-conv-pytorch>
7. Khvedchenya, E.: Pytorch toolbelt. <https://github.com/BloodAxe/pytorch-toolbelt> (2019)
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2017)
9. Kobayashi, H., Solak, A.C., Batson, J., Royer, L.A.: Image deconvolution via noise-tolerant self-supervised inversion. arXiv preprint arXiv:2006.06156 (2020)
10. Krull, A., Buchholz, T.O., Jug, F.: Noise2void-learning denoising from single noisy images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2129–2137 (2019)
11. Krull, A., Vičar, T., Prakash, M., Lalit, M., Jug, F.: Probabilistic noise2void: Unsupervised content-aware denoising. *Frontiers in Computer Science* **2**, 5 (2020)
12. Laasmaa, M., Vendelin, M., Peterson, P.: 3d confocal microscope image enhancement by richardson-lucy deconvolution algorithm with total variation regularization: parameters estimation. *Biophysical Journal* **98**(3), 178a (2010)
13. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4681–4690 (2017)
14. Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., Aila, T.: Noise2noise: Learning image restoration without clean data. arXiv preprint arXiv:1803.04189 (2018)
15. Lemarchand, F., Montesuma, E.F., Pelcat, M., Nogues, E.: Opendedenoising: an extensible benchmark for building comparative studies of image denoisers. arXiv preprint arXiv:1910.08328 (2019)
16. Lim, S., Park, H., Lee, S.E., Chang, S., Sim, B., Ye, J.C.: Cyclegan with a blur kernel for deconvolution microscopy: Optimal transport geometry. *IEEE Transactions on Computational Imaging* **6**, 1127–1138 (2020)

17. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? In: International conference on machine learning. pp. 3481–3490. PMLR (2018)
18. Moran, N., Schmidt, D., Zhong, Y., Coady, P.: Noisier2noise: Learning to denoise from unpaired noisy data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12064–12072 (2020)
19. Papkov, M., Roberts, K., Madissoon, L.A., Shilts, J., Bayraktar, O., Fishman, D., Palo, K., Parts, L.: Noise2stack: Improving image restoration by learning from volumetric data. In: International Workshop on Machine Learning for Medical Image Reconstruction. pp. 99–108. Springer (2021)
20. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in neural information processing systems. pp. 8026–8037 (2019)
21. Richardson, W.H.: Bayesian-based iterative method of image restoration. *JoSA* **62**(1), 55–59 (1972)
22. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
23. Russakoff, D.B., Tomasi, C., Rohlfing, T., Maurer, C.R.: Image similarity using mutual information of regions. In: European Conference on Computer Vision. pp. 596–607. Springer (2004)
24. Sage, D., Donati, L., Soulez, F., Fortun, D., Schmit, G., Seitz, A., Guiet, R., Vonesch, C., Unser, M.: Deconvolutionlab2: An open-source software for deconvolution microscopy. *Methods* **115**, 28–41 (2017)
25. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The Thirly-Seventh Asilomar Conference on Signals, Systems & Computers, 2003. vol. 2, pp. 1398–1402. Ieee (2003)
26. Weigert, M., Schmidt, U., Boothe, T., Müller, A., Dibrov, A., Jain, A., Wilhelm, B., Schmidt, D., Broaddus, C., Culley, S., et al.: Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nature methods* **15**(12), 1090–1097 (2018)
27. Xie, Y., Wang, Z., Ji, S.: Noise2same: Optimizing a self-supervised bound for image denoising. *Advances in Neural Information Processing Systems* **33**, 20320–20330 (2020)
28. Yuanji, W., Jianhua, L., Yi, L., Yao, F., Qinzhong, J.: Image quality evaluation based on image weighted separating block peak signal to noise ratio. In: International Conference on Neural Networks and Signal Processing, 2003. Proceedings of the 2003. vol. 2, pp. 994–997. IEEE (2003)
29. 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: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11710–11718 (2019)
30. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)

Supplementary material

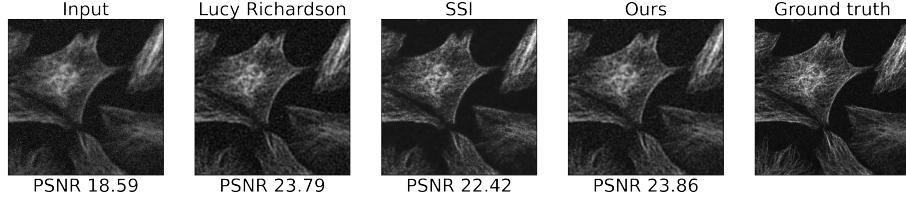


Fig. S1. Example deconvolution results on *Tubules* 2D image, magnified. For Lucy Richardson the result is shown after 5 iterations. Our algorithm uses loss from Eq. (4) with $\lambda_{inv(d)} = 2$, $\lambda_{bound(d)} = 0.1$. PSNR metric is reported for the presented image.

Table S1. Our best-performing method for 2D data compared to Lucy Richardson (LR) baseline by denoising metrics (see Section 3 for details), training and inference time (measured for *Drosophila* image 1352×532 pixels). For all metrics except RMSE, higher is better. The best values are highlighted in bold. Our algorithm uses loss (4) with $\lambda_{inv(d)} = 2$, $\lambda_{bound(d)} = 0.1$

	PSNR ↑	SSIM ↑	MI ↑	SMI ↑	RMSE ↓	Train t (s)	Inference t (ms)
input	17.8	0.18	0.07	0.18	0.131	-	-
LR $n = 2$	22.0	0.47	0.13	0.25	0.086	-	46
LR $n = 5$	22.2	0.44	0.12	0.25	0.082	-	48
LR $n = 10$	21.1	0.36	0.10	0.25	0.091	-	62
LR $n = 20$	18.5	0.26	0.08	0.19	0.119	-	97
ours	22.8	0.46	0.13	0.30	0.078	856	53

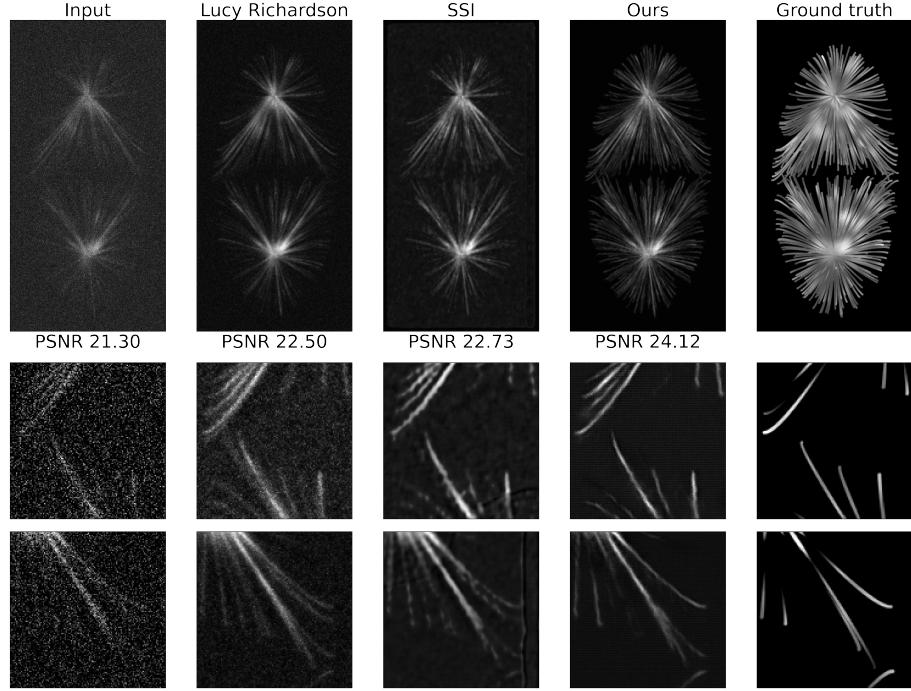


Fig. S2. Example deconvolution results on microtubules 3D image, frontal max projection (top), cross-section at plane 68 in two magnified crops (middle, bottom). For Lucy Richardson the best result is shown after 2 iterations. Our algorithm uses loss from Eq. (3) with $\lambda_{inv} = 2$, $\lambda_{bound} = 0$.

Table S2. Our best-performing method for 3D data compared to Lucy Richardson (LR) baseline by denoising metrics (see Section 3 for details), training, and inference time. For all metrics except RMSE, higher is better. The best values are highlighted in bold. Our algorithm uses loss (3) with $\lambda_{inv} = 2$, $\lambda_{bound} = 0.1$. Training time is reported with and without FFT (in brackets)

	PSNR \uparrow	SSIM \uparrow	RMSE \downarrow	Train t (s)	Inference t (ms)
input	21.2	0.64	0.087	-	-
LR $n = 2$	22.4	0.32	0.076	-	69
LR $n = 5$	21.8	0.33	0.080	-	167
LR $n = 10$	21.5	0.44	0.084	-	222
LR $n = 20$	21.3	0.55	0.086	-	369
ours	24.1	0.39	0.062	8160 (18900)	734