Research Article Vol. 4, No. 11 / November 2017 / Optica 1437

Deep learning microscopy

Yair Rivenson,[1] Zoltán Göröcs,[1,2,3,†] Harun Günaydin,[1,†] Yibo Zhang,[1,2,3] Hongda Wang,[1,2,3] and Aydogan Ozcan[1,2,3,4,*]

[1]Electrical and Computer Engineering Department, University of California, Los Angeles, California 90095, USA
[2]Bioengineering Department, University of California, Los Angeles, California 90095, USA
[3]California NanoSystems Institute (CNSI), University of California, Los Angeles, California 90095, USA
[4]Department of Surgery, David Geffen School of Medicine, University of California, Los Angeles, California 90095, USA
*Corresponding author: ozcan@ucla.edu

Received 11 September 2017; revised 27 October 2017; accepted 29 October 2017 (Doc. ID 306825); published 20 November 2017

We demonstrate that a deep neural network can significantly improve optical microscopy, enhancing its spatial resolution over a large field of view and depth of field. After its training, the only input to this network is an image acquired using a regular optical microscope, without any changes to its design. We blindly tested this deep learning approach using various tissue samples that are imaged with low-resolution and wide-field systems, where the network rapidly outputs an image with better resolution, matching the performance of higher numerical aperture lenses and also significantly surpassing their limited field of view and depth of field. These results are significant for various fields that use microscopy tools, including, e.g., life sciences, where optical microscopy is considered as one of the most widely used and deployed techniques. Beyond such applications, the presented approach might be applicable to other imaging modalities, also spanning different parts of the electromagnetic spectrum, and can be used to design computational imagers that get better as they continue to image specimens and establish new transformations among different modes of imaging.

OCIS codes: (180.0180) Microscopy; (100.3010) Image reconstruction techniques; (100.4996) Pattern recognition, neural networks; (100.3190) Inverse problems; (110.1758) Computational imaging.

https://doi.org/10.1364/OPTICA.4.001437

1. INTRODUCTION

Deep learning is a class of machine learning techniques that uses multilayered artificial neural networks for automated analysis of signals or data [1,2]. The name comes from the general structure of deep neural networks, which consist of several layers of artificial neurons stacked over each other. One type of a deep neural network is the deep convolutional neural network (CNN). Typically, an individual layer of a deep convolutional network is composed of a convolutional layer and a nonlinear operator. The kernels (filters) in these convolutional layers are randomly initialized and can then be trained to learn how to perform specific tasks using supervised or unsupervised machine learning techniques. CNNs form a rapidly growing research field with various applications in, e.g., image classification [3], annotation [4], style transfer [5], compression [6], and deconvolution in photography [7–10], among others [11–14]. Recently, deep neural networks have also been successfully applied to solve numerous imaging-related problems in, e.g., computed tomography [15], magnetic resonance imaging [16], photoacoustic tomography [17], and phase retrieval [18], among others.

Here, we demonstrate the use of a deep neural network to significantly enhance the performance of an optical microscope without changing its design or hardware. This network uses a single image that is acquired under a standard microscope as input, and quickly outputs an improved image of the same specimen, e.g., in less than 1 s using a laptop, matching the resolution of higher-numerical-aperture (NA) objectives, while at the same time surpassing their limited field of view (FOV) and depth of field (DOF). The first step in this deep-learning-based microscopy framework involves learning the statistical transformation between low-resolution and high-resolution microscopic images, which is used to train a CNN. Normally, this transformation can be physically understood as a spatial convolution operation followed by an under-sampling step (going from a high-resolution and high-magnification microscopic image to a low-resolution and low-magnification one). However, the proposed CNN framework instead focuses on training multiple layers of artificial neural networks to statistically relate low-resolution images (input) to high-resolution images (output) of a specimen. In fact, to train and blindly test this deep-learning-based imaging framework, we have chosen bright-field microscopy with spatially and temporally incoherent broadband illumination, which presents challenges to provide an exact analytical or numerical modelling of light–sample interaction and the related physical image formation process, making the relationship between high-resolution images and low-resolution ones significantly more complicated to exactly model or predict. Although bright-field microscopic imaging has been our focus in this paper, the same deep learning framework

might be applicable to other microscopy modalities, including, e.g., holography, dark-field, fluorescence, multi-photon, optical coherence tomography, among others.

2. METHODS

Sample Preparation: A de-identified formalin-fixed paraffin-embedded (FFPE) hematoxylin and eosin (H&E)-stained human breast tissue section from a breast cancer patient, a Masson's-trichrome-stained lung tissue section from two pneumonia patients, and a Masson's-trichrome-stained kidney tissue section from a moderately advanced diabetic nephropathy patient were obtained from the Translational Pathology Core Laboratory at UCLA. Sample staining was done at the Histology Lab at UCLA. All the samples were obtained after de-identification of the patient and related information and were prepared from the existing specimen. Therefore, this work did not interfere with standard practices of care or sample collection procedures.

Microscopic Imaging: Image data acquisition was performed using an Olympus IX83 microscope equipped with a motorized stage and controlled by MetaMorph microscope automation software (Molecular Devices, LLC). The images were acquired using a set of Super Apochromat objectives, (UPLSAPO 40X2 / 0.95NA, 100XO / 1.4NA—oil immersion objective lens). The color images were obtained using a Qimaging Retiga 4000R camera with a pixel size of 7.4 μm.

3. RESULTS

To initially train the deep neural network, we acquired microscopy images of Masson's-trichrome-stained lung tissue sections using a pathology slide, obtained from an anonymous pneumonia patient. The lower-resolution images were acquired with a 40 × /0.95NA objective lens, providing a FOV of 150 μm × 150 μm per image, while the higher-resolution training images were acquired with a 100 × /1.4NA oil-immersion objective lens, providing a FOV of 60 μm × 60 μm per image, i.e., 6.25-fold smaller in area. Both the low-resolution and high-resolution images were acquired with 0.55-NA condenser illumination, leading to a diffraction-limited resolution of ∼0.36 μm and ∼0.28 μm, respectively, both of which were adequately sampled by the image sensor chip, with an "effective" pixel size of ∼0.18 μm and ∼0.07 μm, respectively. Following a digital registration procedure to match the corresponding FOVs of each set of images (Section 2 in Supplement 1), we generated 179 low-resolution images corresponding to different regions of the lung tissue sample, which were used as input to our network, together with their corresponding high-resolution labels for each FOV. Out of these images, 149 low-resolution input images and their corresponding high-resolution labels were randomly selected to be used as our training image set, while 10 low-resolution images and their corresponding high-resolution labels were used for selecting and validating the final network model, and the remaining 20 low-resolution inputs and their corresponding high-resolution labels formed our test images used to blindly quantify the average performance of the final network (see the structural similarity index, SSIM, reported in Table S1 in Supplement 1). This training dataset was further augmented by extracting 60 × 60-pixel and 150 × 150-pixel image patches with 40% overlap from the low-resolution and high-resolution images, respectively, which effectively increased our training data size by more than 6-fold.

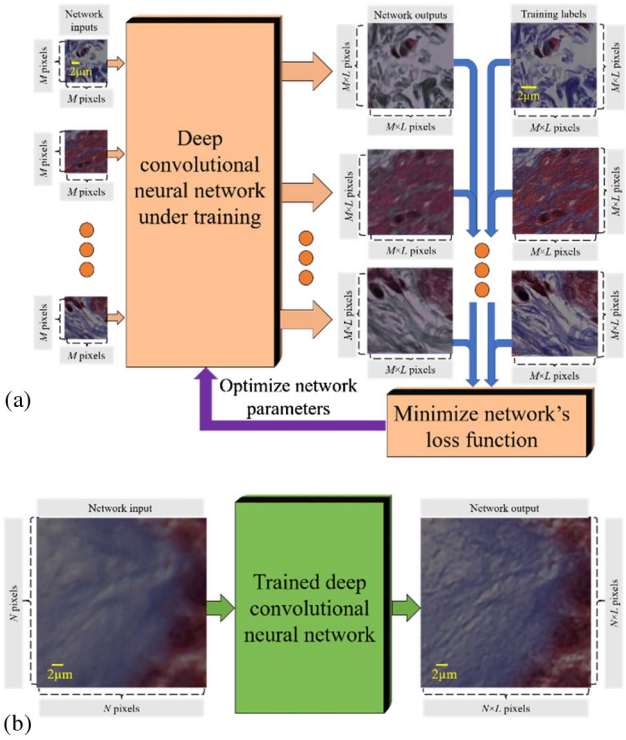

Fig. 1. Schematics of the deep neural network trained for microscopic imaging. (**a**) The input is composed of a set of lower-resolution images, and the training labels are their corresponding high-resolution images. The deep neural network is trained by optimizing various parameters, which minimize the loss function between the network's output and the corresponding high-resolution training labels. (**b**) After the training phase is complete, the network is blindly given an $N \times N$ pixel input image and rapidly outputs an $(N \times L) \times (N \times L)$ image, showing improved spatial resolution, field of view, and depth of field.

As shown in Fig. 1(a) and further detailed in Section 1 in Supplement 1, these training image patches were randomly assigned to 149 batches, each containing 64 randomly drawn low- and high-resolution image pairs, forming a total of 9,536 input patches for the network training process (Section 3 in Supplement 1, Table S2 in Supplement 1). The pixel count and the number of the image patches were empirically determined to allow rapid training of the network, while at the same time containing distinct sample features in each patch. In this training phase, as further detailed in the supplementary information, we utilized an optimization algorithm to adjust the network's parameters using the training image set and utilized the validation image set to determine the best network model, also helping to avoid overfitting to the training image data.

After this training procedure, which needs to be performed only once, the CNN is fixed (Fig. 1(b), Sections 1 and 4 in Supplement 1) and ready to blindly output high-resolution images of samples of any type, i.e., not necessarily from the same tissue type that the CNN has been trained on. To demonstrate the success of this deep-learning-enhanced microscopy approach, first we blindly tested the network's model on entirely different sections of Masson's-trichrome-stained lung tissue, which were not used in our training process, and in fact were taken from another anonymous patient. These samples were imaged using the same 40 × /0.95NA and 100 × /1.4NA objective lenses with

0.55-NA condenser illumination, generating various input images for our CNN. The output images of the CNN for these input images are summarized in Fig. 2, which clearly demonstrate the ability of the network to significantly enhance the spatial resolution of the input images, whether or not they were initially acquired with a 40 × /0.95NA or a 100 × /1.4NA objective lens. For the network output image shown in Fig. 2(a), we used an input image acquired with a 40 × /0.95NA objective lens, and therefore it has a FOV that is 6.25-fold larger compared to the 100 × objective lens FOV, which is highlighted with a red box in Fig. 2(a). Zoomed-in regions of interest (ROI) corresponding to various input and output images are also shown in Figs. 2(b)–2(p), better illustrating the fine spatial improvements in the network output images compared to the corresponding

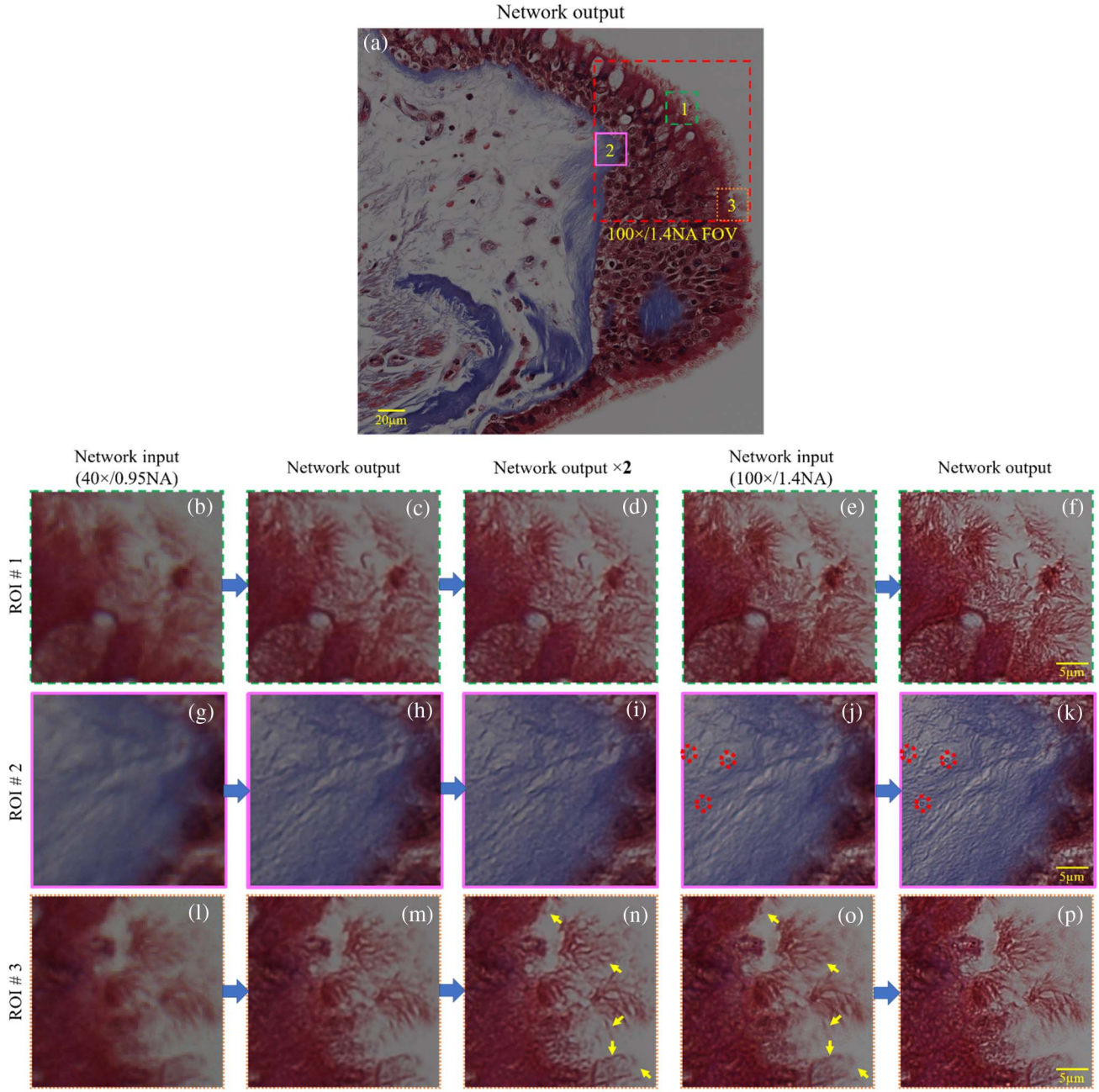

Fig. 2. Deep neural network output image corresponding to a Masson's-trichrome-stained lung tissue section taken from a pneumonia patient. The network was trained on images of a Masson's-trichrome-stained lung tissue sample taken from another patient. (**a**) Image of the deep neural network output corresponding to a 40 × /0.95NA input image. The red highlighted region denotes the FOV of a 100 × /1.4NA objective lens. (**b, g, l**) Zoomed-in regions of interest (ROIs) of the input image (40 × /0.95NA). (**c, h, m**) Zoomed-in ROIs of the neural network output image. (**d, i, n**) Zoomed-in ROIs of the neural network output image, taking the first output of the network, shown in (**c**, **h**) and (**m**), as input. (**e, j, o**) Comparison images of the same ROIs, acquired using a 100 × /1.4NA objective lens (also see Fig. S7 in Supplement 1 for difference maps). (**f, k, p**) Result of the same deep neural network model applied on the 100 × /1.4NA objective lens images (also see Fig. S8 in Supplement 1). The yellow arrows in (**o**) point to some of the out-of-focus features that are brought to focus in the network output image shown in (**n**). Red circles in (**j, k**) point to some dust particles in the images acquired with our 100 × /1.4NA objective lens, and that is why they do not appear in (**g–i**). The average network computation time for different ROIs is listed in Table S3 in Supplement 1.

input images. To give an example of the computational load of this approach, the network output images shown in Fig. 2(a) and Figs. 2(c), 2(h), and 2(m) (with FOVs of 378.8 × 378.8 μm and 29.6 × 29.6 μm, respectively) took on average ∼0.695 s and 0.037 s, respectively, to compute using a dual graphics processing unit (GPU) running on a laptop computer (see Section 5 and Table S3 in Supplement 1).

In Fig. 2, we also illustrate that "self-feeding" the output of the network as its new input significantly improves the resulting output image, as demonstrated in Figs. 2(d), 2(i), and 2(n). A minor disadvantage of this self-feeding approach is increased computation time, e.g., ∼0.062 s on average for Figs. 2(d), 2(i), and 2(n) on the same laptop computer, in comparison to ∼0.037 s on average for Figs. 2(c), 2(h), and 2(m) (see Section 5 and Table S3 in

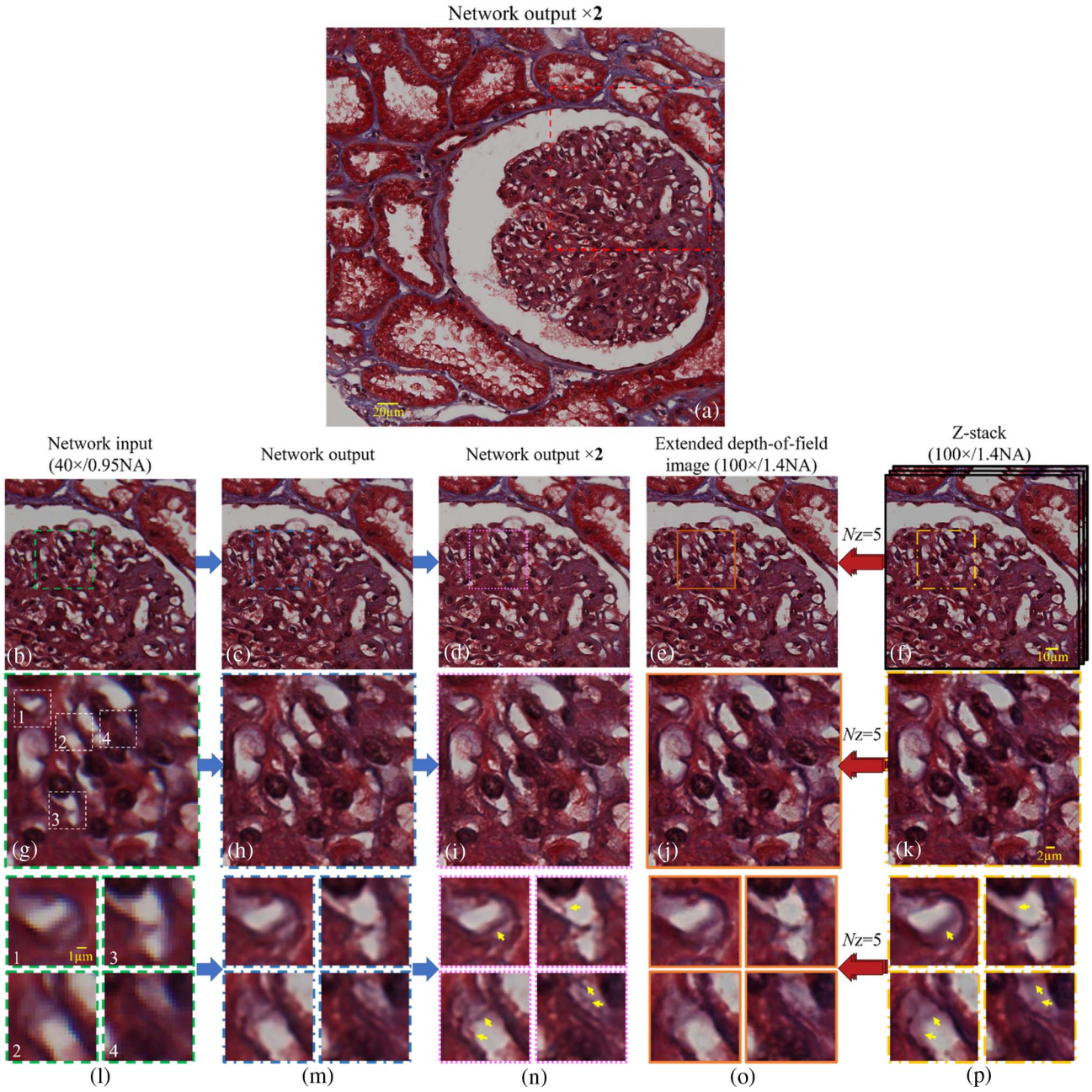

Fig. 3. Deep neural network output image of a Masson's-trichrome-stained *kidney* tissue section obtained from a moderately advanced diabetic nephropathy patient. The network was trained on images of a Masson's-trichrome-stained *lung* tissue taken from another patient. (**a**) Result of two successive applications of the same deep neural network on a 40 × /0.95NA image of the kidney tissue that is used as input. The red highlighted region denotes the FOV of a 100 × /1.4NA objective lens. (**b**, **g**, **l**) Zoomed-in ROIs of the input image (40 × /0.95NA). (**c**, **h**, **m**) Zoomed-in ROIs of the neural network output image, taking the corresponding 40 × /0.95NA images as input. (**d**, **i**, **n**) Zoomed-in ROIs of the neural network output image, taking the first output of the network, shown in (**c**, **h, m**) as input. (**e**, **j**, **o**) Extended depth-of-field image, algorithmically calculated using $N_z = 5$ images taken at different depths using a 100 × /1.4NA objective lens. (**f**, **k**, **p**) The auto-focused images of the same ROIs, acquired using a 100 × /1.4NA objective lens. The yellow arrows in (**p**) point to some of the out-of-focus features that are brought to focus in the network output images shown in (**n**). Also see Fig. S8 in Supplement 1.

Supplement 1). After one cycle of feeding the network with its own output, the next cycles of self-feeding do not change the output images in a noticeable manner, as also highlighted in Fig. S6 in Supplement 1.

Quite interestingly, when we use the same deep neural network model on input images acquired with a 100 × /1.4NA objective lens, the network output also demonstrates significant enhancement in spatial details that appear blurry in the original input images. These results are demonstrated in Figs. 2(f), 2(k), and 2(p) and in Fig. S8 in Supplement 1, revealing that the same learned model (which was trained on the transformation of 40 × /0.95NA images into 100 × /1.4NA images) can also be used to super-resolve images that were captured with higher-magnification and higher-numerical-aperture lenses compared to the input images of the training model. This feature suggests the scale-invariance of the image transformation (from lower-resolution input images to higher-resolution ones) that the CNN is trained on.

Next, we blindly applied the same lung-tissue-trained CNN for improving the microscopic images of a Masson's-trichrome-stained kidney tissue section obtained from an anonymous moderately advanced diabetic nephropathy patient. The network output images shown in Fig. 3 emphasize several important features of our deep-learning-based microscopy framework. First, this tissue type, although stained with the same dye (Masson's trichrome) is entirely new to our lung-tissue-trained CNN, and yet, the output images clearly show a similarly outstanding performance as in Fig. 2. Second, similar to the results shown in Fig. 2, self-feeding the output of the same lung tissue network as a fresh input back to the network further improves our reconstructed images, even for a kidney tissue that has not been part of our training process; see, e.g., Figs. 3(d), 3(i), and 3(n). Third, the output images of our deep learning model also exhibit a significantly larger DOF. To better illustrate this, the output image of the lung-tissue-trained CNN on a kidney tissue section imaged with a 40 × /0.95NA objective was compared to an extended DOF image, which was obtained by using a depth-resolved stack of five images acquired using a 100 × /1.4NA objective lens (with 0.4-μm axial increments). To create the gold standard, i.e., the extended DOF image used for comparison to our network output, we merged these five depth-resolved images acquired with a 100 × /1.4NA objective lens using a wavelet-based depth-fusion algorithm [19]. The network's output images, shown in Figs. 3(d), 3(i), and 3(n), clearly demonstrate that several spatial features of the sample that appear in focus in the deep network output image can only be inferred by acquiring a depth-resolved stack of 100 × /1.4NA objective images because of the shallow DOF

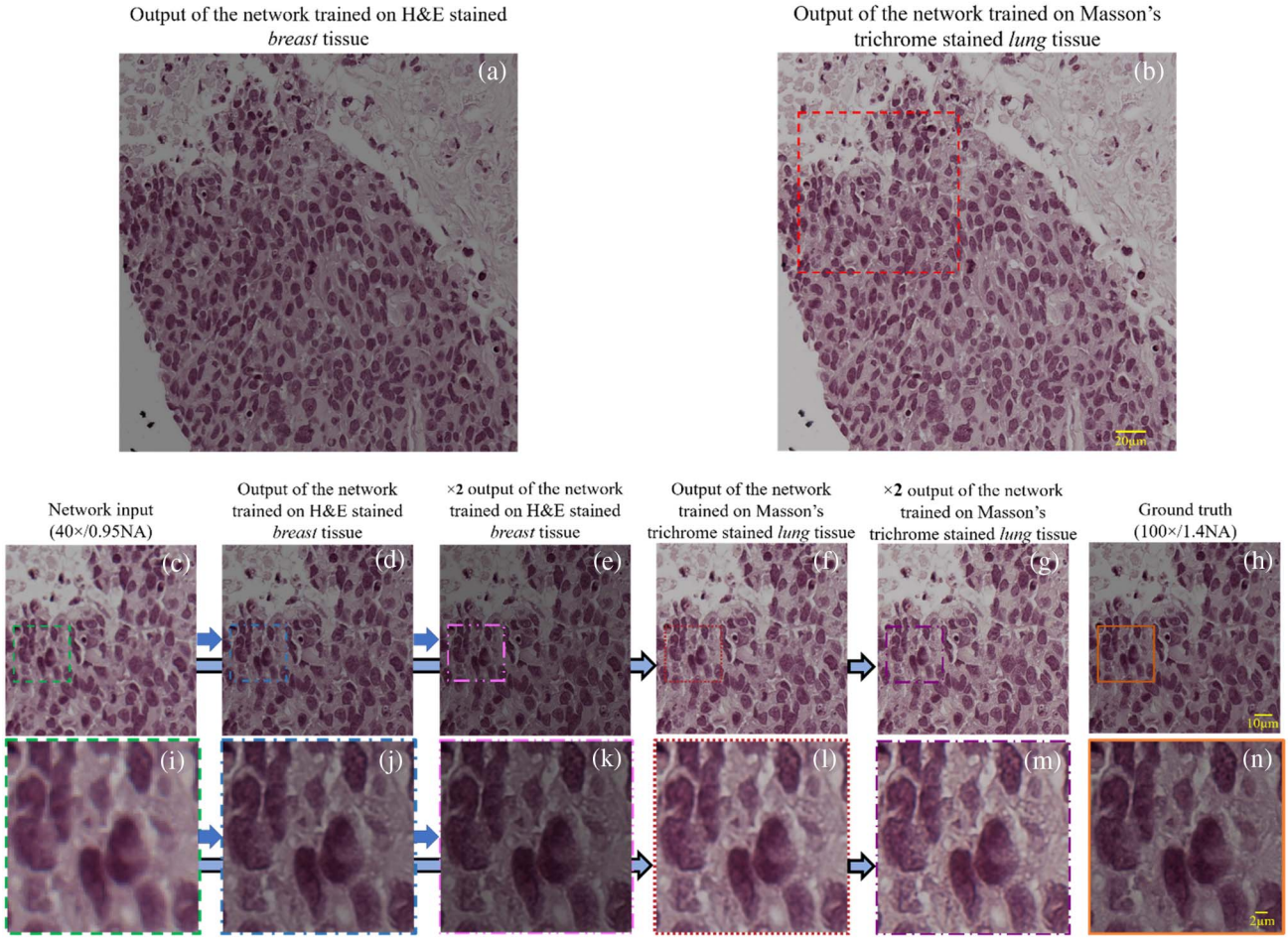

Fig. 4. Deep-neural-network-based imaging of H&E-stained breast tissue section. The output images of two different deep neural networks are compared to each other. (**a**) The first network is trained on H&E-stained breast tissue, taken from a different tissue section that is not used in the training phase. (**b**) The second network is trained on a different tissue type and stain, i.e., Masson's-trichrome-stained lung tissue sections. (**c**–**n**) Illustrate zoomed-in images of different ROIs of the input and output images, similar to Figs. 2–3. A similar comparison is also provided in Fig. S9 in Supplement 1.

of such high-NA objective lenses—also see the yellow pointers in Figs. 3(n) and 3(p) to better visualize this DOF enhancement. Stated differently, the network output image not only has 6.25-fold larger FOV (~379 × 379 μm) compared to the images of a 100 × /1.4NA objective lens, but it also exhibits a significantly enhanced DOF. The same extended DOF feature of the deep neural network image inference is further demonstrated using lung tissue samples shown in Figs. 2(n) and 2(o).

Until now, we have focused on bright-field microscopic images of different tissue types, all stained with the same dye (Masson's trichrome), and used a deep neural network to blindly transform lower-resolution images of these tissue samples into higher-resolution ones, also showing significant enhancement in FOV and DOF of the output images. Next, we tested to see if a CNN that is trained on one type of stain can be applied to other tissue types that are stained with another dye. To investigate this, we trained a new CNN model (with the same network architecture) using microscopic images of a hematoxylin and eosin (H&E)-stained human breast tissue section obtained from an anonymous breast cancer patient. As before, the training pairs were created from 40 × /0.95NA lower-resolution images and 100 × /1.4NA high-resolution images (see Tables S1, S2 in Supplement 1 for specific implementation details). First, we blindly tested the results of this trained deep neural network on images of breast tissue samples (which were not part of the network training process) acquired using a 40 × /0.95NA objective lens. Figure 4 illustrates the success of this blind testing phase, which is expected since this network has been trained on the same type of stain and tissue (i.e., H&E-stained breast tissue). To compare, in the same Fig. 4 we also report the output images of a previously used deep neural network model (trained using lung tissue sections stained with Masson's trichrome) for the same input images reported in Fig. 4. Except a relatively minor color distortion, all the spatial features of the H&E-stained breast tissue sample have been resolved using a CNN trained on Masson's-trichrome-stained lung tissue. These results, together with the earlier ones discussed so far, clearly demonstrate the universality of the deep neural network approach, and how it can be used to output enhanced microscopic images of various types of samples from different patients and organs and using different types of stains. A similarly outstanding result, with the same conclusion, is provided in Fig. S9 in Supplement 1, where the deep learning network trained on H&E-stained breast tissue images was applied on Masson's-trichrome-stained lung tissue samples imaged using a 40 × /0.95NA objective lens, representing the opposite case of Fig. 4. To mitigate possible color distortions when inferring images that are stained differently compared to the training image set, one can train a universal network with various types of samples, as demonstrated in, e.g., Ref. [18] for holography and phase recovery. Such an approach would then increase the number of feature maps and the overall complexity of the network [18].

4. DISCUSSION

To quantify the effect of our deep neural network on the spatial frequencies of the output image, we have applied the CNN that was trained using the lung tissue model on a resolution test target, which was imaged using a 100 × /1.4NA objective lens with a 0.55-NA condenser. The objective lens was oil-immersed as depicted in Fig. 5(a), while the interface between the resolution test target and the sample cover glass was not oil-immersed, leading to an effective NA of ≤1 and a lateral diffraction-limited resolution of ≥0.355 μm. The modulation transfer function (MTF) was evaluated by calculating the contrast of different elements of the resolution test target (Section 6 in Supplement 1). Based on this experimental analysis, the MTFs for the input image and the output image of the deep neural network that was trained on lung tissue are compared to each other in Fig. 5(e) and Table S4 in Supplement 1. The output image of the deep neural network, despite the fact that it was trained on tissue samples imaged with a 40 × /0.95NA objective lens, shows an increased modulation contrast for a significant portion of the spatial frequency spectrum at especially high frequencies, while also resolving a period of 0.345 μm (Table S4 in Supplement 1).

To conclude, we have demonstrated how deep learning significantly enhances optical microscopy images by improving their resolution, FOV, and DOF. This deep learning approach is extremely fast to output an improved image, e.g., taking on average ~0.69 s per image with a FOV of ~379 × 379 μm even using a laptop computer, and only needs a single image taken with a standard optical microscope without the need for extra hardware or user-specified post-processing. After appropriate training, this framework and its derivatives might be applicable to other forms of optical microscopy and imaging techniques and can be used to transfer images that are acquired under low-resolution systems into high-resolution and wide-field images, significantly extending the space bandwidth product of the output images. Furthermore, using the same deep learning approach we have also demonstrated the extension of the spatial frequency response of the imaging system along with an extended DOF. In addition to

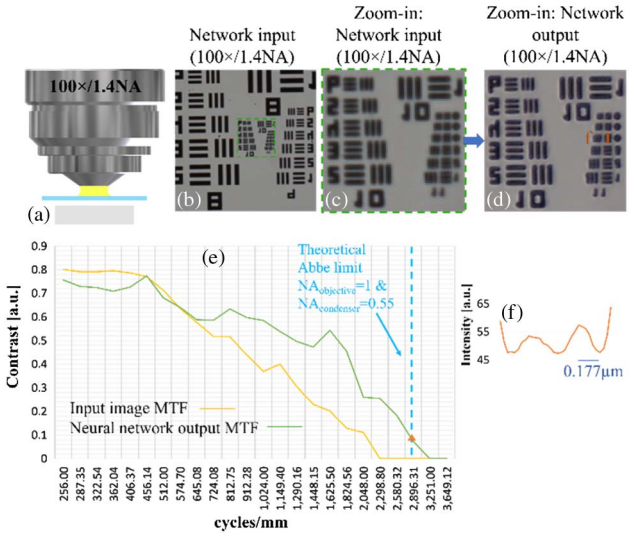

Fig. 5. Modulation transfer function (MTF) comparison for the input image and the output image of a deep neural network that is trained on images of a lung tissue section. (**a**) Experimental apparatus: the US Air Force (USAF) resolution target lies on a glass slide with an air gap in between, leading to an effective numerical aperture of ≤1. The resolution test target was illuminated using a condenser with a numerical aperture of 0.55, leading to lateral diffraction-limited resolution of ≥0.355 μm. (**b**) Input image acquired with a 100 × /1.4NA lens. (**c**), Zoom-in on the green highlighted ROI highlighted in (**b**). (**d**) Output image of the deep neural network applied on (**b**, **c**). (**e**) MTF calculated from the input and output images of the deep network. (**f**) Cross-sectional profile of group 11, element 4 (period: 0.345 μm) extracted from the network output image shown in (**d**).

optical microscopy, this entire framework can also be applied to other computational imaging approaches, also spanning different parts of the electromagnetic spectrum, and can be used to design computational imagers with improved resolution, FOV, and DOF.

Funding. Presidential Early Career Award for Scientists and Engineers (PECASE); Army Research Office (ARO) (W911NF-13-1-0419, W911NF-13-1-0197); ARO Life Sciences Division; National Science Foundation (NSF) (0963183); Division of Chemical, Bioengineering, Environmental, and Transport Systems (CBET) Division Biophotonics Program; Division of Emerging Frontiers in Research and Innovation (EFRI), NSF EAGER Award, NSF INSPIRE Award, NSF Partnerships for Innovation: Building Innovation Capacity (PFI:BIC) Program; Office of Naval Research (ONR); National Institutes of Health (NIH); Howard Hughes Medical Institute (HHMI); Vodafone Foundation; Mary Kay Foundation (TMKF); Steven & Alexandra Cohen Foundation; King Abdullah University of Science and Technology (KAUST); American Recovery and Reinvestment Act of 2009 (ARRA). European Union's Horizon 2020 Framework Programme (H2020); H2020 Marie Skłodowska-Curie Actions (MSCA) (H2020-MSCA-IF-2014-65959).

See Supplement 1 for supporting content.

†These authors contributed equally to this work.

REFERENCES

1. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature **521**, 436–444 (2015).
2. J. Schmidhuber, "Deep learning in neural networks: an overview," Neural Netw. **61**, 85–117 (2015).
3. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds. (Curran Associates, Inc., 2012), pp. 1097–1105.
4. V. N. Murthy, S. Maji, and R. Manmatha, "Automatic image annotation using deep learning representations," in *5th ACM on International Conference on Multimedia Retrieval, ICMR'15* (ACM, 2015), pp. 603–606.
5. L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 2414–2423.
6. C. Dong, Y. Deng, C. Change Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 576–584.
7. J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 1646–1654.
8. C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," IEEE Trans. Pattern Anal. Mach. Intell. **38**, 295–307 (2016).
9. W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 1874–1883.
10. M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand, "Deep bilateral learning for real-time image enhancement," ACM Trans. Graph. **36**, 118 (2017).
11. V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, R. Kim, R. Raman, P. C. Nelson, J. L. Mega, and D. R. Webster, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," JAMA, J. Am. Med. Assoc. **316**, 2402–2410 (2016).
12. D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," Nature **529**, 484–489 (2016).
13. N. Jean, M. Burke, M. Xie, W. M. Davis, D. B. Lobell, and S. Ermon, "Combining satellite imagery and machine learning to predict poverty," Science **353**, 790–794 (2016).
14. A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," Nature **542**, 115–118 (2017).
15. K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," IEEE Trans. Image Process. **26**, 4509–4522 (2017).
16. S. Wang, Z. Su, L. Ying, X. Peng, S. Zhu, F. Liang, D. Feng, and D. Liang, "Accelerating magnetic resonance imaging via deep learning," in *IEEE 13th International Symposium on Biomedical Imaging (ISBI)* (2016), pp. 514–517.
17. S. Antholzer, M. Haltmeier, and J. Schwab, "Deep learning for photoacoustic tomography from sparse data," arXiv preprint arXiv:1704.04587 (2017).
18. Y. Rivenson, Y. Zhang, H. Gunaydin, D. Teng, and A. Ozcan, "Phase recovery and holographic image reconstruction using deep learning in neural networks," Light: Sci. Appl. **7**, e17141 (2018).
19. B. Forster, D. Van De Ville, J. Berent, D. Sage, and M. Unser, "Complex wavelets for extended depth-of-field: a new method for the fusion of multichannel microscopy images," Microsc. Res. Tech. **65**, 33–42 (2004).

optica

Deep learning microscopy: supplementary material

YAIR RIVENSON,[1] ZOLTÁN GÖRÖCS,[1,2,3,†] HARUN GÜNAYDIN,[1,†] YIBO ZHANG,[1,2,3] HONGDA WANG,[1,2,3] AYDOGAN OZCAN,[1,2,3,4,*]

[1]*Electrical Engineering Department, University of California, Los Angeles, CA 90095, USA*
[2]*Bioengineering Department, University of California, Los Angeles, CA 90095, USA*
[3]*California NanoSystems Institute (CNSI), University of California, Los Angeles, CA 90095, USA*
[4]*Department of Surgery, David Geffen School of Medicine, University of California, Los Angeles, CA 90095, USA*
[†]*These authors contributed equally to the paper*
**Corresponding author: ozcan@ucla.edu*

Published 20 November 2017

This document provides supplementary information to "Deep learning microscopy," https://doi.org/10.1364/optica.4.001437.

https://doi.org/10.6084/m9.figshare.5552338

1. DEEP LEARNING NETWORK ARCHITECTURE

The schematics of the architecture for training our deep neural network is depicted in Fig. S1. The input images are mapped into 3 color channels: red, green and blue (RGB). The input convolutional layer maps the 3 input color channels, into 32 channels, as depicted in Fig. S2. The number of output channels of the first convolutional layer was empirically determined to provide the optimal balance between the deep neural network's size (which affects the computational complexity and image output time) and its image transform performance. The input convolutional layer is followed by K=5 residual blocks [1]. Each residual block is composed of 2 convolutional layers and 2 rectified linear units (ReLU) [2,3], as shown in Fig. S1. The ReLU is an activation function which performs $\mathrm{ReLU}(x) = \max(0, x)$. The formula of each residual block can be summarized as:

$$X_{k+1} = X_k + \mathrm{ReLU}(\mathrm{ReLU}(X_k * W_k^{(1)}) * W_k^{(2)}), \quad \textbf{(S1)}$$

where $*$ refers to convolution operation, X_k is the input to the k-th block, X_{k+1} denotes its output, $W_k^{(1)}$ and $W_k^{(2)}$ denote an ensemble of learnable convolution kernels of the k-th block, where the bias terms are omitted for simplicity. The output feature maps of the convolutional layers in the network are calculated as follows:

$$g_{k,j} = \sum_i f_{k,i} * w_{k,i,j} + \beta_{k,j}\mathbf{\Omega}, \quad \textbf{(S2)}$$

where $w_{k,i,j}$ is a learnable 2D kernel (i.e., the (i,j)-th kernel of W_k) applied to the i-th input feature map, $f_{k,i}$ (which is an M×M-pixel image in the residual blocks), $\beta_{k,j}$ is a learnable bias term, $\mathbf{\Omega}$ is an M×M matrix with all its entries set as 1, and $g_{k,j}$ is the convolutional layer j-th output feature map (which is also an M×M-pixel image in the residual blocks). The size of all the kernels (filters) used throughout the network's convolutional layers is 3×3. To resolve the dimensionality mismatch of Eq. (2), prior to convolution, the feature map $f_{k,i}$ is zero-padded to a size of (M+2)×(M+2) pixels, where only the central M×M-pixel part is taken following the convolution with kernel $w_{k,i,j}$.

To allow high level feature inference we increase the number of features learnt in each layer, by gradually increasing the number of channels, using the pyramidal network concept [4]. Using such pyramidal networks helps to keep the network's width compact in comparison to designs that sustain a constant number of channels throughout the network. The channel increase formula was empirically set according to [4]:

$$A_k = A_{k-1} + floor((\alpha \times k) / K + 0.5) \quad \textbf{(S3)}$$

where $A_0 = 32$, k=[1:5], which is the residual block number, K=5 is the total number of residual blocks used in our architecture and α is a constant that determines the number of channels that will be added at each residual block. In our implementation, we used α=10, which yields $A_5 = 62$ channels at the output of the final residual block. In addition, we utilized the concept of residual connections (shortcutting the block's input to its output, see Fig. S1), which was demonstrated to improve the training of deep neural networks by providing a clear path for information flow [3] and speed up the convergence of the training phase. Nevertheless,

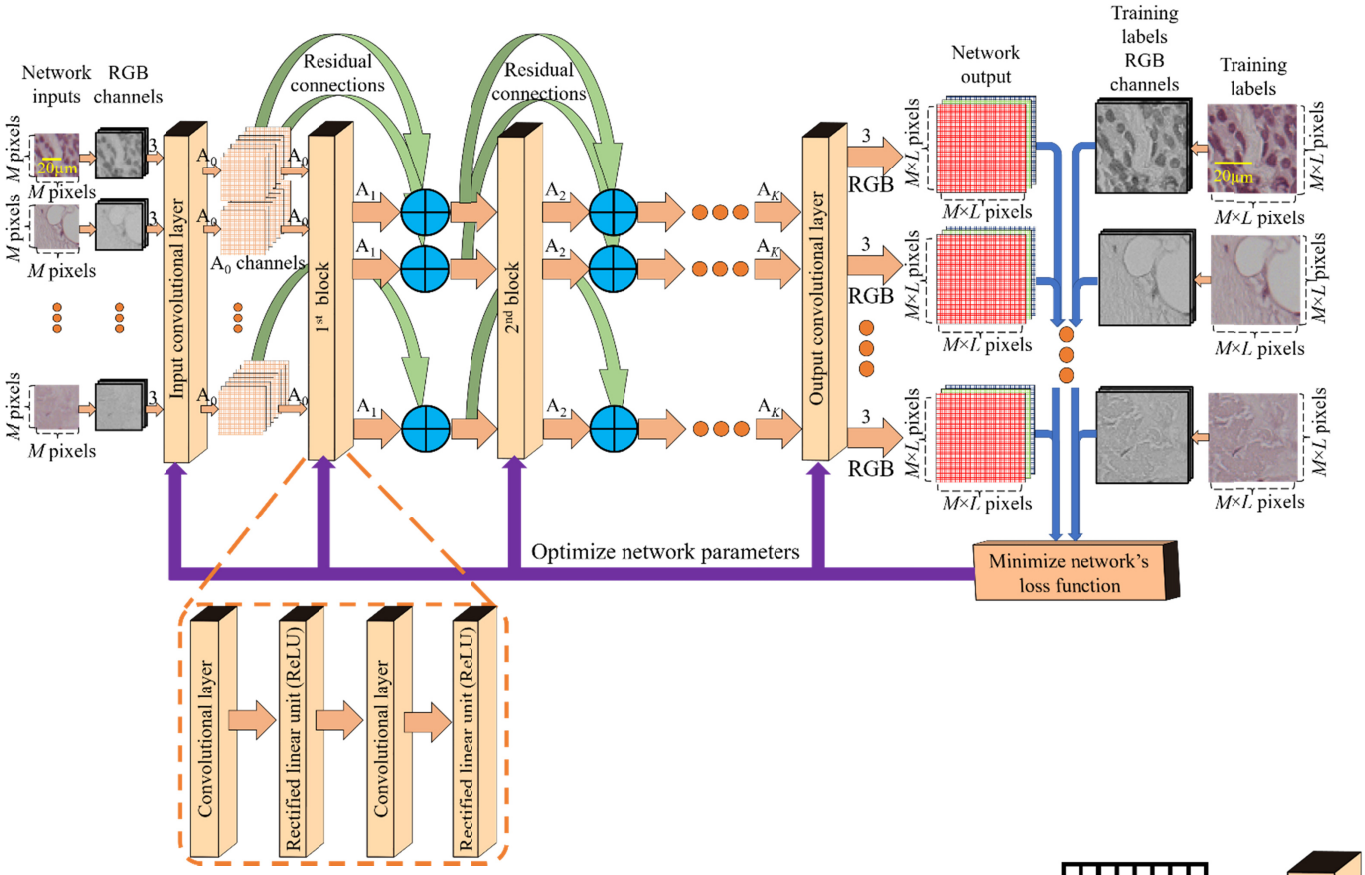

Fig. S1. Detailed schematics of the deep neural network training phase.

increasing the number of channels at the output of each layer leads to a dimensional mismatch between the inputs and outputs of a block, which are element-wise summed up in Eq. (1). This dimensional mismatch is resolved by augmenting each block's input channels with zero valued channels, which virtually equalizes the number of channels between a residual block input and output.

In our experiments, we have trained the deep neural network to extend the output image space-bandwidth-product by a non-integer factor of L^2=2.5^2=6.25 compared to the input images. To do so, first the network learns to enhance the input image by a factor of 5×5 pixels followed by a learnable down-sampling operator of 2×2, to obtain the desired L=2.5 factor (Fig. S3). More specifically, at the output of the K-th residual block $A_K = A_5 = 62$ channels are mapped to $3\times5^2 = 75$ channels (Fig. S3), followed by resampling of these 75 $(M\times M)$ pixels channels to three channels with $(M\times5)\times(M\times5)$ pixels grid [5,6]. These three $(M\times5)\times(M\times5)$ pixels channels are then used as input to an additional convolutional layer (with learnable kernels and biases, as the rest of the network), that two-times down-samples these images to three $(M\times2.5)\times(M\times2.5)$ color pixels. This is performed by using a two-pixel stride convolution, instead of a single pixel stride convolution, as performed throughout the other convolutional layers of the network. This way, the network learns the optimal down-sampling procedure for our microscopic imaging task. It is important to note that during the testing phase, if the number of input pixels to the network is odd, the resulting number of output image pixels will be determined by the ceiling operator. For instance, a 555×333-pixel input image will result in a 1388×833-pixel image for L=2.5.

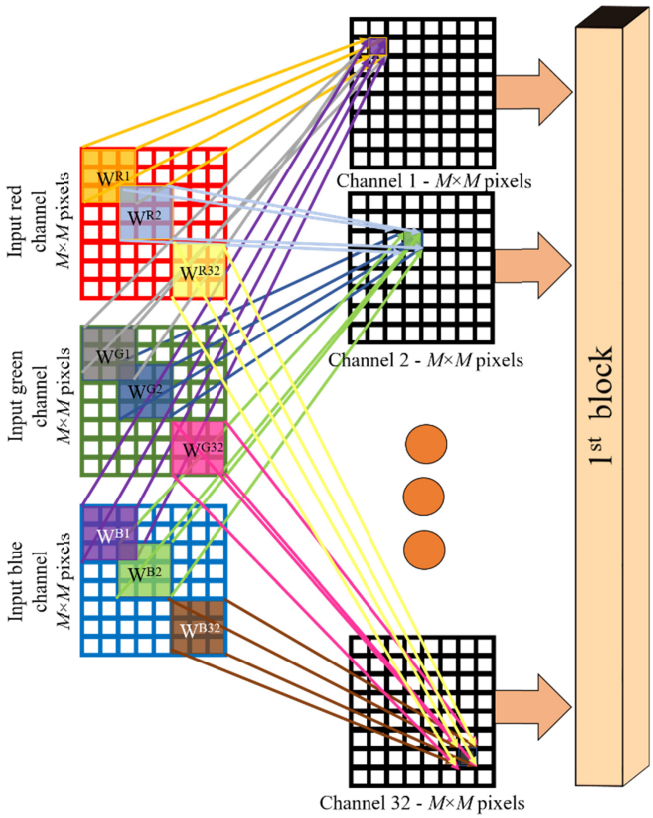

Fig. S2. Detailed schematics of the input layer of the deep neural network.

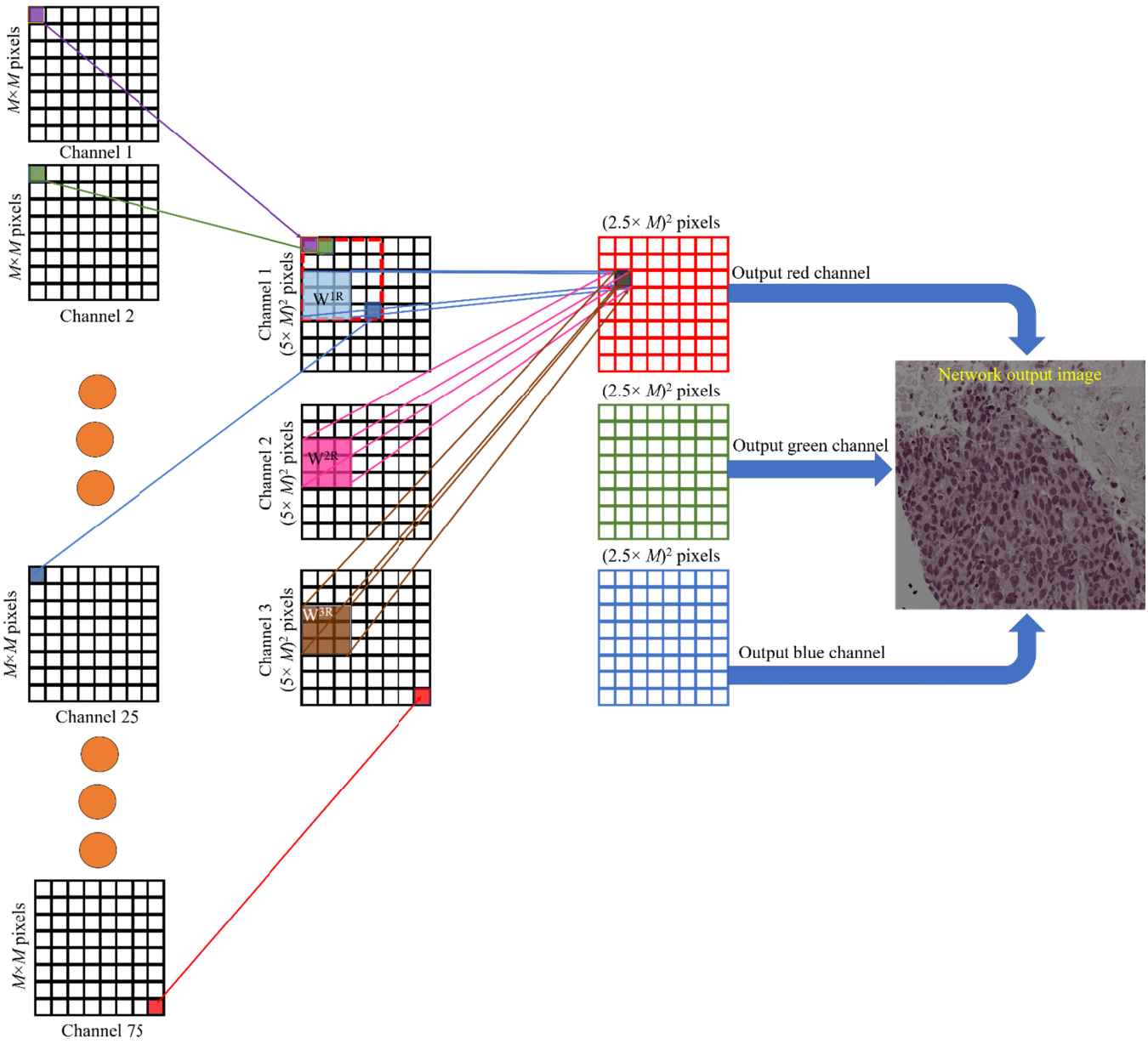

Fig. S3. Detailed schematics of the output layer of the deep neural network for L=2.5.

The above-discussed deep network architecture provides two major benefits: first, the up-sampling procedure becomes a learnable operation with supervised learning, and second, using low resolution images throughout the network's layers makes the time and memory complexities of the algorithm L^2 times smaller [6] when compared to approaches that up-sample the input image as a precursor to the deep neural network. This has a positive impact on the convergence speed of both the training and image transformation phases of our network.

2. DATA PRE-PROCESSING

To achieve optimal results, the network should be trained with accurately aligned low-resolution input images and high-resolution label image data. We match the corresponding input and label image pairs using the following steps: (A) Color images are converted to grayscale images. (B) A large field-of-view image is formed by stitching a set of low resolution images. (C) Each high-resolution label image is down-sampled (bicubic) by a factor L. This down-sampled image is used as a template image to find the highest correlation matching patch in the low-resolution stitched image. The highest correlating patch from the low-resolution stitched image is then digitally cropped. This cropped low-resolution image and the original high-resolution image, form an input-label pair, which is used for the network's training and testing. (D) Additional alignment is then performed on each of the

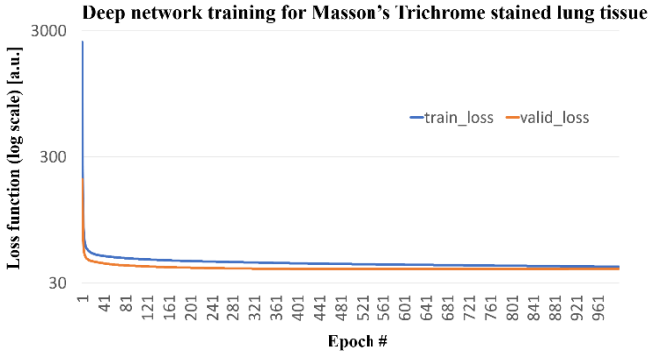

Fig S4. Training and validation dataset errors as a function of the number of epochs for the Masson's trichrome stained lung tissue dataset.

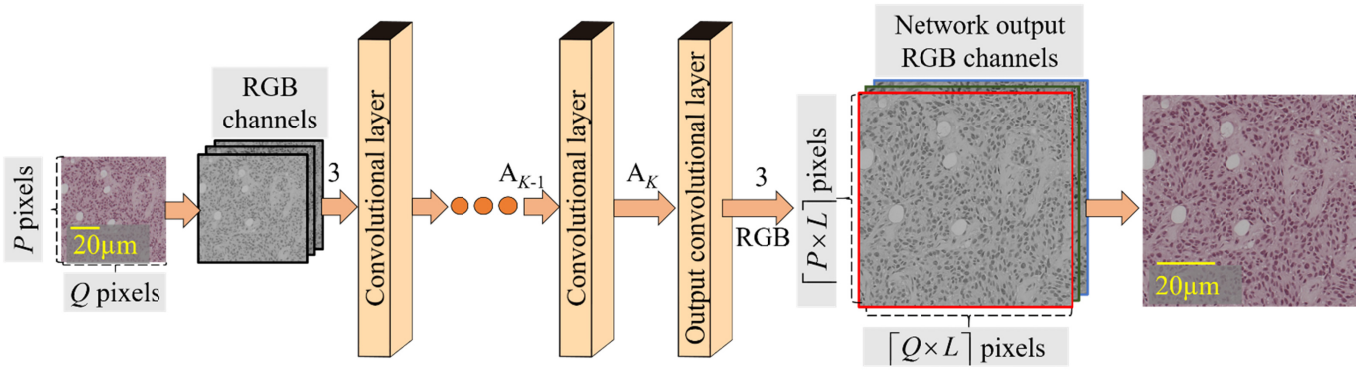

Fig. S5. Detailed schematics of the deep neural network high-resolution image inference (i.e., the testing phase).

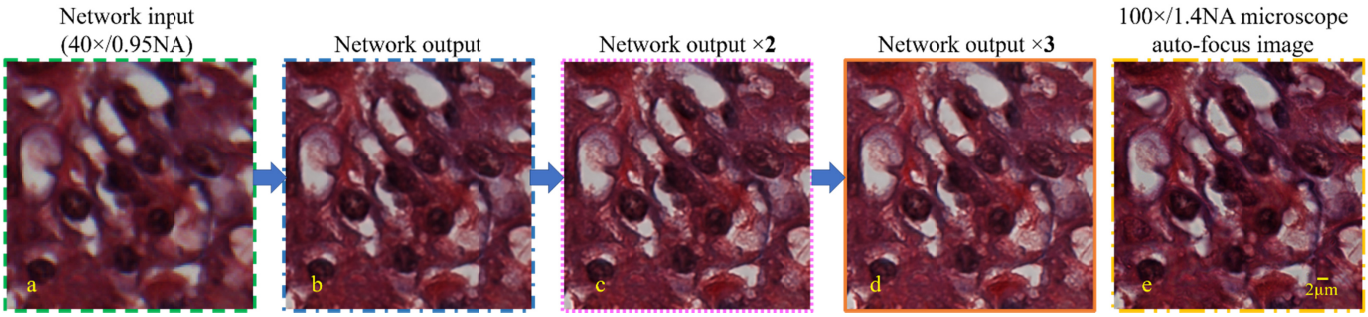

Fig. S6. Result of applying the deep neural network in a cyclic manner on Masson's trichrome stained kidney section images. (**a**), Input image acquired with a 40×/0.95NA objective lens. The deep neural network is applied on this input image once, twice and three times, where the results are shown in (**b**, **c)** and (**d**), respectively. (**e**), 100×/1.4NA image of the same field-of-view is shown for comparison.

input-label pairs to further refine the input-label matching, mitigating rotation, translation and scaling discrepancies between the lower resolution and higher resolution images. This step was performed by matching Speeded-Up Robust Features (SURF) [7], implemented using Matlab [8].

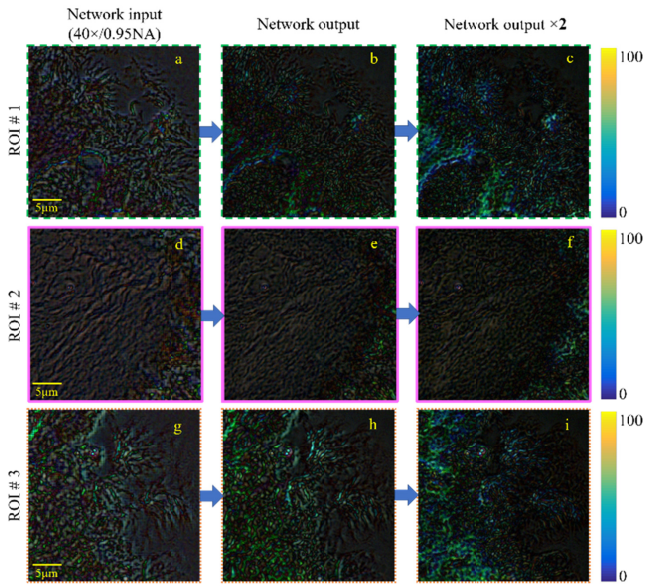

Fig S7. The percentage of the pixel-level differences for the network input or output images calculated with respect to the gold standard images captured using a 100×/1.4NA objective lens. ROIs correspond to the Masson's trichrome stained lung tissue shown in Fig. 2 of the main text. The colorbar spans 0-100%.

3. NETWORK TRAINING

The network was trained by optimizing the following loss function (ℓ) given the high-resolution training labels Y^{HR}:

$$\ell(\Theta)=\frac{1}{3\times M^2\times L^2}\sum_{c=1}^{3}\sum_{u=1}^{M\times L}\sum_{v=1}^{M\times L}\left\|Y^{\Theta}_{c,u,v}-Y^{HR}_{c,u,v}\right\|_2^2+ \\ +\lambda\frac{1}{3\times M^2\times L^2}\sum_{c=1}^{3}\sum_{u=1}^{M\times L}\sum_{v=1}^{M\times L}\left|\nabla Y^{\Theta}\right|^2_{c,u,v}, \quad \textbf{(S4)}$$

where $Y^{\Theta}_{c,u,v}$ and $Y^{HR}_{c,u,v}$ denote the u,v-th pixel of the c-th color channel (where in our implementation we use three color channels, RGB) of the network's output image and the high resolution training label image, respectively. The network's output is given by $Y^{\Theta}=F(X^{LR}_{input};\Theta)$, where F is the deep neural network's operator on the low-resolution input image X^{LR}_{input} and Θ is the network's parameter space (e.g., kernels, biases, weights). Also, $(M\times L)\times(M\times L)$ is the total number of pixels in each color channel, λ is a regularization parameter, empirically set to ~0.001. $\left|\nabla Y^{\Theta}\right|^2_{c,u,v}$ is u,v-th pixel of the c-th color channel of the network's output image gradient [9], applied separately for each color channel, which is defined as: $\left|\nabla Y^{\Theta}\right|^2=\left(h*Y^{\Theta}\right)^2+\left(h^T*Y^{\Theta}\right)^2$, with:

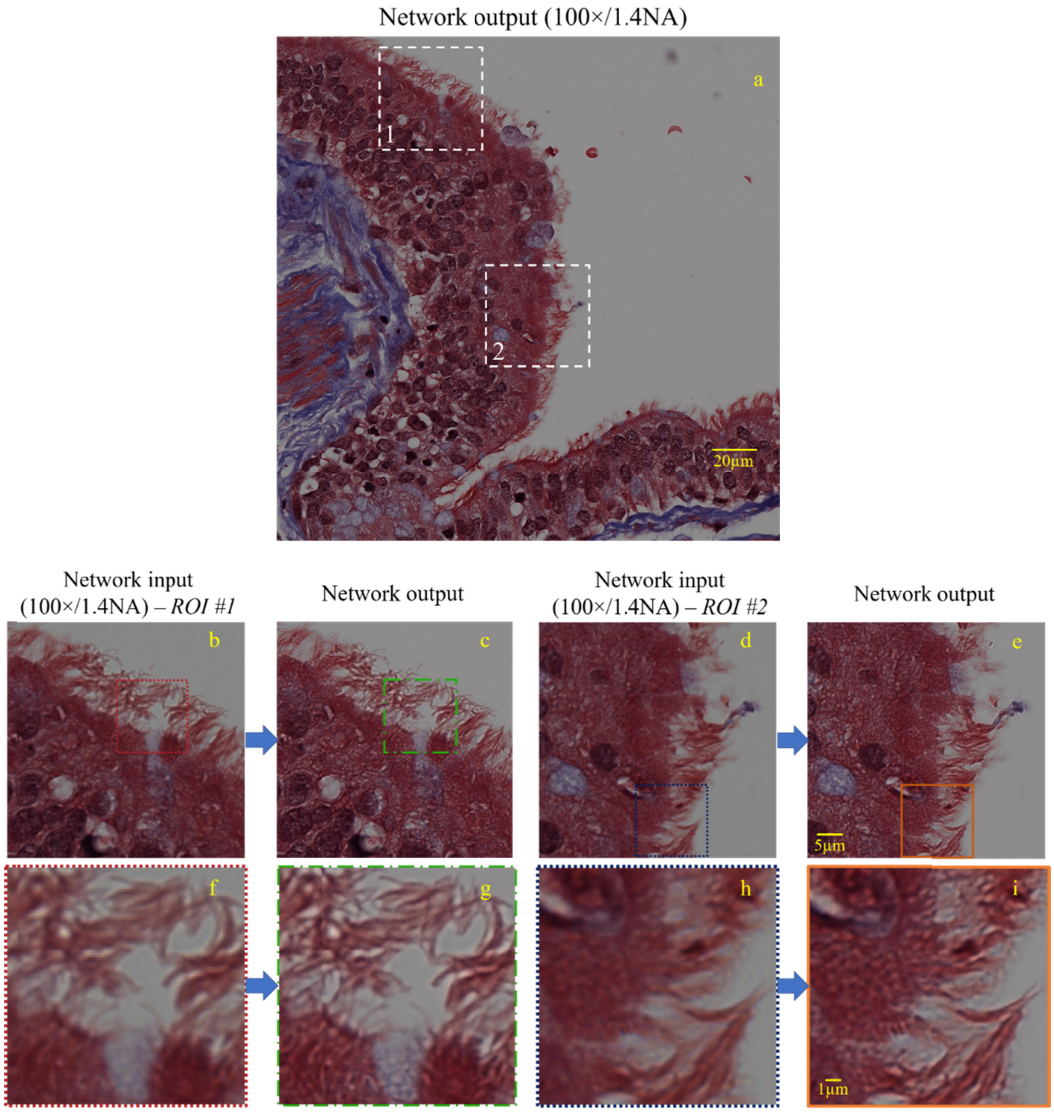

Fig. S8. Deep neural network output image corresponding to a Masson's trichrome stained lung tissue section taken from a pneumonia patient. The network was trained on images of a Masson's trichrome stained lung tissue taken from a different tissue block that was not used as part of the CNN training phase. (**a**), Image of the deep neural network output corresponding to a 100×/1.4NA input image. (**b**, **f**, **d**, **h**) Zoomed-in ROIs of the input image (100×/1.4NA). (**c**, **g**, **e**, **i**) Zoomed-in ROIs of the neural network output image.

$$h = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \qquad \textbf{(S5)}$$

and $(.)^T$ refers to the matrix transpose operator.

The above defined loss function balances between the mean-squared-error (MSE) and the image sharpness with a regularization parameter, λ. The MSE is used as a data fidelity term and the l_2-norm image gradient approximation helps mitigating the spurious edges that result from the pixel up-sampling process. Following the estimation of the loss function, the error is backpropagated through the network, and the network's parameters are learnt by using the Adaptive Moment Estimation (ADAM) optimization [10], which is a stochastic optimization method, that we empirically set a learning rate parameter of 10^{-4} and a mini-batch size of 64 image patches. All the kernels (for instance $w_{k,i,j}$) used in convolutional layers have 3×3 elements and their entries are initialized using truncated normal distribution with 0.05 standard deviation and 0 mean [1] All the bias terms (for instance, $\beta_{k,j}$) are initialized with 0.

Table S1. Average structural similarity index (SSIM) for the Masson's trichrome stained lung tissue and H&E stained breast tissue datasets, comparing bicubic up-sampling and the deep neural network output.

	Test set	Bicubic up-sampling SSIM	Deep neural network SSIM
Masson's trichrome stained lung tissue	20 images (224×224 pixels)	0.672	0.796
H&E stained breast tissue	7 images (660×660 pixels)	0.685	0.806

Table S2. Deep neural network training details for the Masson's trichrome stained lung tissue and H&E stained breast tissue datasets.

	Number of input-output patches (number of pixels for each low-resolution image)	Validation set (number of pixels for each low-resolution image)	Number of epochs till convergence	Training time
Masson's trichrome stained lung tissue	9,536 patches (60×60 pixels)	10 images (224×224 pixels)	630	4hr, 35min
H&E stained breast tissue	51,008 patches (60×60 pixels)	10 images (660×660 pixels)	460	14hr, 30min

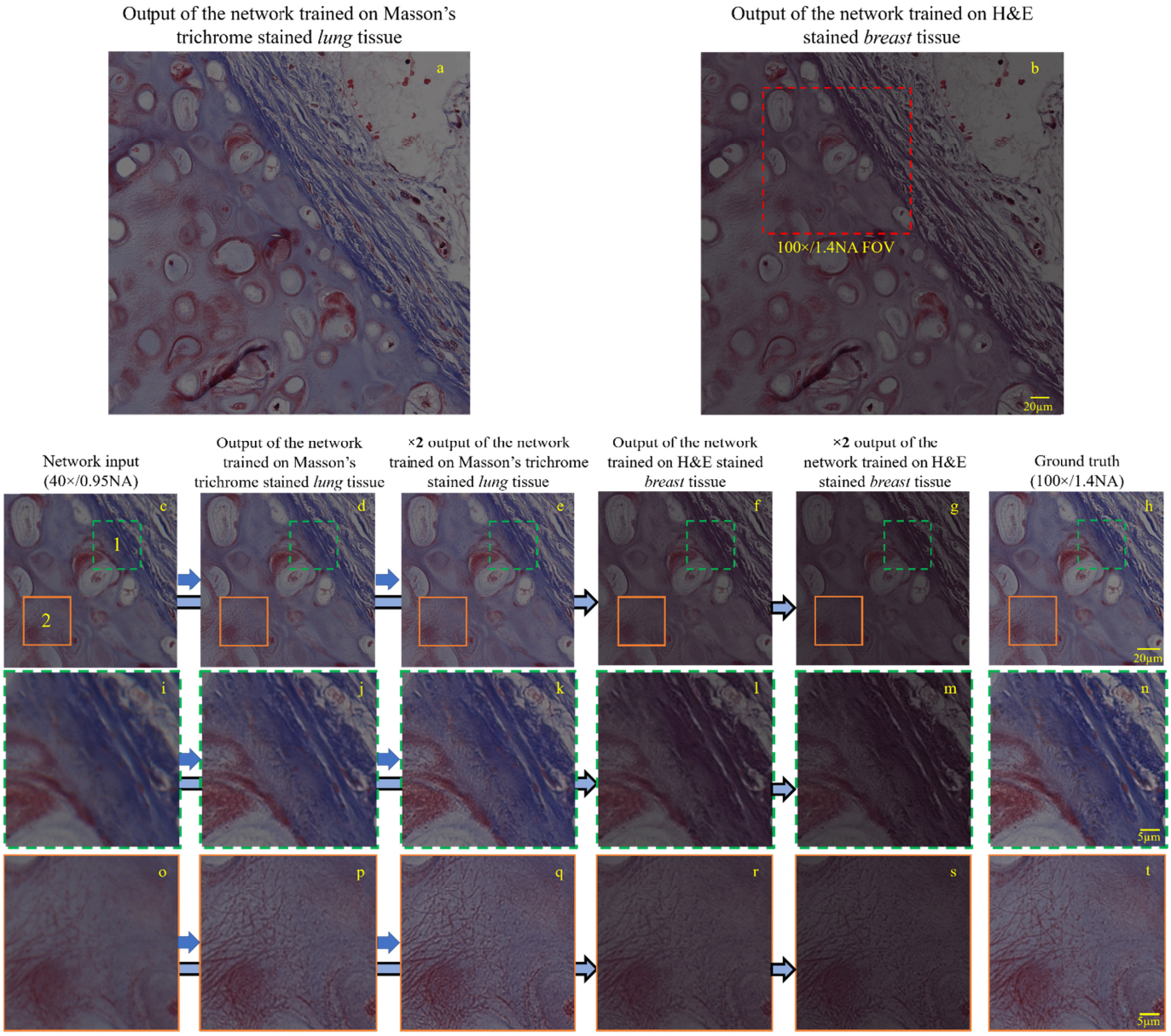

Fig. S9. (**a**), Result of applying the *lung* tissue trained deep neural network model on a 40×/0.95NA *lung* tissue input image. (**b**), Result of applying the *breast* tissue trained deep neural network model on a 40×/0.95NA *lung* tissue input image. (**c**, **i**, **o**) Zoomed in ROIs corresponding to the 40×/0.95NA input image. (**d**, **j**, **p**) Neural network output images, corresponding to input images (**c**, **i**) and (**o**), respectively; the network is trained with lung tissue images. (**e**, **k**, **q**) Neural network output images, corresponding to input images (**d**, **j**), and (**p**), respectively; the network is trained with lung tissue images. (**f**, **l**, **r**) Neural network output images, corresponding to input images (**c**, **i**) and (**o**), respectively; the network is trained with breast tissue images stained with a different dye, H&E. (**g**, **m**, **s**) Neural network output images, corresponding to input images (**f**, **l**), and (**r**), respectively; the network is trained with breast tissue images stained with H&E. (**h**, **n**, **t**) Comparison images of the same ROIs acquired using a 100×/1.4NA objective lens.

4. NETWORK TESTING

A fixed network architecture, following the training phase is shown in Fig. S5, which receives an input of $P\times Q$-pixel image and outputs a $\lceil (P\times L)\rceil\times\lceil (Q\times L)\rceil$-pixel image, where $\lceil . \rceil$ is the ceiling operator. To numerically quantify the performance of our trained network models, we independently tested it using

Table S3. Average runtime for different regions-of-interest shown in Fig. 2.

		Single GPU runtime (*sec*)		Dual GPU runtime (*sec*)	
Image FOV	Number of Pixels (input)	Network Output	Network Output x2 (Self-feeding)	Network Output	Network Output x2 (Self-feeding)
378.8 × 378.8 μm (e.g., Fig. 2A)	2048×2048	1.193	8.343	0.695	4.615
151.3 × 151.3 μm (e.g., red box in Fig. 2A)	818×818	0.209	1.281	0.135	0.730
29.6 × 29.6 μm (e.g., Figs. 2B-L)	160×160	0.038	0.081	0.037	0.062

validation images, as detailed in Table S1. The output images of the network were quantified using the structural similarity index (SSIM) [11]. SSIM, which has a scale between 0 and 1, quantifies a human observer's perceptual loss from a gold standard image by considering the relationship among the contrast, luminance, and structure components of the image. SSIM is defined as 1 for an image that is identical to the gold standard image.

Table S4. Calculated contrast values for the USAF resolution test target elements.

Period (Cycles/mm)	100×/1.4NA input contrast (a.u.)	Network output contrast (a.u.)
256	0.801144658	0.75627907
287.3502844	0.790853855	0.729511618
322.5397888	0.790801661	0.72449378
362.038672	0.795555918	0.709122843
406.3746693	0.787270294	0.726269147
456.1401437	0.771249585	0.774461828
512	0.713336904	0.681675286
574.7005687	0.636153491	0.640392221
645.0795775	0.577148622	0.588478766
724.0773439	0.517576094	0.585453198
812.7493386	0.516547441	0.63392948
912.2802874	0.439410917	0.597450901
1024	0.368925748	0.585228416
1149.401137	0.400244051	0.53887823
1290.159155	0.303987367	0.496261545
1448.154688	0.228926978	0.4729755
1625.498677	0.20194026	0.542857143
1824.560575	0.12865681	0.454976303
2048	0.110901071	0.259743799
2298.802275	0	0.254320988
2580.31831	0	0.182785747
2896.309376	0	0.072
3250.997354	0	0
3649.12115	0	0

5. IMPLEMENTATION DETAILS

The program was implemented using Python version 3.5.2, and the deep neural network was implemented using TensorFlow framework version 0.12.1 (Google). We used a laptop computer with Core i7-6700K CPU @ 4GHz (Intel) and 64GB of RAM, running a Windows 10 professional operating system (Microsoft).

The network training and testing were performed using GeForce GTX 1080 GPUs (NVidia). For the training phase, using a dual-GPU configuration resulted in ~33% speedup compared to training the network with a single GPU. The training time of the deep neural networks for the lung and breast tissue image datasets is summarized in Table S2 (for the dual-GPU configuration).

Following the conclusion of the training stage, the fixed deep neural network intakes an input stream of 100 low-resolution images each with 2,048×2,048-pixels, and outputs for each input image a 5,120×5,120-pixel high-resolution image at a total time of ~119.3 seconds (for all the 100 images) on a single laptop GPU. This runtime was calculated as the average of 5 different runs. Therefore, for L=2.5 the network takes 1.193 sec per output image on a single GPU. When employing a dual-GPU for the same task, the average runtime reduces to 0.695 sec per 2,048×2,048-pixel input image (see Table S3 for additional details on the network output runtime corresponding to other input image sizes, including self-feeding of the network output).

6. MODULATION TRANSFER FUNCTION (MTF) ANALYSIS

To quantify the effect of our deep neural network on the spatial frequencies of the output image, we have applied the CNN that was trained using the Masson's trichrome stained lung tissue samples on a resolution test target (Extreme USAF Resolution Target on 4×1 mm Quartz Circle Model 2012B, Ready Optics), which was imaged using a 100×/1.4NA objective lens, with a 0.55NA condenser. The objective lens was oil immersed as depicted in Fig. 5(a), while the interface between the resolution test target and the sample cover glass was not oil immersed, leading to an effective objective NA of ≤1 and a lateral diffraction limited resolution ≥ 0.354μm (assuming an average illumination wavelength of 550 nm). MTF was evaluated by calculating the contrast of different elements of the resolution test target [12] For each element, we horizontally averaged the resulting image along the element lines (~80-90% of the line length). We then located the center pixels of the element's minima and maxima and used their values for contrast calculation. To do that, we calculated the length of the element's cross-section from the resolution test target group and element number in micrometers, cut out a corresponding cross section length from the center of the horizontally averaged element lines. This also yielded the center pixel locations of the

element's local maximum values (2 values) and minimum values (3 values). The maximum value, $I_{\max}$, was set as the maximum of the local maximum values and the minimum value, $I_{\min}$, was set as the minimum of the local minimum values. For the elements, where the minima and maxima of the pattern matched their calculated locations in the averaged cross section, the contrast value was calculated as: $(I_{\max} - I_{\min})/(I_{\max} + I_{\min})$. For the elements where the minima and maxima were not at their expected positons, thus the modulation of the element was not preserved, we set the contrast to 0. Based on this experimental analysis, the calculated contrast values are given Table S4 and the MTFs for the input image and the output image of the deep neural network (trained on Masson's trichrome lung tissue) are compared to each other in Supplementary Fig. 5(e).

References

1. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in (2016), pp. 770–778.
2. K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15 (IEEE Computer Society, 2015), pp. 1026–1034.
3. K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, eds., Lecture Notes in Computer Science (Springer International Publishing, 2016), pp. 630–645.
4. D. Han, Kim, Jiwhan, and Kim, Junmo, "[1610.02915] Deep Pyramidal Residual Networks," https://arxiv.org/abs/1610.02915.
5. S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," IEEE Signal Process. Mag. **20**, 21–36 (2003).
6. W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network," in (2016), pp. 1874–1883.
7. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," Comput. Vis. Image Underst. **110**, 346–359 (2008).
8. "Find Image Rotation and Scale Using Automated Feature Matching - MATLAB & Simulink," https://www.mathworks.com/help/vision/examples/find-image-rotation-and-scale-using-automated-feature-matching.html.
9. A. Kingston, A. Sakellariou, T. Varslot, G. Myers, and A. Sheppard, "Reliable automatic alignment of tomographic projection data by passive auto-focus," Med. Phys. **38**, 4934–4945 (2011).
10. D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in (2014).
11. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," IEEE Trans. Image Process. **13**, 600–612 (2004).
12. J. Rosen, N. Siegel, and G. Brooker, "Theoretical and experimental demonstration of resolution beyond the Rayleigh limit by FINCH fluorescence microscopic imaging," Opt. Express **19**, 26249–26268 (2011).