Solving Fourier ptychographic imaging problems via neural network modeling and TensorFlow

SHAOWEI JIANG,[1,3] KAIKAI GUO,[1,3] JUN LIAO,[1,3] AND GUOAN ZHENG[1,2,*]

[1]*Biomedical Engineering, University of Connecticut, Storrs, CT 06269, USA*
[2]*Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269, USA*
[3]*These authors contributed equally to this work*
*guoan.zheng@uconn.edu

Abstract: Fourier ptychography is a recently developed imaging approach for large field-of-view and high-resolution microscopy. Here we model the Fourier ptychographic forward imaging process using a convolutional neural network (CNN) and recover the complex object information in a network training process. In this approach, the input of the network is the point spread function in the spatial domain or the coherent transfer function in the Fourier domain. The object is treated as 2D learnable weights of a convolutional or a multiplication layer. The output of the network is modeled as the loss function we aim to minimize. The batch size of the network corresponds to the number of captured low-resolution images in one forward/backward pass. We use a popular open-source machine learning library, TensorFlow, for setting up the network and conducting the optimization process. We analyze the performance of different learning rates, different solvers, and different batch sizes. It is shown that a large batch size with the Adam optimizer achieves the best performance in general. To accelerate the phase retrieval process, we also discuss a strategy to implement Fourier-magnitude projection using a multiplication neural network model. Since convolution and multiplication are the two most-common operations in imaging modeling, the reported approach may provide a new perspective to examine many coherent and incoherent systems. As a demonstration, we discuss the extensions of the reported networks for modeling single-pixel imaging and structured illumination microscopy (SIM). 4-frame resolution doubling is demonstrated using a neural network for SIM. The link between imaging systems and neural network modeling may enable the use of machine-learning hardware such as neural engine and tensor processing unit for accelerating the image reconstruction process. We have made our implementation code open-source for researchers.

References and links

1. G. Zheng, R. Horstmeyer, and C. Yang, "Wide-field, high-resolution Fourier ptychographic microscopy," Nat. Photonics **7**(9), 739–745 (2013).
2. X. Ou, R. Horstmeyer, C. Yang, and G. Zheng, "Quantitative phase imaging via Fourier ptychographic microscopy," Opt. Lett. **38**(22), 4845–4848 (2013).
3. L. Tian, X. Li, K. Ramchandran, and L. Waller, "Multiplexed coded illumination for Fourier Ptychography with an LED array microscope," Biomed. Opt. Express **5**(7), 2376–2389 (2014).
4. K. Guo, S. Dong, and G. Zheng, "Fourier ptychography for brightfield, phase, darkfield, reflective, multi-slice, and fluorescence imaging," IEEE J. Sel. Top. Quantum Electron. **22**(4), 1–12 (2016).
5. V. Mico, Z. Zalevsky, P. García-Martínez, and J. García, "Synthetic aperture superresolution with multiple off-axis holograms," J. Opt. Soc. Am. A **23**(12), 3162–3170 (2006).
6. J. Di, J. Zhao, H. Jiang, P. Zhang, Q. Fan, and W. Sun, "High resolution digital holographic microscopy with a wide field of view based on a synthetic aperture technique and use of linear CCD scanning," Appl. Opt. **47**(30), 5654–5659 (2008).
7. T. R. Hillman, T. Gutzler, S. A. Alexandrov, and D. D. Sampson, "High-resolution, wide-field object reconstruction with synthetic aperture Fourier holographic optical microscopy," Opt. Express **17**(10), 7873–7892 (2009).

8. L. Granero, V. Micó, Z. Zalevsky, and J. García, "Synthetic aperture superresolved microscopy in digital lensless Fourier holography by time and angular multiplexing of the object information," Appl. Opt. **49**(5), 845–857 (2010).

9. T. Gutzler, T. R. Hillman, S. A. Alexandrov, and D. D. Sampson, "Coherent aperture-synthesis, wide-field, high-resolution holographic microscopy of biological tissue," Opt. Lett. **35**(8), 1136–1138 (2010).

10. A. B. Meinel, "Aperture synthesis using independent telescopes," Appl. Opt. **9**(11), 2501 (1970).

11. T. Turpin, L. Gesell, J. Lapides, and C. Price, "Theory of the synthetic aperture microscope," Proc. SPIE **2566**, 230–240 (1995).

12. J. R. Fienup, "Phase retrieval algorithms: a comparison," Appl. Opt. **21**(15), 2758–2769 (1982).

13. V. Elser, "Phase retrieval by iterated projections," J. Opt. Soc. Am. A **20**(1), 40–55 (2003).

14. H. M. L. Faulkner and J. M. Rodenburg, "Movable aperture lensless transmission microscopy: a novel phase retrieval algorithm," Phys. Rev. Lett. **93**(2), 023903 (2004).

15. A. M. Maiden and J. M. Rodenburg, "An improved ptychographical phase retrieval algorithm for diffractive imaging," Ultramicroscopy **109**(10), 1256–1262 (2009).

16. R. Gonsalves, "Phase retrieval from modulus data," JOSA **66**(9), 961–964 (1976).

17. J. R. Fienup, "Reconstruction of a complex-valued object from the modulus of its Fourier transform using a support constraint," JOSA A **4**(1), 118–123 (1987).

18. I. Waldspurger, A. d'Aspremont, and S. Mallat, "Phase recovery, maxcut and complex semidefinite programming," Math. Program. **149**(1-2), 47–81 (2015).

19. X. Ou, G. Zheng, and C. Yang, "Embedded pupil function recovery for Fourier ptychographic microscopy," Opt. Express **22**(5), 4960–4972 (2014).

20. G. Zheng, "Breakthroughs in photonics 2013: Fourier ptychographic imaging," Photonics Journal, IEEE **6**, 1–7 (2014).

21. G. Zheng, X. Ou, R. Horstmeyer, J. Chung, and C. Yang, "Fourier ptychographic microscopy: a gigapixel superscope for biomedicine," Optics and Photonics News **4**, 26–33 (2014).

22. W. Hoppe and G. Strube, "Diffraction in inhomogeneous primary wave fields. 2. Optical experiments for phase determination of lattice interferences," Acta Crystallogr. A **25**, 502–507 (1969).

23. J. M. Rodenburg, A. C. Hurst, A. G. Cullis, B. R. Dobson, F. Pfeiffer, O. Bunk, C. David, K. Jefimovs, and I. Johnson, "Hard-x-ray lensless imaging of extended objects," Phys. Rev. Lett. **98**(3), 034801 (2007).

24. J. Rodenburg, "Ptychography and related diffractive imaging methods," Adv. Imaging Electron Phys. **150**, 87–184 (2008).

25. R. Horstmeyer and C. Yang, "A phase space model of Fourier ptychographic microscopy," Opt. Express **22**(1), 338–358 (2014).

26. T. B. Edo, D. J. Batey, A. M. Maiden, C. Rau, U. Wagner, Z. D. Pešić, T. A. Waigh, and J. M. Rodenburg, "Sampling in x-ray ptychography," Phys. Rev. A **87**(5), 053850 (2013).

27. D. J. Batey, D. Claus, and J. M. Rodenburg, "Information multiplexing in ptychography," Ultramicroscopy **138**, 13–21 (2014).

28. P. Thibault and A. Menzel, "Reconstructing state mixtures from diffraction measurements," Nature **494**(7435), 68–71 (2013).

29. A. M. Maiden, M. J. Humphry, and J. M. Rodenburg, "Ptychographic transmission microscopy in three dimensions using a multi-slice approach," J. Opt. Soc. Am. A **29**(8), 1606–1614 (2012).

30. T. M. Godden, R. Suman, M. J. Humphry, J. M. Rodenburg, and A. M. Maiden, "Ptychographic microscope for three-dimensional imaging," Opt. Express **22**(10), 12513–12523 (2014).

31. P. Thibault, M. Dierolf, O. Bunk, A. Menzel, and F. Pfeiffer, "Probe retrieval in ptychographic coherent diffractive imaging," Ultramicroscopy **109**(4), 338–343 (2009).

32. S. Ruder, "An overview of gradient descent optimization algorithms," arXiv preprint arXiv:1609.04747 (2016).

33. Y. Rivenson, Z. Göröcs, H. Günaydin, Y. Zhang, H. Wang, and A. Ozcan, "Deep learning microscopy," Optica **4**(11), 1437–1443 (2017).

34. Y. Rivenson, Y. Zhang, H. Günaydın, D. Teng, and A. Ozcan, "Phase recovery and holographic image reconstruction using deep learning in neural networks," Light Sci. Appl. **7**(2), 17141 (2018).

35. Y. Rivenson, H. Ceylan Koydemir, H. Wang, Z. Wei, Z. Ren, H. Günaydın, Y. Zhang, Z. Göröcs, K. Liang, D. Tseng, and A. Ozcan, "Deep learning enhanced mobile-phone microscopy," ACS Photonics acsphotonics.8b00146 (2018).

36. U. S. Kamilov, I. N. Papadopoulos, M. H. Shoreh, A. Goy, C. Vonesch, M. Unser, and D. Psaltis, "Learning approach to optical tomography," Optica **2**(6), 517–522 (2015).

37. S. Jiang, J. Liao, Z. Bian, K. Guo, Y. Zhang, and G. Zheng, "Transform- and multi-domain deep learning for single-frame rapid autofocusing in whole slide imaging," Biomed. Opt. Express **9**(4), 1601–1612 (2018).

38. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard, "TensorFlow: A System for Large-Scale Machine Learning," in *OSDI*, 2016), 265–283.

39. N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, and A. Borchers, "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the 44th Annual International Symposium on Computer Architecture* (ACM, 2017), 1–12.

40. L. Bian, J. Suo, G. Zheng, K. Guo, F. Chen, and Q. Dai, "Fourier ptychographic reconstruction using Wirtinger flow optimization," Opt. Express **23**(4), 4856–4866 (2015).

41. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980 (2014).

42. A. Maiden, D. Johnson, and P. Li, "Further improvements to the ptychographical iterative engine," Optica **4**(7), 736–745 (2017).
43. M. Odstrčil, A. Menzel, and M. Guizar-Sicairos, "Iterative least-squares solver for generalized maximum-likelihood ptychography," Opt. Express **26**(3), 3108–3123 (2018).
44. M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," IEEE Signal Process. Mag. **25**(2), 83–91 (2008).
45. B. Sun, M. P. Edgar, R. Bowman, L. E. Vittert, S. Welsh, A. Bowman, and M. J. Padgett, "3D computational imaging with single-pixel detectors," Science **340**(6134), 844–847 (2013).
46. Z. Zhang, X. Ma, and J. Zhong, "Single-pixel imaging by means of Fourier spectrum acquisition," Nat. Commun. **6**(1), 6225 (2015).
47. M. G. Gustafsson, "Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy," J. Microsc. **198**(Pt 2), 82–87 (2000).
48. M. G. Gustafsson, "Nonlinear structured-illumination microscopy: wide-field fluorescence imaging with theoretically unlimited resolution," Proc. Natl. Acad. Sci. U.S.A. **102**(37), 13081–13086 (2005).
49. M. G. Gustafsson, L. Shao, P. M. Carlton, C. J. Wang, I. N. Golubovskaya, W. Z. Cande, D. A. Agard, and J. W. Sedat, "Three-dimensional resolution doubling in wide-field fluorescence microscopy by structured illumination," Biophys. J. **94**(12), 4957–4970 (2008).
50. S. Dong, K. Guo, S. Jiang, and G. Zheng, "Recovering higher dimensional image data using multiplexed structured illumination," Opt. Express **23**(23), 30393–30398 (2015).
51. S. Dong, J. Liao, K. Guo, L. Bian, J. Suo, and G. Zheng, "Resolution doubling with a reduced number of image acquisitions," Biomed. Opt. Express **6**(8), 2946–2952 (2015).
52. S. Jiang, K. Guo, J. Liao, and G. Zheng, "Neural network models for Fourier ptychography," figshare (2018) [retrieved 19 June 2018], https://doi.org/10.6084/m9.figshare.6607700.

1. Introduction

Many biomedical applications require imaging with both large field-of-view and high resolution at the same time. One example is whole slide imaging (WSI) in digital pathology, which converts tissue sections into digital images that can be viewed, managed, and analyzed on computer screens. To this end, Fourier ptychography (FP) is a recently developed coherent imaging approach for achieving both large field-of-view and high resolution at the same time [1–4]. This approach integrates the concepts of aperture synthesizing [5–11] and phase retrieval [12–18] for recovering the complex object information. In a typical microscopy setting, FP sequentially illuminates the sample with angle-varied plane waves and uses a low numerical aperture (NA) objective lens for image acquisition. Changing the incident angle of the illumination beam results in a shift of the light field's Fourier spectrum at the pupil plane. Therefore, part of the light field that would normally lie outside the pupil aperture can now transmit through the system and be detected by the image sensor. To recover the complex object information, FP iteratively synthesizes the captured intensity images in the Fourier space (aperture synthesizing) and recover the phase information (phase retrieval) at the same time. The final achievable resolution of FP is determined by the synthesized passband at the Fourier space. As such, it is able to use a low-NA objective with a low-magnification factor to produce a high-resolution complex object image, combining the advantages of wide field-of-view and high resolution at the same time [19–21].

The FP approach is also closely related to the real-space ptychography, which is a lensless phase retrieval technique originally proposed for transmission electron microscopy [14, 22–24]. Real-space ptychography employs a confined beam for sample illumination and records the Fourier diffraction patterns as the sample is mechanically scanned to different positions. FP has a similar operating principle as real-space ptychography but switching the real space and the Fourier space using a lens [1, 20, 25]. The mechanical scanning of the sample in real-space ptychography is replaced by the angle scanning process in FP. Despite the difference in hardware implementation, many algorithm developments of real-space ptychography can be directly applied in FP, including the sub-sampling scheme [26], the coherent-state multiplexing scheme [27, 28], the multi-slice modeling approach [29, 30], and the object-probe recovering scheme [15, 31].

In this work, we model the Fourier ptychographic forward imaging process using a feed-forward neural network model and recover the complex object information in a network training process. A typical feed-forward neural network consists of an input and output layer,

as well as multiple hidden layers in between. For a typical convolutional neural network (CNN), the hidden layers consist of convolutional layers, pooling layers, fully connected layers and normalization layers. In the network training process, a forward pass refers to the calculation of the loss function, where the input data travels through all layers and generates output values for the loss function calculation. A backward pass refers to the process of updating learnable weights of different layers based on the calculated loss function, and the computation is made from the last layer backward to the first layer. Different gradient-descent-based algorithms can be used in the backward pass, including momentum, Nesterov accelerated gradient, Adagrad, Adadelta, RMSprop, and Adam [32]. The use of neural networks for tackling microscopy problems is a rapidly growing research field with various applications [33–37].

In our neural network models for FP, the input layer of the network is the point spread function (PSF) in the spatial domain or the coherent transfer function (CTF) in the Fourier domain. The object is treated as 2D learnable weights of a convolutional or a multiplication layer. The output of the network is modeled as the loss function we aim to minimize. The batch size of the network corresponds to the number of captured low-resolution images in one forward / backward pass. We use a popular open-source machine learning library, TensorFlow [38], for setting up the network and conducting the optimization process. We analyze the performance of different learning rates, different solvers, and different batch sizes. It is shown that a large batch size with the Adam optimizer achieves the best performance in general. To accelerate the phase retrieval process, we also discuss a strategy to implement Fourier-magnitude projection using a multiplication neural network model.

Since convolution and multiplication are the two most-common operations in imaging modeling, the reported approach may provide a new perspective to examine many coherent and incoherent systems. As a demonstration, we discuss the extension of the reported networks for modeling single-pixel imaging and structured illumination microscopy. 4-frame resolution doubling is demonstrated using a neural network for structured illumination microscopy. The link between imaging systems and neural network models may enable the use of machine-learning hardware such as neural engine (also known as AI chips) and tensor processing unit [39] for accelerating the image reconstruction process. We have made our implementation code open-source for the interested readers.

This paper is structured as follows: in Section 2, we discuss the forward imaging model for the Fourier ptychographic imaging process and propose a CNN for modeling this process. We then analyze the performance of different learning rates, different solvers, and different batch sizes of the proposed CNN. In Section 3, we discuss a strategy to implement the Fourier-magnitude projection using a multiplication neural network model. In Section 4, we discuss the extension of the reported approach for modeling single-pixel imaging and structured illumination microscopy via CNNs. Finally, we summarize the results and discuss the future directions in Section 5.

2. Modelling Fourier ptychography using a convolutional neural network

The forward imaging process of FP can be expressed as

$$I_n(x,y) = \left| (O(x,y) \cdot e^{i(k_{xn}x + k_{yn}y)}) * PSF(x,y) \right|^2, \tag{1}$$

where '$\cdot$' denotes element-wise multiplication, '$*$' denotes convolution, $O(x,y)$ denotes the complex object, $e^{i(k_{xn}x + k_{yn}y)}$ denotes the nth illumination plane wave with a wave vector (k_{xn}, k_{yn}), $PSF(x,y)$ denotes the PSF of the objective lens, and $I_n(x,y)$ denotes the nth intensity measurement by the image sensor. The Fourier transform of $PSF(x,y)$ is the CTF of the objective lens. For diffraction-limited imaging, we have

$$FT\{PSF(x,y)\} = circ(k_x^2 + k_y^2 < (NA \cdot k_0)^2) \quad,$$ where FT denotes Fourier transform, $k_0 = 2\pi / \lambda$ and λ is the wavelength, $circ$ is the circle function (it is 1 if the condition is met and 0 otherwise). Equation (1) can be rewritten as

$$I_n(x,y) = \left|O(x,y) * (PSF(x,y) \cdot e^{-i(k_{xn}x + k_{yn}y)})\right|^2 = \left|O(x,y) * PSF_n(x,y)\right|^2, \quad (2)$$

where $PSF_n(x,y) = PSF(x,y) \cdot e^{-i(k_{xn}x + k_{yn}y)}$. In most existing machine learning libraries, the learning weights need to be real numbers. As such, we need to expand the complex object and the PSF with the following equations:

$$O(x,y) = O_r(x,y) + i \cdot O_i(x,y), PSF_n(x,y) = PSF_{nr}(x,y) + i \cdot PSF_{ni}(x,y), \quad (3)$$

where subscript 'r' denotes the real part and 'i' denotes the imaginary part. Combining Eqs. (2) and (3), we can express the forward imaging model of FP as

$$I_n(x,y) = (PSF_{nr} * O_r - PSF_{ni} * O_i)^2 + (PSF_{ni} * O_r + PSF_{nr} * O_i)^2, \quad (4)$$

The goal of the Fourier ptychographic imaging process is to recover O_r and O_i based on many measurements $I_n(x,y)$ (n = 1,2,3…). Since O_r and O_i are real numbers, we can model them as a two-channel learnable filter in a CNN.

The proposed CNN model for the Fourier ptychographic imaging process is shown in Fig. 1. This model contains an input layer for the nth PSF, a convolutional layer with two channels for the real and imaginary parts of the object, an activation layer for performing the square operation, an add layer for adding the two inputs, and an output layer for the predicted FP image. For the convolutional layer, we can choose different stride value to model the down-sampling effect of the image sensor. In our implementation, we choose a stride value of 4, i.e., the pixel size of the object is 4 times smaller than that of the FP low-resolution measurements.
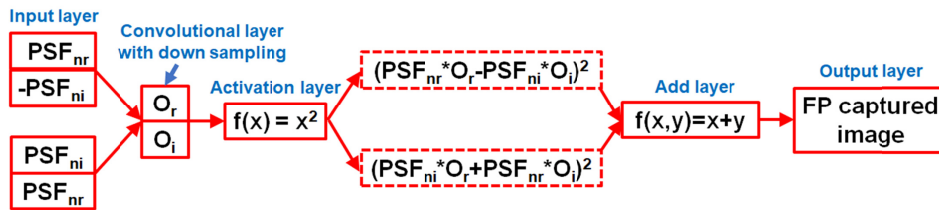

Fig. 1. The CNN model for the Fourier ptychographic imaging process. The input of the network is the nth PSF. The object is treated as a two-channel learnable filter of a convolutional layer. We use a stride of 4 for the convolutional layer, and thus, the pixel size of the output intensity images is 4 times larger than that of the object. The output of the network represents the captured FP image. The optimization process of the network is to minimize Eq. (6).

The training process of this CNN model is to recover the two-channel object (O_r, O_i) based on all FP captured images $I_n(x,y)$ (n = 1,2,3…). For an initial guess of the two-channel object (O_r, O_i), the CNN in Fig. 1 outputs a prediction $I_{n\_predict}$ in a forward pass:

$$I_{n\_predict}(x,y) = (PSF_{nr} * O_r - PSF_{ni} * O_i)^2 + (PSF_{ni} * O_r + PSF_{nr} * O_i)^2, \quad (5)$$

In the backward pass, the difference between the prediction and the captured FP image $diff(I_n, I_{n_{predict}})$ is back-propagated to the convolutional layer and the two-channel object is updated accordingly. Therefore, the training process of the CNN model can be viewed as a minimization process for the following loss function:

$$loss = diff\left(I_n, I_{n_{predict}}\right) = \sum_{n=i}^{i-1+batchSize} \left|I_n - I_{n\_predict}\right|, \tag{6}$$

where L1-norm is used to measure the difference between the prediction and the actual measurement, and 'batchSize' corresponds to the number of images in one forward / backward pass. If the batch size equals to 1, it is stochastic gradient descent with the gradient evaluated by a single image at one forward / backward pass. If the batch size equals to the total number of measurements, it is similar to using Wirtinger derivatives and gradient descent scheme to recover the complex object [40], except that we use L1-norm in Eq. (6) (the difference between L1 / L2 norms will be discussed in a later section).
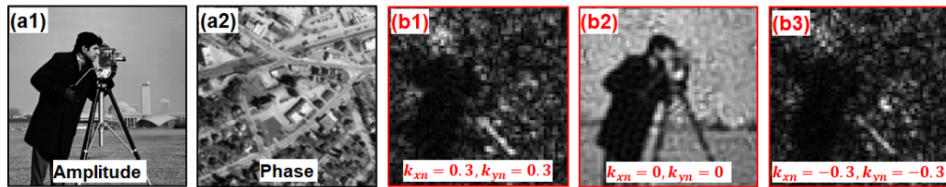

Fig. 2. (a) High-resolution amplitude and phase images for simulation. (b) The output of the CNN based on (a) and different wave vector (k_{xn}, k_{yn})s.

We first analyze the performance using simulation. Figure 2(a) shows the high-resolution object amplitude and phase. Figure 1(b) shows the CNN output for the low-resolution intensity images with different wave vector (k_{xn}, k_{yn})s. In this simulation, we use 15 by 15 plane waves for illumination and 0.1 NA objective lens to acquire images. The step size for k_{xn} and k_{yn} is 0.05, and the maximal synthetic NA is ~0.64. The pixel size in this simulation is 0.43 μm for the high-resolution object and 1.725 μm for the low-resolution measurements at the object plane (assuming magnification factor is 1). The use of these parameters is to simulate a microscope platform with 2X, 0.1 NA objective and an image sensor with 3.45 μm pixel size.

In Fig. 3, we show the recovered results with different learning rates. The Adam optimizer is used in this simulation. This optimizer is a first-order gradient-based optimizer using adaptive estimates of lower-order moments [41]. It combines the advantages of two extensions of stochastic gradient descent Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp) [41], and it is the default optimizer for many deep learning problems.
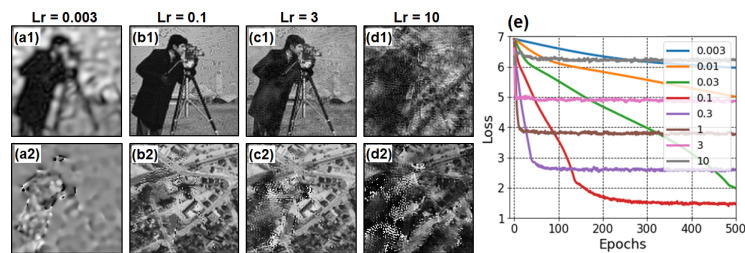

Fig. 3. Different learning rates of the Adam optimizer in TensorFlow for the training process. (a)-(d) The recovered complex object images with learning rates ranging from 0.003 to 10. (e) The L1 loss (in log scale) as a function of epochs. A higher learning rate can decay the loss faster but gets stuck at a worse value of loss. This is because there is too much 'energy' in the optimization process and the learnable weights are bouncing around chaotically, unable to settle in a nice spot in the optimization landscape.

Different learning rates in Fig. 3 represent different step sizes of the gradient descent approach. We can see that a higher learning rate can decay the loss faster but gets stuck at a

worse value of loss. This is because there is too much 'energy' in the optimization process and the learnable weights are bouncing around chaotically, unable to settle in a nice spot in the optimization landscape. On the other hand, a lower learning rate is able to reach a lower minimum point in a slower process. A straight-forward approach for a better learning-rate schedule is to use a large learning rate at the beginning and reduce it for every epoch. How to schedule the learning rate for FP is an interesting topic and requires further investigation in the future.
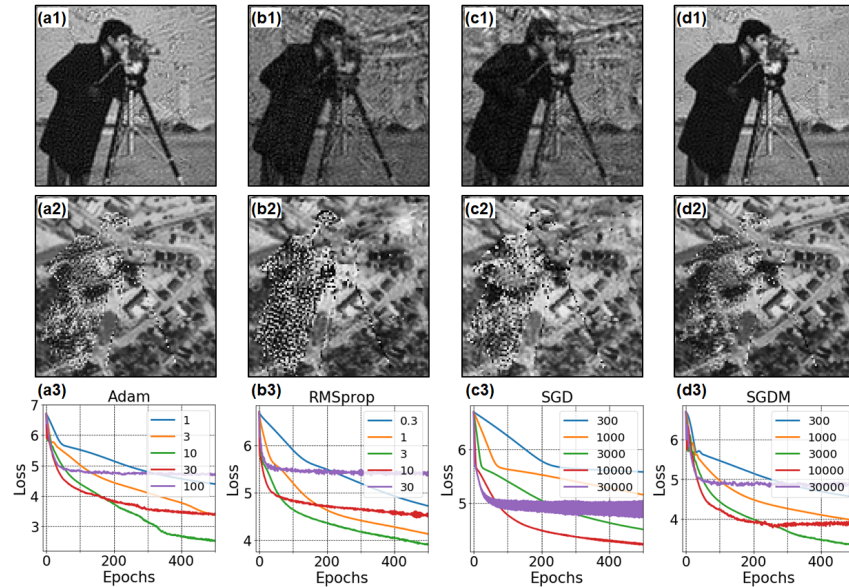

Fig. 4. Performance of different solvers in TensorFlow: (a) Adam, (b) RMSprop, (c) SGD, and (d) SGDM. We use 5 by 5 plane waves for sample illumination and the step size for k_{xn} and k_{yn} is 0.15 in this simulation. The recovered amplitude ((a1)-(d1)) and phase ((a2)-(d2)) with 500 epochs (the best learning rates are chosen in this simulation). Different color curves in (a3)-(d3) represent different learning rates and the loss is in log scale. Adam gives the best performance overall. Batch size is 1 in this simulation.
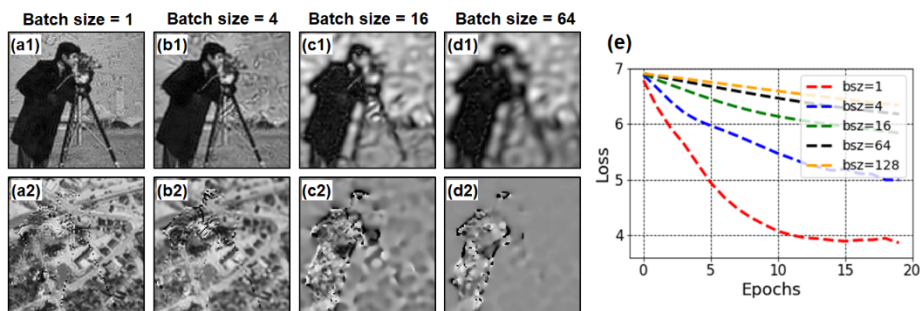

Fig. 5. Performance of different batch sizes as a function of epoch. (a)-(d) The recovered object for different batch sizes and with 20 epochs. (e) The loss (in log scale) with different batch sizes.

In Fig. 4, we compare the performance of different optimizers in TensorFlow and show their corresponding recovered results. We note that all optimizers give similar results if the step size for k_{xn} and k_{yn} is small (i.e., aperture overlap is large in the Fourier domain). In this simulation, we use 5 by 5 plane wave illumination with 0.15 step size for k_{xn} and k_{yn}. Other

parameters are the same as before. Figure 4 shows that Adam achieves the best performance and stochastic gradient descent (SGD) is the worst among the 4. Stochastic gradient descent with momentum (SGDM) is the second-best option. This justifies the adding of momentum in the recent ptychographical iterative engine [42].

In Fig. 5, we investigate the effect of different batch sizes for the optimization process. We can see that batch size of 1 gives the best performance in Fig. 5(a). This justifies the stochastic gradient descent scheme used in the extend ptychography iterative engine (ePIE) [15].
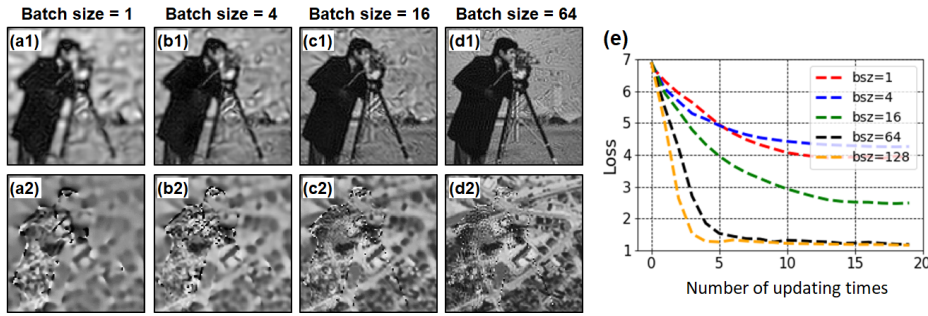

Fig. 6. Performance of different batch sizes as a function of the updating times. (a)-(d) The recovered object for different batch sizes and with 20 updating times. (e) The loss curves (in log scale) with different batch sizes.

However, one advantage of using TensorFlow library is to perform parallel processing via graphics processing unit (GPU) or tensor processing unit (TPU). As a reference point, a modern GPU can handle hundreds of images in one patch. When we use a large batch size, the processing time is about the same as that of batch size = 1. For example, the batch size is 1 in Fig. 5(a) and the epoch number is 20; therefore, we update the object with 225*20 times in this simulation. On the other hand, the batch size is 64 in Fig. 5(d) and the epoch number is 20; therefore, we update the object with (225/64)*20 times for this figure. We define 'number of updating times' as the number of epochs divided by the batch size. This 'number of updating times' is directly related to the processing time of the recovery process.

In Fig. 6(a)-6(d), we show the recovered results with the same number of updating times. In this case, we can see a large batch size leads to a better performance. Based on Figs. 5 and 6, we can draw the following conclusion: batch size of 1 is preferred for serial operation via CPU and a large batch size is preferred for parallel operation using GPU or TPU.

3. Modelling Fourier-magnitude projection in neural network

All widely used iterative phase retrieval algorithms have at their core an operation termed Fourier-magnitude projection (FMP), where an exit complex wave estimate is updated by replacing its Fourier magnitude with measured data while keeping the phase untouched. In this section, we discuss the implementation of FMP in neural network modeling. The motivation is to implement many existing phase retrieval algorithms via neural network training processes. We demonstrate it with FP and it can be easily extended for other phase retrieval problems.

In the Fourier ptychographic imaging process, the exit complex wave $\widehat{\varphi}_n\left(k_x, k_y\right)$ in the Fourier domain can be expressed as

$$\widehat{\varphi}_n\left(k_x, k_y\right) = \widehat{O}\left(k_x, k_y\right) \cdot \mathrm{CTF}_n\left(k_x, k_y\right), \tag{7}$$

where $\hat{O}(k_x, k_y)$ and $\mathrm{CTF}_n(k_x, k_y)$ are the Fourier spectrum of the object $O(x,y)$ and the nth PSF $PSF_n(x,y)$. The FMP operation for the exit complex wave $\widehat{\varphi_n}(k_x, k_y)$ can be written as

$$\varphi_n^{\widehat{update}}(k_x, k_y) = FT\{\sqrt{I_n} \cdot \frac{O(x,y)*PSF_n(x,y)}{|O(x,y)*PSF_n(x,y)|}\} \tag{8}$$

In many ptychographic phase retrieval schemes [15, 19, 31, 42, 43], it is common to perform the recovery process by dividing the optimization problem into two sub tasks: 1) perform an FMP to update the exit wave, and 2) the difference between the updated exit wave and the original exit wave is back-propagated to update the object and/or the illumination probe. For the Fourier ptychographic imaging process, we can minimize the following loss function after the FMP process in Eq. (8):

$$loss = diff\left(\varphi_n^{\widehat{update}}(k_x, k_y), \hat{O}(k_x, k_y) \cdot \mathrm{CTF}_n(k_x, k_y)\right)$$
$$= \sum_{n=i}^{i-1+batchSize} \left|\varphi_n^{\widehat{update}} - \hat{O} \cdot \mathrm{CTF}_n\right|^2, \tag{9}$$

where L2-norm is used to measure the difference between the updated exit wave and the original exit wave. If the batch size equals to 1, Eq. (9) is similar the ePIE scheme [15].
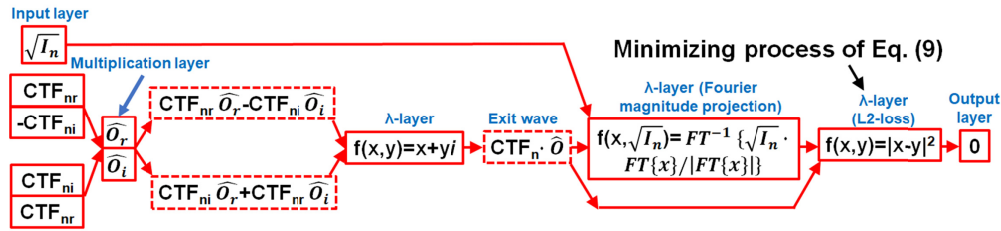

Fig. 7. The multiplication neural network model for the Fourier ptychographic imaging process in Eq. (9). The input of the network is the nth CTF and the measured data $\sqrt{I_n}$. The object's Fourier spectrum is modeled as learnable weights of a multiplication layer. We use a λ-layer to define the Fourier-magnitude-projection operation. The output of the network is 0, and thus, the training process of the network minimizes the loss function defined in Eq. (9).

The neural network for minimizing Eq. (9) is shown in Fig. 7, where we model the object's Fourier spectrum as learnable weights of a multiplication layer. Since the learnable weights need to be real in TensorFlow implementation, we separate the complex object spectrum $\hat{O}(k_x, k_y)$ into two channels with subscripts 'r' and 'i' representing the real and imaginary parts in Fig. 7. The input layer for this network is the nth CTF_n and the captured low-resolution amplitude $\sqrt{I_n}$. We use a λ-layer in TensorFlow to define the Fourier-magnitude-projection operation. The output of the network is 0, and thus, the training process of the network minimizes the loss function defined in Eq. (9). Once the complex object spectrum $\hat{O}(k_x, k_y)$ is recovered in the network training process, we can perform an inverse Fourier transform to get the complex object $O(x,y)$ in the spatial domain.

In Fig. 8, we compare the recovered results in 4 cases: 1) minimizing the loss function in Eq. (6) with L2-norm ('L2 intensity' in Fig. 8(a), 2) minimizing the loss function in Eq. (6) with L1-norm ('L1 intensity' in Fig. 8(b), 3) minimizing the loss function in Eq. (9) with L2-norm ('L2 exit wave' in Fig. 8(c), and 4) minimizing the loss function in Eq. (9) with L1-norm ('L1 exit wave' in Fig. 8(d). We quantify the performance of different approaches using

mean square error (MSE) in Fig. 8(e). We can see that the cases of 'L1 intensity' and 'L2 exit wave' give the best results. In particular, 'L2 exit wave' converges faster at the first few epochs while 'L1 intensity' reaches a lower MSE with more iterations. We also note that the intensity updating cases tend to recover the low-resolution features first while the exit-wave updating cases tend to recover features at all levels. This behavior can be explained by the loss functions in Eqs. (6) and (9). The loss function in Eq. (6) is to reduce the difference between two intensity images in the spatial domain. Therefore, it tends to correct the low-frequency difference first since most energy concentrates in this region. On the other hand, the loss function in Eq. (9) is to reduce the difference between two Fourier spectrums and it does not focus on the low-frequency regions. As such, the resolution improvement is more obvious for the exit-wave updating cases shown in Fig. 8(c).
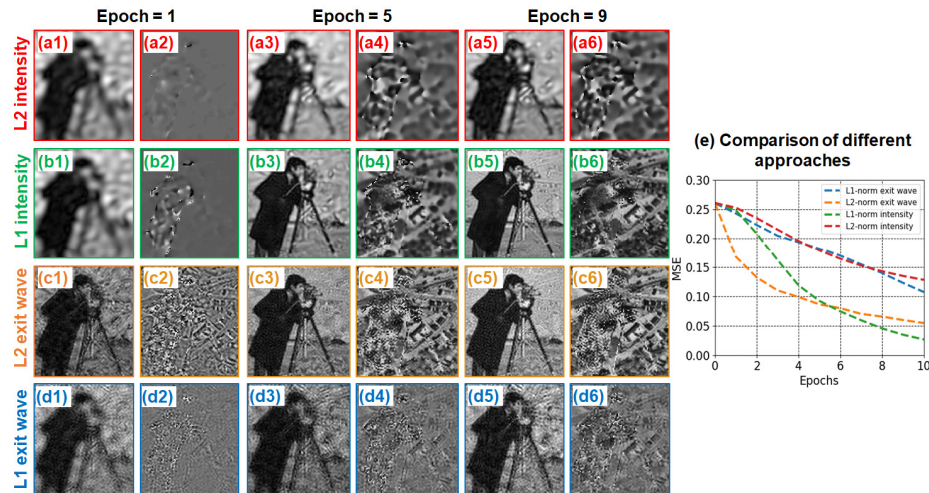

Fig. 8. Comparison of different cases with batch size = 1. The learning rates are chosen based on the fastest loss decay in 10 epochs. The Adam optimizer is used for all cases in this simulation study. (a) Minimizing the loss function in Eq. (6) with L2-norm, termed 'L2 intensity'. (b) Minimizing the loss function in Eq. (6) with L1-norm, termed 'L1 intensity'. (c) Minimizing the loss function in Eq. (9) with L2-norm, termed 'L2 exit wave'. (d) Minimizing the loss function in Eq. (9) with L1-norm, termed 'L1 exit wave'. The resolution improvement is more obvious for the exit-wave cases. (e) The performances of different approaches are quantified using MSE.

In Fig. 9, we test the L2-norm exit-wave network using experimental data. Figure 9(a) shows the experimental setup where we use a 2X, 0.1 NA Nikon objective with a 200 mm tube lens (Thorlabs TTL200) to build a microscope platform. A 5-megapixel camera (BFS-U3-51S5M-C) with a 3.45 μm pixel size is used to acquire the intensity images. We use an LED array (Adafruit 32 by 32 RGB LED matrix) to illuminate the sample from different incident angles and the distance between the LED array and the sample is ~85 mm. In this experiment, we illuminate the sample from 15 by 15 different incident angles and the corresponding maximum synthetic NA is ~0.55. Figure 9(b1)-9(d1) show the low-resolution images captured by the microscope platform in Fig. 9(a). We use the L2-norm exit-wave network with the Adam optimizer to recover the complex object spectrum in the multiplication layer. The batch size in this experiment is 1 and we use 20 epochs in the network training (optimization) process. The recovered object intensity images are shown in Fig. 9(b2)-9(d2) and the recovered phase images are shown in Fig. 9(b3)-9(d3). As a comparison, we also show the standard FPM reconstructions [19] in Fig. 9(b4)-(d4) and 9(b5)-(d5). Figure 9 validates the effectiveness of reported neural network models.
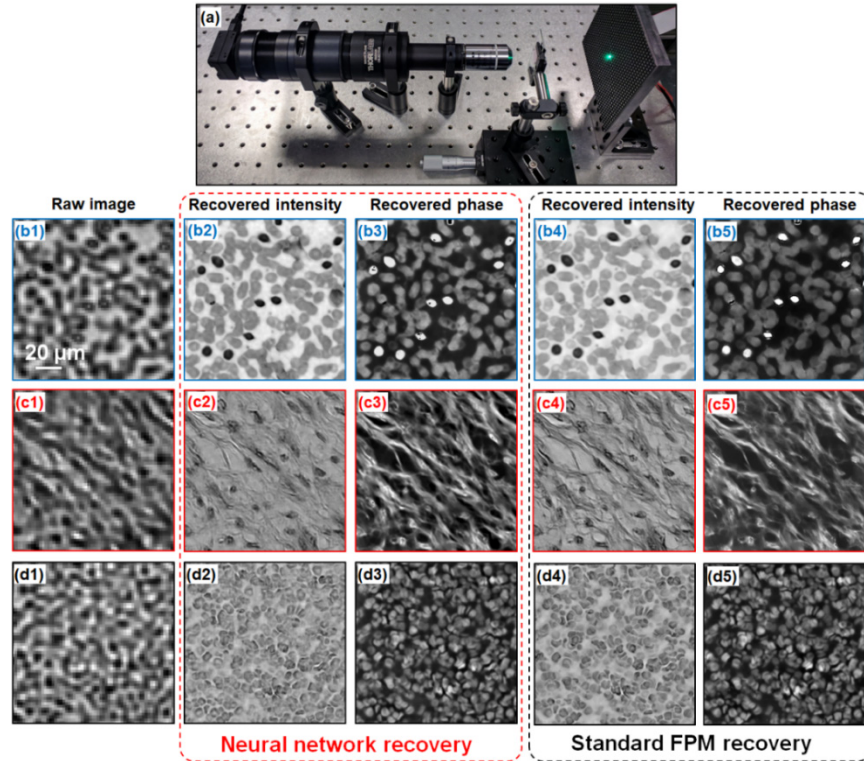
Fig. 9. Test the L2-norm exit-wave network with experimental data. We use Adam optimizer with 20 epochs in this experiment and the batch size is 1. (a) The experimental setup with a 2X, 0.1 NA objective lens and a 3.45 µm pixel size camera, which is the same as the simulation setting. We test three different samples: (b) a blood smear, (c) a brain slide, and (d) a tissue section stained by immunohistochemistry methodology. (b1)-(d1) show the captured raw images using the 2X objective lens. The recovered intensity images using neural network are shown in (b2)-(d2) and the recovered phase images are shown in (b3)-(d3). As a comparison, (b4)-(d4) and (b5)-(d5) show the standard FPM reconstructions for intensity and phase [19].

4. Extensions for single-pixel imaging and structured illumination microscopy

In this section, we extend the network models discussed above for single-pixel imaging and structured illumination microscopy. Single pixel imaging captures images using single-pixel detectors. It enables imaging in a variety of situations that are impossible or challenging with conventional 2D image sensors [44–46]. The forward imaging process of single-pixel imaging can be expressed as

$$I_n = \sum_{x,y} O(x,y) \cdot P_n(x,y), \tag{10}$$

where $O(x,y)$ denotes the 2D object, $P_n(x,y)$ denotes the nth 2D illumination pattern on the object, and I_n denotes the nth single-pixel measurement. The summation sign in Eq. (10) represents the signal summation over the x-y plane. Since the dimensions of the object and pattern are the same, the forward imaging model in Eq. (10) can be modeled by a 'valid convolutional layer' which outputs a predicted single-pixel measurement. The CNN model for single-pixel imaging is shown in Fig. 10(a). The training of this model is to minizine the following loss function:

$$loss = diff\left(I_n, \sum_{x,y} O(x,y) \cdot P_n(x,y)\right) = \sum_{n=i}^{i-1+batchSize} \left| I_n - \sum_{x,y} O(x,y) \cdot P_n(x,y) \right|, \quad (11)$$

where L1-norm is used to measure the difference between the network prediction and the actual measurements. A detailed analysis of the different solvers' performance and different regularization schemes is beyond the scope this paper.
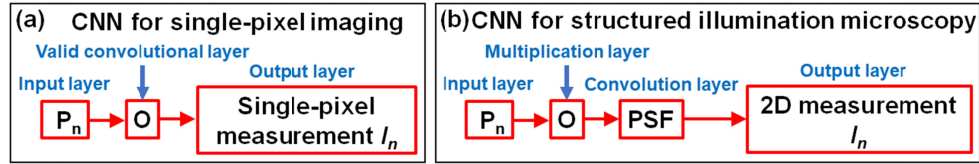

Fig. 10. CNN models for (a) single-pixel imaging and (b) structured illumination microscopy. For single-pixel imaging, the input is the illumination pattern $P_n(x,y)$, the object is modeled as learnable weights in a valid convolutional layer, and the output is the predicted single-pixel measurement. Similarly, for structured illumination microscopy, the object is modeled as learnable weights for a multiplication layer and the output is the predicted 2D image.

Structured illumination microscopy (SIM) uses non-uniform patterns for sample illumination and combine multiple acquisitions for super-resolution image recovery [47–50]. Frequency mixing between the sample and the non-uniform illumination pattern modulates the high-frequency components to the passband of the collection optics. Therefore, the recorded images contain sample information that is beyond the limit of the employed optics. Conventional SIM employs sinusoidal patterns for sample illumination. In a typical implementation, three different lateral phase shifts (0, $2\pi/3$, $4\pi/3$) are needed for each orientation of the sinusoidal pattern, and 3 different orientations are needed to double the bandwidth isotopically in the Fourier domain. Therefore, 9 acquisitions are needed.

The forward imaging process of SIM can be expressed as

$$I_n(x,y) = \left(O(x,y) \cdot P_n(x,y)\right) * PSF, \quad (12)$$

where $O(x,y)$ denotes the 2D object, $P_n(x,y)$ denotes the nth 2D illumination pattern on the object, PSF denotes the PSF of the objective lens, and $I_n(x,y)$ denotes the nth 2D image measurement. The CNN model for SIM is shown in Fig. 10(b), where the object is modeled as learnable weights of a multiplication layer. The training of this model is to minizine the following loss function:

$$loss = diff\left(I_n, \left(O(x,y) \cdot P_n(x,y)\right) * PSF\right)$$
$$= \sum_{n=i}^{i-1+batchSize} \left| I_n - \left(O(x,y) \cdot P_n(x,y)\right) * PSF \right|, \quad (13)$$

where L1-norm is used to measure the difference between the network prediction and the actual measurements. Figure 11 shows our simulation of using the proposed CNN model for SIM reconstruction. Figure 11(a) shows the object image under uniform illumination; the resolution of this image represents the resolution of the employed objective lens. Different from the conventional implementation using 9 illumination patterns, we use 4 sinusoidal patterns for sample illumination and the 4 SIM measurements are shown in Fig. 11(b) [51]. The recovered super-resolution image using the proposed CNN is shown in Fig. 11(c). The minimization process via Eq. (13) is able to reduce the number of acquisitions for SIM.
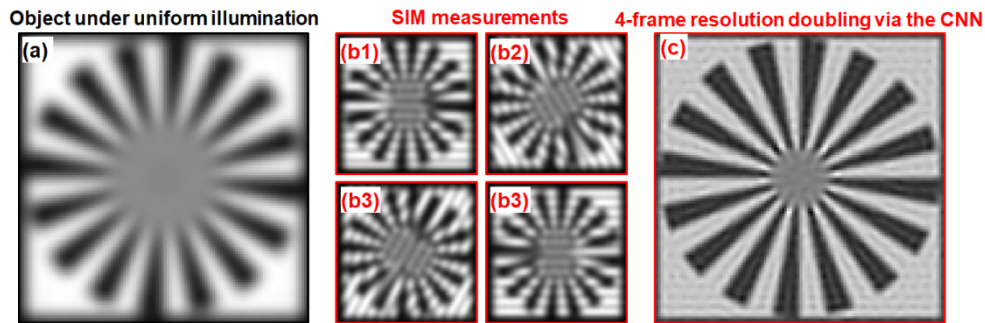
Fig. 11. 4-frame resolution doubling using the CNN-based SIM model. (a) The object image under uniform illumination. (b) The 4 SIM measurements using 4 sinusoidal patterns for sample illumination. The resolution doubled image recovered by the training process of the CNN in Fig. 10(b).

5. Summary and discussion

In summary, we model the Fourier ptychographic forward imaging process using a convolutional neural network and recover the complex object information in the network training process. In our approach, the object is treated as 2D learnable weights of a convolutional or a multiplication layer. The output of the network is modeled as the loss function we aim to minimize. The batch size of the network corresponds to the number of captured low-resolution images in one forward / backward pass. We use the popular open-source machine learning library, TensorFlow, for setting up the network and conducting the optimization process. We show that the Adam optimizer achieves the best performance in general and a large batch size is preferred for GPU / TPU-based parallel processing.

Another contribution of our work is to model the Fourier-magnitude projection via a neural network model. The Fourier-magnitude projection is the most important operation in iterative phase retrieval algorithms. Based on our model, we can easily perform exit-wave-based optimization using TensorFlow. We show that L2-norm is preferred for exit-wave-based optimization while L1-norm is preferred for intensity-based optimization.

Since convolution and multiplication are the two most-common operations in imaging modeling, the reported approach may provide a new perspective to examine many coherent and incoherent systems. As a demonstration, we discuss the extensions of the reported networks for modeling single-pixel imaging and structured illumination microscopy. We show that single-pixel imaging can be modeled by a convolutional layer implementing 'valid convolution'. For structured illumination microscopy, we propose a network model with one multiplication layer and one convolutional layer. In particular, we demonstrate 4-frame resolution doubling via the proposed CNN. Since the proposed network model can be implemented in neural engine and TPU, we envision many opportunities for accelerating the image reconstruction process via machine-learning hardware.

There are many future directions for this work. First, we can implement the CTF updating scheme in the proposed neural network models. One solution is to make the incident wave vector as the input and we can then convolute the CTF with $\delta(k_{xn}, k_{yn})$ to generate CTF_n. In this case, we can model CTF as learnable weights in a convolutional layer and it can be updated in the network training process. Second, correcting positional error is an important topic for real-space ptychographic imaging. The positional errors in real-space ptychography is equivalent to the errors of incident wave vectors in FP. We can, for example, model (k_{xn}, k_{yn}) as learnable weights in a layer and they can be updated in the network training process. Similarly, we can also generate CTF based on coefficients of different Zernike modes and model such coefficients as learnable weights. Third, the proposed network models are developed for one coherent state. It is straight forward to extend our networks to model

multi-state cases. Fourth, we use a fixed learning rate in our models. How to schedule the learning rates for faster convergence is an interesting topic and requires further investigations. Fifth, we can add regularization term such as total variation loss in the model to better handle the measurement noises.

We provide our implantation code in in the format of Jupyter notebook [Code 1, 52].

Funding

Disclosures

G. Zheng has the conflict of interest with Clearbridge Biophotonics and Instant Imaging Tech, which did not support this work.