

Solving Inverse Problems With Multi-scale Deep Convolutional Neural Networks

Feng Wang^{1,2} and Johannes Müller¹ and Alberto Eljarrat¹ and Trond R. Henninen²,
and Erni Rolf² and Christoph T. Koch¹

¹ *Institut für Physik, IRIS Adlershof der Humboldt-Universität zu Berlin, 12489 Berlin, Germany*

² *Electron Microscopy Center, Empa, Swiss Federal Laboratories for Materials Science and Technology, CH-8600 Dübendorf, Switzerland*

October 30, 2018

We propose multi-scale deep convolutional neural networks (MDCNN) as a general purpose solution for image(s)-to-image(s) inverse problems. Deep convolutional neural networks (DCNN)[1] have shown great potential for general and highly variable tasks across many image-based domains, especially successful for pixel-to-pixel applications in which the input images and the output images are similar[2, 3]. Here, we present this generalized framework to solve inverse problems with a traditional DCNN architecture enabling residual connections between the encapsulated layers of a pair of encoder and decoder, trained in an end-to-ends fashion from direct or indirect measurements to multi-scale reconstructions, no longer constrained by the choice of the mathematical formulations and the implementation of the reconstruction algorithms. The expediency of this framework is demonstrated by benchmarking predicted retrieval with a classic transport of intensity equation (TIE)[4] scheme, showing a performance on par with existing solvers. By predicting phases from a single astigmatism image, which is not feasible by other methods for its inner complexity, MDCNN shows good potential for application independent generalization, demonstrates the possibility to solve inversion problems unconstrained by complex mathematical formulas and complicated reconstruction algorithms. Furthermore, utilizing the capability of multi-channel input inherited from DCNN, MDCNN is able to address the stalling and non-uniqueness problems arising from single measurements, showing good robustness against parameters even when the mathematical nature is not evident. This opens a way for a large variety of new applications in a wide range of inverse problems, especially in the places where phase imaging or structure retrieval is an issue. The core idea of our architecture is, learning to solve inverse problems through big data, rather than coding explicit reconstruction algorithms from the mathematical formulas that characterized inverse problems. Previous works has targeted much more on *how* can we reconstruct rather than *what* we can reconstruct. Our strategy offers a shift of this paradigm.

Introduction

Physics theories allow us to make predictions: given a complete description of the state of a physical system, we can make predictions of some measurements. Predicting the result of measurements is a simulation problem. On the other hand, the inverse problem consists of using the actual result of some measurements to deduce the parameters characterizing the system which are not obtainable using direct measurements: either because they are lost during an experiment, for example the uncertainty relation of the quantum mechanical wave equations, or perhaps because they are never measured accurately in the first place, for example the defection in the loss of

information process. The goal of this paper is to develop a generic framework to solve the image(s)-to-image(s) inverse problems generally, without being limited to specific applications.

Powered by recent advances of hardwares and very large datasets[5, 6], deep learning algorithms[1] have recently shown great potential in image(s)-to-image(s) translation tasks, such as super resolution[7], image denoising[3], image generation[8] and in the field of physics[9, 10].

In this paper we delineate the framework of a general purposed MDCNN, benchmark its performance with a traditional transmission of intensity equation (TIE) application, show its potential for the new applications in a wide range of inverse problems by predicting phases from a complex mathematical model, demonstrate how to address ill-formed problems with multiple observations, and how to deal with special problems with hidden parameters in which a direct simulation is not technically feasible.

An encoder-decoder[11] is able to reduce the input low-dimensional data into high-dimensional data (encoding) by training a DCNN with a relatively smaller central layer to generate desired output data (decoding). By adding skipped connections between corresponding layers of the same dimension to an encoder-decoder, U-Net[12] enables shared information between the input and the output directly transported across the whole network.

A commonly used U-Net architecture is demonstrated in the upper-left part of Fig.1(A). because of the high non-linearity of such deep a U-Net architecture, it is difficult to use a gradient descent optimization algorithm for fine-tuning the weights in very deep layers. For example, when training a U-Net to predict phases from an intensity gradient, the slow convergence is a big problem: as is presented in Fig.1(B), within a large time frames, the U-Net converges very slowly and can even get stuck at a poor local minimum.

DCNNs usually designed with millions of learnable parameters and most of them take days or weeks to train on one or more fast and power-hungry GPUs. Many tricks exist to tune the training process, but there is no free lunch for the speed-up and convergence. Awaiting that DCNN first quickly captures the dominant low-frequency components, and then relative slowly captures high-frequency ones[13, 14], inspired by the idea of multigrid methods[15], we extend the functionality of the hidden layers in the decoder by connecting them to additional convolution layers to produce coarse outputs, in attempt to match the low-frequency components. A topological set-up of those coarse output branches are visualized in the lower part of Fig.1(A), in which the outputs of dimension 256×256 , 128×128 , 64×64 , 32×32 , 16×16 , 8×8 and 4×4 represent the low frequency components. When the training starts, all these low frequency components at the output share the same weight as the 512×512 branch, then get deserted at a latter stage. The convergence curves of an MDCNN and a U-Net with identical set-ups, are presented in the chart in Fig.1(B) for comparison: MDCNN quickly reaches a mean-absolute-error (MAE) loss around 0.07 while the conventional U-Net is very difficult to train, being trapped around 0.2.

Cross validation with conventional algorithms

The phase problem[17], one of the fundamental inversion problem in optics, neutron, atomic and electron optics, has attracted a lot of efforts and led to the invention of numerous methods to reconstruct the missing phases of the propagated wave from direct intensity measurements. After Dennis Gabor's proposal of the holographic schemes[18], putting more efforts extracting phases by post-processing images rather than pushing the imaging resolution limits, many techniques[19, 20, 16, 21, 22] emerged to retrieve phase information from a series of observations with deliberate approximations of the mathematical formulas, elaborate configurations of the experiments and increasingly complicated implementations of the reconstruction algorithms.

The TIE scheme is one of the most popular phase retrieval approaches, solving phase problem in a two-step fashion. For the first step, this method estimates intensity gradient from a series of images recorded at different axial position z_i , commonly taking a linear combination form of

$$\frac{\partial I(\vec{r}, z)}{\partial z} \Big|_{z=0} \simeq \sum_i a_i I(\vec{r}, z_i) \quad i \geq 2, \quad (1)$$

in which $I(\vec{r}, z)$ is the intensity at axial position z with 2D position vector $\vec{r} = (x, y)$, and a_i are the coefficients determined by the algorithm of the choice. Then this method tries to recover phase $\phi(\vec{r}, z)$ from a second order, elliptic partial differential equation

$$\frac{\partial I(\vec{r}, z)}{\partial z} = -\frac{2\pi}{\lambda} \nabla \cdot (I(\vec{r}, z) \nabla \phi(\vec{r}, z)), \quad (2)$$

in which λ is the wave length and $I(\vec{r}, z)$ is constant for pure phase object case.

With this scheme, two new problems cropped out. The first one is the estimation of $\frac{\partial I(\vec{r}, z)}{\partial z} \Big|_{z=0}$: finite difference method indicates the intensities taken closer to the focal plane $z = 0$ have much larger coefficients than the ones further away, while the noise reduction method requires all the coefficients to be distributed as equitably as possible[23]. A compromise has to be made balancing the high order non-linearity and the noise levels, which

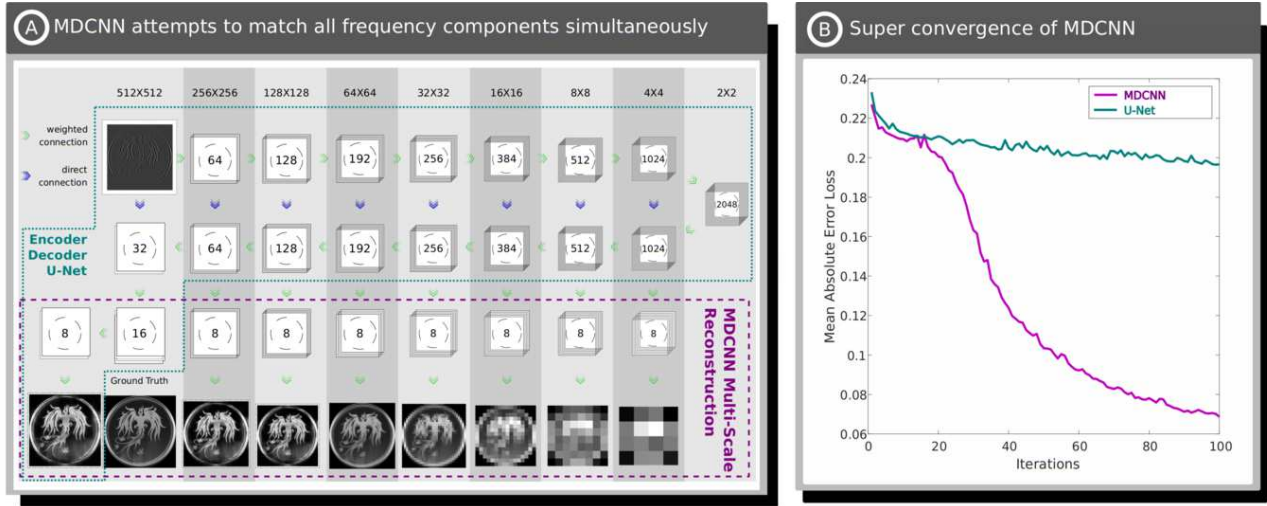


Figure 1: MDCNN achieves super convergence by matching components at all frequencies. (A) An MDCNN designed to predict phases from an astigmatism image. This deep neural network is composed of a classic U-Net (the upper-and-left part) and a multi-scale reconstruction networks (the lower part). The data flow is from left to right, from top to bottom, and from encoder-decoder to multi-scale reconstruction networks. The dimensions of each layer are presented at the topmost, indicating the expansion and shrinkage as an input astigmatism image is being sequentially transformed into next layers, and the channels at different levels are shown on top of each corresponding tensor. In this configuration, a single astigmatism image of size 512×512 is encoded into a high dimension tensor of $2 \times 2 \times 2048$, then this tensor is decoded sequentially into 8 different sizes of images to match different frequency components of the desired target image. By matching the intensities between the output images and the target images, all the weighted connections are updated using the technique of neural network back-propagation. (B) The convergence curves presented to demonstrate the significant advantages of an MDCNN over a classic U-Net when training to recover phases from a same scheme: starting from the same initial guess, the mean-absolute-error (MAE) of the MDCNN (the 512×512 branch) drops dramatically from 0.23 to 0.07 in 100 iterations, while the U-Net seems getting stuck at a local minima around a loss of 0.2.

makes the optimal coefficients a_i signal dependent. The second problem is, Eq.2 does not necessarily have a unique solution unless appropriate boundary conditions[24] are met.

Directly matching measurements $I(\vec{r}, z)$ to phases to predict, MDCNN applications address phase problem in an explicit way. Instead of digging into the complex reconstruction algorithms for Eq.2 and careful approximations for Eq.1, MDCNN applications solve this problem by training a neural network matching one or more intensities $I(\vec{r}, z_i)$ to the desired phase $\phi(\vec{r}, z)$. For such a generic and straightforward workflow, many other inversion problems can directly take this framework as bare bones to start with.

Several applications have been trained to do phase retrieval. As are shown in Fig.2, the first one (MDCNN-I), serving as a prove of concept in a same way as TIE, predicts phases from intensity gradient; the second one (MDCNN-II) predicts from a defocused image; the third one (MDCNN-III) predicts from an astigmatic image acquired after a cylindrical lens; and the last one (MDCNN-IV) phases from 32 astigmatic images by rotating the cylindrical lens. For all the numeric experimental results shown in this figure, the first row presents the input image; the second row gives out the prediction with mean-absolute-error (MAE) as a metric value between the ground phase ϕ_g and the predicted phase ϕ_p , which is defined as

$$MAE(\phi_p, \phi_g) = \sum_{m=1}^M \sum_{n=1}^N |\phi_p[m][n] - \phi_g[m][n]|, \quad (3)$$

where M and N are the row and column of the phase image; the third row gives out the ground truth images desired to be restored; the last row gives the normalised cross-correlation between the prediction and the ground-truth images in Fourier space using the Fourier ring correlation (FRC)[25]:

$$FRC(r) = \frac{\sum_{r_i \in r} F_p(r_i) F_g^*(r_i)}{\sqrt{\sum_{r_i \in r} |F_p(r_i)|^2 |F_g(r_i)|^2}}, \quad (4)$$

where F_g and F_p are the Fourier transform of ϕ_g and ϕ_p , and r is the radius. Judging from FRC, all these applications give good performance in predicting low-frequency components.

MDCNN-II is further bench-marked with a conventional TIE solver. Using same experimental data set presented in Fig.3, comparing with the fitting result from GPTIE[16] in Fig.3(A), two similar results predicted by

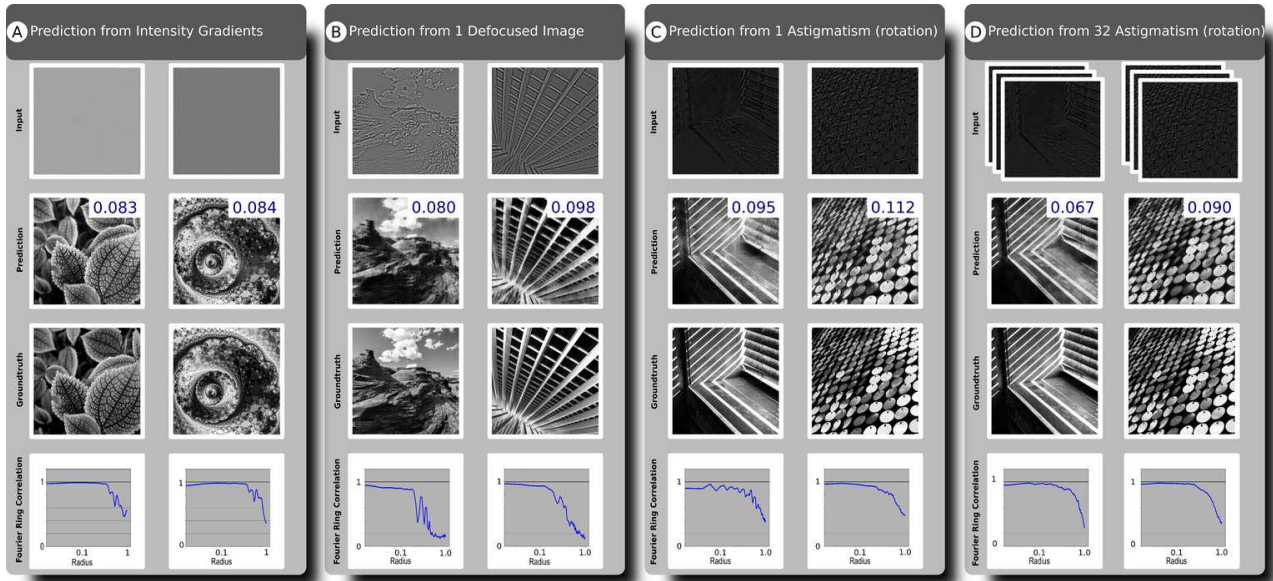


Figure 2: Numerical performance of MDCNNs with different phase retrieval schemes. The first row: input(s) of MDCNN. The second row: phase predicted by MDCNN, the value printed in the upper-right corner is the mean-absolute-error. The third row: the ground truth phase. The last row: Fourier ring correlation between the ground truth and the predicted phases. (A): (MDCNN-I) Phase prediction from intensity gradients. (B): (MDCNN-II) Phase prediction directly from intensity of one image. (C): (MDCNN-III) Phase prediction from a measured astigmatic image after a cylindrical lens. (D): (MDCNN-IV) Phase prediction from 32 astigmatic images after a cylindrical lens, with 32 different rotations.

MDCNN-II are presented in Fig.3(B) and (C), which are predicted from measured intensities at $-108\mu\text{m}$ and $108\mu\text{m}$ independently. Thus, MDCNN is capable of phase retrieval on par with conventional TIE applications without sacrificing the quality.

While in the deterministic systems the forward simulations have unique solutions, the inverse problems do not necessarily have. Inverse problems most often are ill-posed. A consequence is that arbitrarily small changes in the data may lead to arbitrarily large changes in the solution. As errors are inevitable in the numeric treatment of inverse problems, we have to use stabilizing procedures for successful treatments of ill-posed problems. Comparing the performance of MDCNN-III and MDCNN-IV in Fig.2(C) and (D), the one taking 32 inputs generally gives better performance in the MAE and FRC metrics, indicating multiple observations can obsolete the singularities in the phase problem. This can be observed from the prediction result presented in Fig.3(D), where phases are predicted from two measured intensities, providing more details than the results shown at Fig.3(B) and (C), which are predicted from single measurement.

Solving problems not well mathematically modelled

When solving phase problems with MDCNN, we make use of the knowledge of physical schematics for the dataset simulation to train the weights of MDCNN architecture we specified. However, in some of the real world problems, we do not have enough knowledge of what effects should be taken into consideration, and therefore we have a problem in generating the training dataset.

Modern scanning transmission electron microscopy (STEM) can provide sub-ångströms images[26], but this is limited by, e.g., the beam sensitivity of the specimen. Lowering the electron dose results in noisy images with a poor signal-to-noise ratio (SNR), and complicated extraction of relevant specimen information. Therefore it is important to reconstruct clear images from noisy observations[27]. Due to the complex unknown variables related to the concrete experimental set-ups, real-world STEM image de-noising problem is too complex for a single monolithic de-noising algorithm. Many conventional algorithms exist to address this problem, but they heavily depend on domain specific knowledge and/or a priori information[28, 29, 30]. Recent variations[31, 3, 32, 33] based on DCNN work well by directly matching a noisy input image to a clean output image. However, they assume high SNR images with known noise sources, and most of them only consider a single signal independent noise source with known level.

We demonstrate how to solve such an inverse problem that is not well mathematically modelled: de-noising images with unknown noises acquired with very poor SNR. If all the noise categories are known to us, even if these noises are signal dependent and are of unknown levels, our MDCNN architecture is sufficient. To prove this point, we simulate noisy images with assumed noise sources (Poisson noise with a parameter randomly distributed in

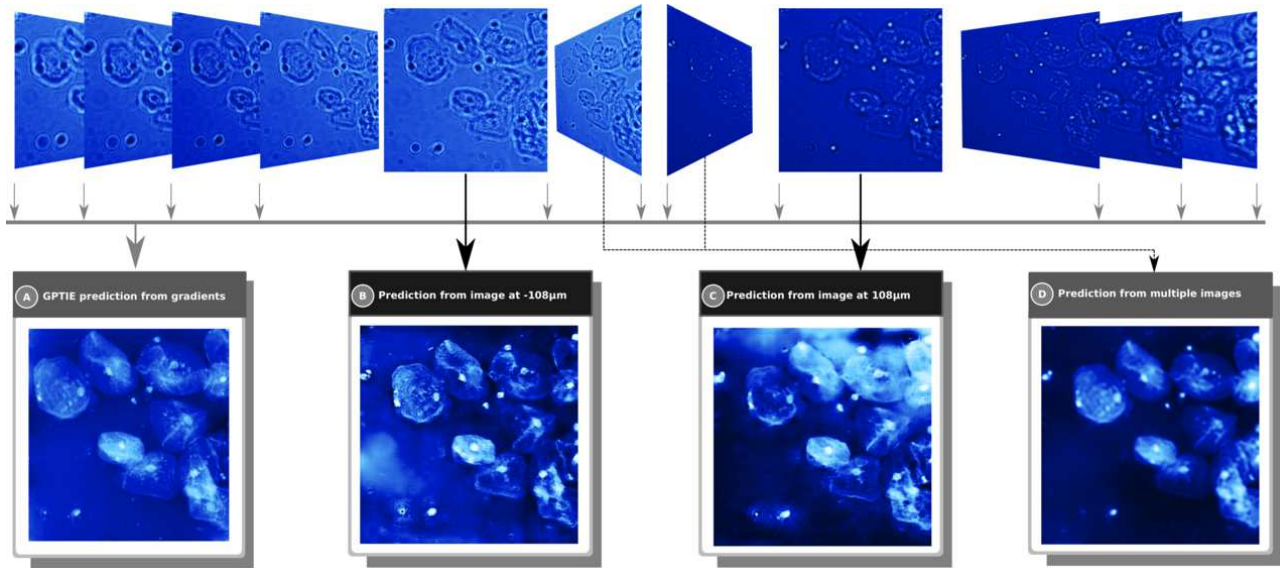


Figure 3: Solving phase problem using MDCNN from a series of defocused images. From 129 images 1024×1024 of a pure phase object acquired equally spaced by $d_z = 4 \mu\text{m}$ from $-256 \mu\text{m}$ to $256 \mu\text{m}$, MDCNNs can make predictions from anyone of them, significantly reduces the observations required. (A): GPTIE reconstruction from gradients estimated using 129 images from different focal planes, adopted from [16]. (B): (MDCNN-II) Phase prediction from a single astigmatism image at a distance of $-108 \mu\text{m}$ from the focal plane. (C): (MDCNN-II) Phase prediction from a single astigmatism image at a distance of $108 \mu\text{m}$ from the focal plane. (D): (MDCNN-V) Phase prediction from two measured intensities at $-52 \mu\text{m}$ and $52 \mu\text{m}$, showing good performance on par with conventional TIE reconstruction algorithm demonstrated in (A).

range $[1, 10]$, dark current noise randomly in range $[0\%, 10\%]$ and saturation noise randomly in range $[90\%, 100\%]$ from previous knowledge, then we employ the same architecture demonstrated in Fig.1 to train a direct de-noising neural network (MDCNN-VI). As is shown in the upper part of Fig.4(A), we find that, this de-noising network performs well in the validation dataset; the low frequency components are predicted very well, but it makes bad prediction with quite a lot of background noise remaining, as is presented in Fig.4(B), when applied to an experimental image with no more than 10 electrons per pixel in average.

To address this problem, we design an indirect de-noiser (MDCNN-VII): first, we pre-process the input images with four conventional low-pass-filters (LPFs), then assemble these prepossessed images to map a clean image to do the training. With two uniform LPFs and two Gaussian LPFs, we convert our noisy image into a pack of blurred images, transforming a de-noising problem into a de-blurring problem. The advantage of this method is that conventional de-blurring applications usually reconstruct a single clear output from a single blurred input, our architecture takes the advantages of much more images fed to the input layer, thus is expected to perform much more robust than the others. Accelerated with a commercially available graphics processing unit (GPU) such as a Nvidia GTX 1080 Ti, this neural network can even make real-time prediction, relieving lots of pressure in the post-processing stage. A detailed data flow of this scheme is demonstrated in the lower part of Fig.4(A): we first simulate noisy images from ground truth images by appending salt-and-pepper noise, then get 4 blurred images after 4 different LPFs to train MDCNN-VII. When passing a same noisy image to this MDCNN nearly all the background intensities are diminished, as is shown in Fig.4(B). Moreover, even if the inputs are recorded under different configurations with unknown types and different levels, this application can still make robust predictions, some of the predictions are presented in Fig.4(C).

Conclusion

The superpower of MDCNN applications in solving different inverse problems have been demonstrated. Solving a phase problem from one or more images in line with the popular TIE scheme, we match the performance of our MDCNN with a state-of-the-art solver, showing it is capable to solve problems with complex mathematical models, and can efficiently make use of the multiple measurements to reduce the side effects shipped from ill-posed problems. Solving a blind de-noising problems, we show that, MDCNN can provide a robust prediction, taking the advantages of the extensibility of multiple input channels in a convolutional neural network. When training the MDCNN architectures presented above, the weights are updated by minimizing the MAE between the prediction and the ground truth. This kind of error metric tends to give out blurry outputs loss of sharpness. Further research, very possibly in how to integrate generative adversarial networks[2, 8], is highly desirable

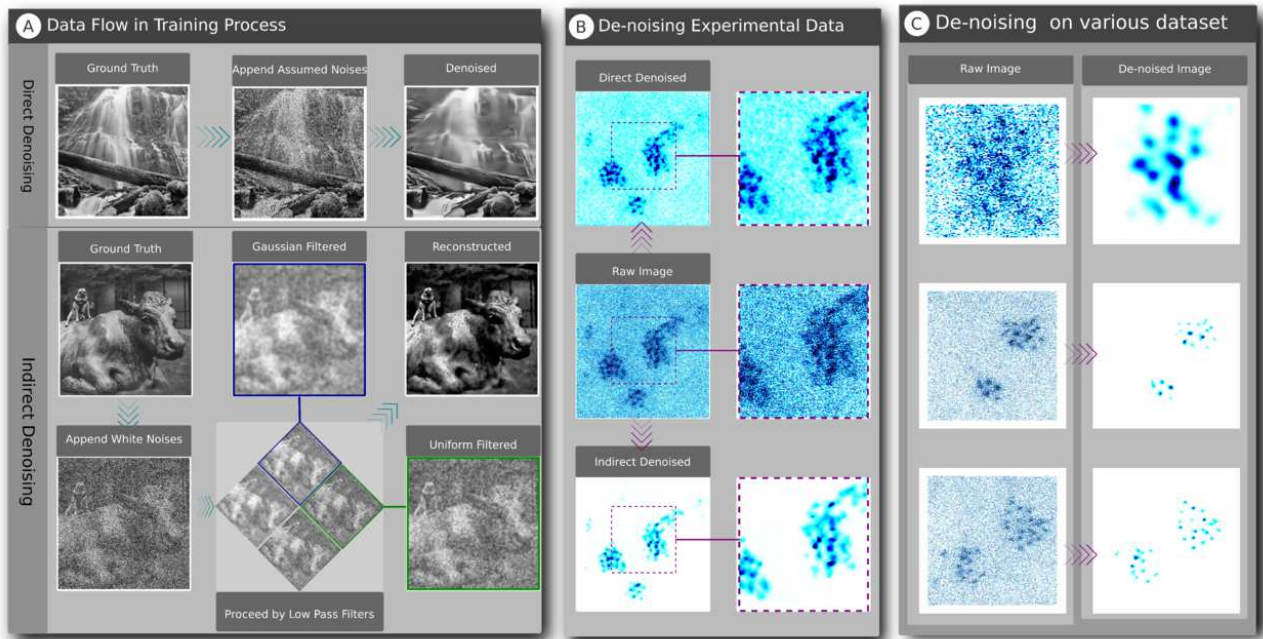


Figure 4: MDCNN as de-noising applications. (A): data flows presented to show the differences between a direct de-noising method and an indirect de-noising method. The first row shows that, when training a direct de-noising neural network (MDCNN-VI), three random noises (Poison noise, dark current noise and saturation noise) are appended to the ground truth image to generate the training set; the second and last rows show that, when training an indirect de-noising neural network (MDCNN-VII), random noises (can be any of Poison noise, Gaussian noise, salt-and-pepper noise or their combinations) are appended to the ground truth image, then send to a set of 4 low pass filters (a uniform filter of dimension 5×5 , a uniform filter of dimension 7×7 , a Gaussian filter with parameter $\sigma = \sqrt{20}$ and a Gaussian filter with parameter $\sigma = \sqrt{30}$) to generate 4 images as the training set. Both the direct and indirect can achieve good performance in generated validation set. (B): MDCNN-VII predicts a much clearer result when working with a real-world experimental image. (C): performance of MDCNN-VII as an indirect de-noiser on various experimental dataset, with an average electrons per pixel varying from 1 to 10. The first one is of dimension 128×128 , taken at 150 fps; The rest two are of dimension 512×512 , taken at 15 fps.

to address this problem. In the wake of this robust, generic purposed architecture, a branch of new schemes dealing with different kinds of inverse problems are supposed to emerge, broadening the scope of inverse problem applications. Equipped with MDCNN, scientists and engineers are expected to think more about *what* would be interesting to figure out, rather than *how* to find out something interesting. Intelligence resource is scarce for most of the time, when the computation resource is getting adequate, we should adapt ourselves to this trend.

Acknowledgement

F. W. and R. E. acknowledge founding from the European Research Council (ERC) under the European Union's Horizon 2020 program (grant agreement No. 681312). **TODO: More founding from HU?**

References

- [1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning 521 (7553) 436–444. [doi:10.1038/nature14539](https://doi.org/10.1038/nature14539).
- [2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, pp. 2672–2680.
- [3] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising 26 (7) 3142–3155. [doi:10.1109/TIP.2017.2662206](https://doi.org/10.1109/TIP.2017.2662206).
- [4] M. R. Teague, Deterministic phase retrieval: A Green's function solution 73 (11) 1434–1441. [doi:10.1364/JOSA.73.001434](https://doi.org/10.1364/JOSA.73.001434).
- [5] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, Imagenet large scale visual recognition challenge 115 (3) 211–252.

- [6] I. Krasin, T. Duerig, N. Alldrin, A. Veit, S. Abu-El-Hajja, S. Belongie, D. Cai, Z. Feng, V. Ferrari, V. Gomes, Openimages: A public dataset for large-scale multi-label and multi-class image classification 2 (6) 7.
- [7] C. Dong, C. C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks 38 (2) 295–307.
- [8] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, CVPR.
- [9] Y. D. Hezaveh, L. P. Levasseur, P. J. Marshall, Fast automated analysis of strong gravitational lenses with convolutional neural networks 548 (7669) 555–557. [doi:10.1038/nature23463](https://doi.org/10.1038/nature23463).
- [10] A. Sinha, J. Lee, S. Li, G. Barbastathis, Lensless computational imaging through deep learning 4 (9) 1117–1125.
- [11] G. E. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks 313 (5786) 504–507. [doi:10.1126/science.1127647](https://doi.org/10.1126/science.1127647).
- [12] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, pp. 234–241.
- [13] Z.-Q. J. Xu, Y. Zhang, Y. Xiao, Training behavior of deep neural network in frequency domain.
- [14] N. Rahaman, D. Arpit, A. Baratin, F. Draxler, M. Lin, F. A. Hamprecht, Y. Bengio, A. Courville, On the Spectral Bias of Deep Neural Networks.
- [15] K. Stüben, U. Trottenberg, Multigrid methods: Fundamental algorithms, model problem analysis and applications, in: Multigrid Methods, Springer, pp. 1–176.
- [16] Z. Jingshan, R. A. Claus, J. Dauwels, L. Tian, L. Waller, Transport of Intensity phase imaging by intensity spectrum fitting of exponentially spaced defocus planes 22 (9) 10661. [doi:10.1364/OE.22.010661](https://doi.org/10.1364/OE.22.010661).
- [17] J. Karle, Recovering phase information from intensity data 232 (4752) 837–843, 00045.
- [18] D. Gabor, A New Microscopic Principle. [doi:10.1038/161777a0](https://doi.org/10.1038/161777a0).
- [19] A. Lubk, G. Guzzinati, F. Börrnert, J. Verbeeck, Transport of intensity phase retrieval of arbitrary wave fields including vortices 111 (17) 173902.
- [20] A. Parvizi, W. V. den Broek, C. T. Koch, Recovering low spatial frequencies in wavefront sensing based on intensity measurements 2 (1). [doi:10.1186/s40679-016-0017-y](https://doi.org/10.1186/s40679-016-0017-y).
- [21] D. Van Dyck, M. Op de Beeck, W. Coene, A new approach to object wavefunction reconstruction in electron microscopy 93 103–103.
- [22] A. Eljarrat, J. Müller, M. R. S. Huang, C. T. Koch, Multi-focus TIE algorithm including partial spatial coherence and overlapping filters 26 (9) 11819. [doi:10.1364/OE.26.011819](https://doi.org/10.1364/OE.26.011819).
- [23] C. Zuo, Q. Chen, Y. Yu, A. Asundi, Transport-of-intensity phase imaging using savitzky-golay differentiation filter - theory and applications 21 (5) 5346. [doi:10.1364/OE.21.005346](https://doi.org/10.1364/OE.21.005346).
- [24] T. E. Gureyev, A. Roberts, K. A. Nugent, Partially coherent fields, the transport-of-intensity equation, and phase uniqueness 12 (9) 1942–1946.
- [25] G. Harauz, M. van Heel, Exact filters for general geometry three dimensional reconstruction, in: Proceedings of the IEEE Computer Vision and Pattern Recognition Conf, Vol. 73, pp. 146–156.
- [26] Y. Jiang, Z. Chen, Y. Han, P. Deb, H. Gao, S. Xie, P. Purohit, M. W. Tate, J. Park, S. M. Gruner, V. Elser, D. A. Muller, Electron ptychography of 2D materials to deep sub-Ångström resolution 559 (7714) 343–349. [doi:10.1038/s41586-018-0298-5](https://doi.org/10.1038/s41586-018-0298-5).
- [27] T. R. Henninen, B. Marta, P. Daniele, Feng Wang, E. Rolf, Dynamics and crystallinity of sub-nm platinum clusters at elevated temperatures, submitted.
- [28] N. Mevenkamp, P. Binev, W. Dahmen, P. M. Voyles, A. B. Yankovich, B. Berkels, Poisson noise removal from high-resolution STEM images based on periodic block matching 1 3. [doi:10.1186/s40679-015-0004-8](https://doi.org/10.1186/s40679-015-0004-8).

- [29] A. Buades, B. Coll, J.-M. Morel, A review of image denoising algorithms, with a new one 4 (2) 490–530.
- [30] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-D transform-domain collaborative filtering 16 (8) 2080–2095.
- [31] D. Ulyanov, A. Vedaldi, V. Lempitsky, Deep image prior, arXiv:1711.10925.
- [32] K. G. Lore, A. Akintayo, S. Sarkar, LLNet: A deep autoencoder approach to natural low-light image enhancement 61 650–662.
- [33] S. W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. T. Barron, F. Kainz, J. Chen, M. Levoy, Burst photography for high dynamic range and low-light imaging on mobile cameras 35 (6) 192.
- [34] I. Krasin, T. Duerig, N. Alldrin, A. Veit, S. Abu-El-Haija, S. Belongie, D. Cai, Z. Feng, V. Ferrari, V. Gomes, Openimages: A public dataset for large-scale multi-label and multi-class image classification 2 (6) 7.
- [35] A. L. Maas, A. Y. Hannun, A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: Proc. Icml, Vol. 30, p. 3.
- [36] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37, JMLR. org, pp. 448–456.
- [37] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization.
- [38] F. Chollet, et al., Keras, <https://keras.io> (2015).
- [39] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, Tensorflow: A system for large-scale machine learning., in: OSDI, Vol. 16, pp. 265–283.
- [40] F. Wang, Demo code for multi-scale deep convolutional neural networks, <https://github.com/fengwang/MDCNN-I> (2018).

Method

Dataset. The training data for these experiments could come from any source. All applications use 1024 or 2048 images randomly sampled from Open Images Dataset[34], which are gray-scaled then scaled and/or cropped to size 512×512 , except for the MDCNN-IV only 512 images are used to fit into the host memory, and for the de-noising applications, MDCNN-VI and MDCNN-VII, half of the training images are simulated with atom columns per image randomly generated in range $[50, 300]$, and atoms per column randomly in range $[1, 10]$. Little performance improvement has been observed by simply increasing the training set.

Data preparation. For MDCNN-I, phase retrieval from intensity gradients, the inputs are simulated with a convolution operation:

$$L = I * K,$$

where L is the desired second order Laplacian, I is the image drawn from our dataset, and K is the convolution kernel

$$K = - \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}.$$

For MDCNN-II to MDCNN-V, the input images are simulated using Fourier optics with element-wise production

$$T = \left| \mathcal{F}^{-1}(P(\phi, \gamma) \odot CTF) \right|,$$

where CTF is for the contrast-transfer-function (CTF) and $P(\phi, \gamma)$ is the Fourier transform of a complex image in range $[0, i\gamma]$ is simulated as

$$P(\Phi, \gamma) = \mathcal{F}(e^{i\gamma\Phi}),$$

where Φ is an input image, and the scaling factor γ is 1.5 for MDCNN-II and MDCNN-V, and 2π for MDCNN-III and MDCNN-IV. The CTF of a defocused image at distance z is

$$CTF_d(z) = \sin(\pi \lambda (q_x^2 + q_y^2) z),$$

where q_x and q_y is the spacial frequencies in x and y direction. The CTF of an astigmatism abbreviation is

$$CTF_a(z, \alpha) = e^{-\frac{z[(q_x^2 + q_y^2) \cos 2\alpha + 2q_x q_y \sin 2\alpha]}{4k}},$$

where z is the defocus depth, α is the rotation angle of the cylinder lens, k is the wave number .

For the direct de-noising applications, MDCNN-VI and MDCNN-VII, a saturation function is introduced

$$S(x, \alpha) = \begin{cases} 0 & x \leq 0, \\ x & 0 < x < \alpha, \\ \alpha & x \geq \alpha \end{cases}$$

then simulate the noisy training set using

$$O = S\left(P(u(1, 10) * I) + u(0, 0.1), u(0.9, 1.0)\right),$$

where I is the ground truth image in $[0, 1]$, $P(\lambda)$ is a Poisson variate generator with a parameter λ , $u(x, y)$ is a uniform random variate generator in range $[x, y]$.

For the indirect de-noising application, we simulate the noisy images using

$$O = S(I + u(-1.0, 1.0), 1.0).$$

then pass them to 4 LPFs simulated using matrix convolution with reflective paddings. These 4 LPFs are

$$\begin{aligned} f_1(r, c) &= 1 & r \in [1, 5] \text{ and } c \in [1, 5], \\ f_2(r, c) &= 1 & r \in [1, 7] \text{ and } c \in [1, 7], \\ f_3(r, c) &= e^{-\frac{(r-8)^2 + (c-8)^2}{\sqrt{20}}} & r \in [1, 15] \text{ and } c \in [1, 15], \\ f_4(r, c) &= e^{-\frac{(r-8)^2 + (c-8)^2}{\sqrt{30}}} & r \in [1, 15] \text{ and } c \in [1, 15]. \end{aligned}$$

Network configuration. The essential part of MDCNN is the low-frequency branches, which enable the convergence of the network. The choice of the U-Net has many degrees of freedom. The convolution kernel sizes are generally designed small, for example 3×3 or 4×4 , just to save memory and operations, except in the input layer and the output layer of dimension 512×512 , the kernel size is 31×31 , a quite large kernel to cover more features. When down-scaling layers, the conventional convolution operation with stride step 2 or average pooling can be used, and for up-scaling, a convolution with a stride 2 or a direct up-sampling can be used. sigmoid activation function is used in all the output layers, and rectified linear unit[35] is used in the hidden layers. Batch normalization[36] after is optional when the batch size is large. Adaptive momentum method[37] is employed to train the MDCNNs with a learning rate starting at 0.001. All applications are trained under Linux environment implemented using Keras[38] with Tensorflow[39] back-end. The workstation is equipped with two Nvidia GTX 1080 Ti graphics processing units (GPUs) and one Intel Xeon Silver 4114 central processing unit (CPU).

Code availability. An open-source version MDCNN-I is available online[40].