Learning to Synthesize: Robust Phase Retrieval at Low Photon Counts

MO DENG1,*, SHUAI LI2,†, ALEXANDRE GOY2,‡, IKSUNG KANG1, AND GEORGE BARBASTATHIS2,3

1Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
2Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
3Singapore-MIT Alliance for Research and Technology (SMART) Centre, Singapore 117543, Singapore
†Current address: SenseTime Research, 2550 N 1st Street, Suite 300, San Jose, CA 95131, USA
‡Current Address: Omnisens SA, Riond Bosson 3, 1110 Morges, VD, Switzerland
*Corresponding author: modeng@mit.edu

Compiled July 30, 2019

The quality of inverse problem solutions obtained through deep learning [Barbastathis et al, 2019] is limited by the nature of the priors learned from examples presented during the training phase. In the case of quantitative phase retrieval [Sinha et al 2017, Goy et al 2019], in particular, spatial frequencies that are underrepresented in the training database, most often at the high band, tend to be suppressed in the reconstruction. Ad hoc solutions have been proposed, such as pre-amplifying the high spatial frequencies in the examples [Li et al, 2018]; however, while that strategy improves resolution, it also leads to high-frequency artifacts as well as low-frequency distortions in the reconstructions. Here, we present a new approach that learns separately how to handle the two frequency bands, low and high; and also learns how to synthesize these two bands into the full-band reconstructions. We show that this “learning to synthesize” (LS) method yields phase reconstructions of high spatial resolution and artifact-free; and it is also resilient to high-noise conditions, e.g. in the case of very low photon flux. In addition to the problem of quantitative phase retrieval, the LS method is applicable, in principle, to any inverse problem where the forward operator treats different frequency bands unevenly, i.e. is ill-posed. © 2019 Optical Society of America under the terms of the OSA Open Access Publishing Agreement

http://dx.doi.org/10.1364/optica.XX.XXXXXX

1. INTRODUCTION

The retrieval of phase from intensity is one of the most important and most challenging problems in Classical Optics. The utility of phase retrieval stems from the fact that it allows the shape of transparent objects, biological cells in particular, to be quantified in two and three spatial dimensions using visible light [1, 2]. In the x-ray band, quantitative phase imaging is also useful because phase contrast in tissue is orders of magnitude higher than attenuation contrast [3, 4]. The same argument can be made for identification of liquids [5] and semiconductor materials for integrated circuit characterization and inspection [6].

There are two well-known challenges in phase retrieval. Firstly, for the phase of the optical field to be well-defined, the illumination needs to be temporally and spatially coherent to a fairly good approximation; this is especially difficult with x-rays. One way to relax this requirement is to acknowledge that it is usually the phase delay through the object that is of interest, and seldom the phase of the optical field itself. The former can be obtained even from partially coherent light [7] but requires the correlation function (mutual intensity) to be measured, which is often problematic because of low intensity. The second challenge is that, since only the intensity of a light beam is observable, the phase may only be inferred indirectly from intensity measurements. Computational approaches to this operation may
be classified as interferometric/holographic [8, 9], where a reference beam is provided; and non-interferometric, or reference-less such as direct/iterative [10, 11] and ptychographic [12, 13], which are both nonlinear, and transport-based [14, 15], where the problem is linearized through a hydrodynamic approximation. Direct methods attempt to retrieve the phase from a single raw intensity image, whereas the transport and ptychographic methods implement axial and lateral scanning, respectively. What reference-less methods have in common is the need to obtain intensity measurements at some distance away from the conjugate plane of the object, i.e. with a small defocus. Direct measurement with defocus is the approach we take here.

All computational phase retrieval approaches, interferometric and non-interferometric, involve the solution of a nonlinear and highly ill-posed inverse problem. For direct phase imaging, which is a nonlinear problem—see Section 2.A—the classical Gerchberg-Saxton-Fienup (GSF) algorithm [10, 11, 16] and its variants [17] are widely used. The main idea is to start with a random estimate for the unknown phase distribution and then to iteratively update it until the modulus-squared of its Fourier (or Fresnel) transform matches the observed intensity. For well-behaved phase fields, the iteration usually converges to the correct phase [18, 19]. Alternatively, the Wiener-Tikhonov functional minimization approach, described in Section 2.B, exploits prior knowledge about the class of phase objects being imaged to combat noise artifacts. A modern and popular implementation is compressive imaging [20–24] that utilizes sparsity priors, valid when the object is expressed in the appropriate set of basis functions. If the sparsifying set is unknown, it can be learned from examples according to the k-SVD method or other dictionary approaches [25–28].

In 2010, [29] proposed a deep neural network in a recurrent scheme to learn the prior from examples, as an alternative to the dictionary approach. Subsequently, generator-discriminator competition [30] was adopted as a more secure means of learning the statistical relationship between valid objects and their forward models; and the recursion was unfolded into a cascade for better numerical stability [31]. In these schemes, the physical model of the measurement is taken explicitly into account as a projection operator applied to the reconstruction estimate repeatedly at each recursion or cascade stage. This generalization of dictionaries to deep learning has been successful in a number of linear inverse problems, most notably superresolution [32–34] and tomography [35, 36].

Recently, deep learning regression has been investigated for application to nonlinear inverse problems, in particular phase retrieval: direct [37–39], holographic [40, 41], and ptychographic [42, 43]. The idea, described briefly in Section 2.B, is to train a deep neural network (DNN) in supervised mode from examples of phase objects and their intensity images so that, after training, given an intensity image as input, the DNN outputs an estimate of the phase object. In this case, the physical model is either learned implicitly together with the prior from the examples [37, 38]; or incorporated as a pre-processor (“Approximant”) [39–41]. An interesting alternative method for the inverse problem, also nonlinear, of reconstructing the three-dimensional (3D) refractive index distribution from intensity projections, is to define the DNN architecture according to the strong scattering model and store the refractive index values as weights of the DNN after training [44]. This “index-storing” DNN itself was subsequently used as Approximant to a traditional DNN for improving the estimates in 3D distributions with exceptionally small and high-contrast features or when the range of available angles of projection is severely limited [45]. Extensive reviews of deep learning use for inverse problems can be found in [36, 46, 47].

Here, we propose a new DNN-based computational architecture for phase retrieval with the unique feature of processing low and high spatial frequency bands as separate channels with two corresponding DNNs trained from an original object database and a high-pass filtered version of the database, respectively. Subsequently, the outputs of the two channels are recombined using a third DNN also specifically trained for this task. The motivation for this new approach is an earlier observation [38] that nonlinearities in DNN training and execution algorithms tend to amplify imbalances in the spatial frequency content of the training database and in the way different spatial frequencies are treated as they propagate through the physical optical system; this amplified imbalance typically results in the lower spatial frequencies becoming dominant and ultimately limiting resolution of fine spatial features in the reconstructions. A more detailed overview of this phenomenon can be found in Section 2.C. Because the essential feature of our new proposed technique is the synthesis of the two spatial bands through a trained DNN, we refer to it as “learning to synthesize” (LS).

Splitting the spatial frequency content into several bands and processing the bands separately has a long history in signal processing [48–55]. For image reconstruction, dual band processing has been used in fluorescence microscopy [56–58] and phase retrieval [59]. However, these cases, unlike ours, required structured illumination. In the context of learning-based inversion, a dual channel method has been tried for superresolution [60] to be understood as upsampling) albeit the two processed channels were combined as a simple convex sum to form the final image. By contrast, the LS method presented here uses a learned nonlinear filter, implemented as a third DNN trained to optimally recombine the two channels according to the spectral properties of the class of objects that the training database represents.

In addition to requiring a single raw image to retrieve the phase through a learned recombination of the spectral channels, the LS method presented here has the desirable property of resilience to noise, especially in the case of weak photon flux down to a single photon per pixel. We achieved that by using an Approximant filter [39] to pre-process the raw image before submitting it to the two spectral channels. The Approximant produces an inverse estimate that expressly uses the physical model (a single iteration of the GSF algorithm in [39] and here). For very noisy inputs, the Approximant is of very poor quality; yet, if the subsequent learning architecture is trained with this low-quality estimate as input, the final reconstruction results are significantly improved. The LS method with Approximant, as presented here, drastically improves over [39], especially in the reconstruction of fine detail, as [39] did not use separate spectral channels to rebalance frequency content.

The detailed implementation of the LS method is described in Section 3, where we also show how to optionally include the Approximant as initial estimate for input to the LS scheme. Results from experiments are presented in Section 4 with detailed characterization of the LS method’s behavior under different noise conditions. Conclusions and suggestions for future work are in Section 5.

2. PROBLEM FORMULATION

A. Phase retrieval

Let

\[ \psi_{\text{obj}}(x, y) = f(x, y)e^{i f(x, y)} \]
denote the complex transmittance of an optically thin object of modulus response \( t(x,y) \) and phase response \( f(x,y) \), and let \( \psi_{\text{inc}}(x,y) \) denote the coherent incident field of wavelength \( \lambda \) on the object plane. The noiseless intensity measurement \( g_0(x,y) \) (also referred to as noiseless raw image) is carried out on the detector plane located at a distance \( z \) away from the object plane, and can be written as

\[
g_0(x,y) = |F_z [\psi_{\text{inc}}(x,y) \psi_{\text{obj}}(x,y)]|^2 = H_0 f(x,y),
\]

(1)

where \( F_z [\cdot] \) denotes the Fresnel (paraxial) propagation operator for distance \( z \), i.e. the convolution

\[
F_z[\psi] = \psi(x,y) \ast \frac{e^{i2\pi z/\lambda}}{i\lambda z} \exp \left\{ i\pi x^2 + y^2 \right\} ;
\]

(2)

and \( H_0 \) is the (nonlinear) noiseless forward operator. Alternatively, \( F_z \) may be expressed in the spatial frequency domain \((\nu_x, \nu_y)\) as

\[
F_z[\psi] = \mathcal{F}^{-1} \left\{ \mathcal{F} \{ \psi \} \exp \left\{ -i\pi \lambda z \left( \nu_x^2 + \nu_y^2 \right) \right\} \right\},
\]

(3)

where \( \mathcal{F} \) denotes the 2D (spatial) Fourier transform operator and \( \mathcal{F}^{-1} \) its inverse.

We are interested in weakly absorbing objects, i.e. we assume \( t(x,y) \approx 1 \). In all the experiments described here, the illumination is also a normally incident plane wave \( \psi_{\text{inc}}(x,y) = 1 \). Therefore, to a good approximation, we may write

\[
g_0(x,y) = H_0 f(x,y) = |F_z [e^{if(x,y)}]|^2.
\]

(4)

This is what we refer to as the direct phase retrieval problem, which Gerchberg-Saxton and related algorithms solve iteratively [10, 16].

In practice, the measurement is subject to Poisson statistics due to the quantum nature of light; and to Gaussian thermal noise added by the photoelectric conversion process. We express the noisy measurement as

\[
g(x,y) = \mathcal{P} \left\{ p \frac{H_0 f(x,y)}{H_0 f} \right\} + \mathcal{N} \equiv H f,
\]

(5)

where \( \mathcal{P} (\cdot) \) denotes a Poisson random variable with mean \( \theta \) and \( \mathcal{N} \), a Gaussian random variable with zero mean and variance \( \sigma^2 \). The photon flux in photons per pixel per frame is denoted as \( p \) and the spatial average \( \langle H_0 f \rangle = \langle g_0 \rangle \) of the noiseless raw image in the denominator is necessary as normalization factor. The noisy forward operator is \( H \) and the purpose of phase retrieval is to invert \( H \) so as to recover \( f \) as accurately as possible, despite the nonlinearity and randomness present in the measurements.

**B. Solution of the inverse problem**

The Wiener-Tikhonov approach to solving inverse problems of the form \( g = H f \) is to obtain the estimate \( \hat{f} \) of the inverse as

\[
\hat{f} = \arg \min_{f} \left\{ D (H_0 f, g) + \Phi(f) \right\} .
\]

(6)

Here, \( D(H_0 f, g) \) is the fitness (or data-fidelity) term and \( D \) is a distance operator. Under the assumption of solely additive Gaussian noise, it is appropriate to use the \( L^2 \) norm

\[
D(H_0 f, g) = \| H_0 f - g \|_2^2
\]

\[
= \sum_{x,y} \left| H_0 f(x,y) - g(x,y) \right|^2.
\]

(7)

Somewhat less rigorously, but conveniently for mathematical manipulations, the same \( L^2 \) metric is used for the fitness term even for measurements strongly subject to non-Gaussian statistics, as we consider here. The topic of appropriate fitness metrics is beyond the scope of this paper; in any case, when machine learning is used to approximate \( D \), the dilemma of choosing a metric shifts to the loss function for training a deep neural network. We will address this latter problem in some detail in Section 3.A.

The second term \( \Phi(f) \) in (6) is the regularizer, or prior knowledge term. Its purpose is to compete with the fitness term in the minimization so as to mitigate ill-posedness in the solution. That is, the regularizer penalizes solutions that are promoted by the noise in the forward problem, as in (5) for example, but does not meet general criteria known \textit{a priori} for valid objects. The prior may be defined explicitly, e.g. as a minimum energy [61, 62] or sparsity [20–24] criterion; or learned from examples as a dictionary [25–28] or through a deep learning scheme [29, 31–43]...

Here, as in earlier works on direct phase retrieval [37–43], and due to the nonlinearity of the forward model, we adopt the End-to-End and Approximant methods. These we denote as

\[
\text{End-to-End:} \quad \hat{f} = \text{DNN}(g); \quad \text{and}
\]

\[
\text{Approximant:} \quad \hat{f} = \text{DNN}(\hat{f}^*),
\]

(8)

where DNN(\cdot) is the output of a deep neural network. In the End-to-End approach, the burden is on the DNN to learn from examples both the forward operator \( H \) and the prior \( \Phi \) so as to execute, in one shot, an approximation to the ideal solution (6). Training takes place in supervised mode, with known pairs of phase objects \( f \) and their raw intensity images \( g \) generated on a phase spatial light modulator (SLM) and measured on a digital camera, respectively. Note that training is generally very slow, taking several hours or days if a few thousand examples are used. However, after training is complete, the execution of (8) or (9) is very fast as it only requires forward (non-iterative) computations. This is one significant advantage over the standard way of minimizing the Wiener-Tikhonov functional (6) iteratively for each image.

When the inverse problem becomes severely ill-posed or the noise is extremely strong, the learning burden on the DNN becomes too high; then, generally, better results are obtained by training the DNN to receive as input the Approximant \( \hat{f}^* \) instead of the raw measurement \( g \). The Approximant is obtained through an approximate inversion of the forward operator; for example, in [40] it was implemented as a digital holographic backpropagation algorithm, whereas in [39] it was the outcome of a single iteration of the Gerchberg-Saxton algorithm [10]. While these Approximants \( \hat{f}^* \) generally do not look very good, especially in highly noisy situations [39], through training the DNN is able to learn a better association of \( \hat{f}^* \) with its corresponding true object \( f \) than it can learn with the noisy raw measurement \( g \).

**C. Spectral properties of training**

The design of deep neural networks is an active field of research and a comprehensive review of methods and caveats is well beyond the scope of this paper. We refer the reader to [36, 46, 47] for more extensive background and references. Here, we discuss the influence on the quality of training of the spatial power spectral density of the database where example are drawn from.

In both End-to-End and Approximant methods (8–9) the examples determine the object class prior to be learned by the
DNN. For instance, if we train on examples drawn from the ImageNet database [63] of natural objects, the prior learned is weaker than if we train on the MNIST [64] database of handwritten digits or a database of integrated circuit (IC) segments because both latter shapes are more restricted than natural images and with stronger spatial correlations across. Recent work [39, 65] has shown that, unsurprisingly, training with stronger priors results in more robust reconstructions as long as the test objects are drawn from the same class.

In [38], we addressed the influence of the spatial power spectral density (PSD) $S(v_x, v_y)$ of the example database on the quality of training. It is well known [66–71] that two dimensional (2D) images of natural objects, such as those contained in ImageNet [63], follow the inverse quadratic PSD law

$$S(v_x, v_y) = \frac{1}{v_x^2 + v_y^2}. \quad (10)$$

Other types of object classes of practical interest exhibit similar power-law decay, perhaps with slightly different exponents. This means that, if a neural network is trained on such an object class, higher spatial frequencies are presented less frequently to the DNN during the training stage. At face value, this is as it should be, since the relative popularity of different spatial frequencies in the database is precisely one of the priors that the DNN ideally should learn.

This understanding needs to be modified in the context of inverse problems because the representation of high spatial frequencies in the raw images is also uneven—typically, to the high spatial frequencies’ disadvantage. In the specific case of phase retrieval, higher spatial frequencies within the spatial bandwidth (as determined by the numerical aperture NA) have uniform transmission modulus but are more severely scrambled by the chirped oscillations of the transfer function (3). Thus, higher spatial frequencies suffer a double penalty [38]: their recovery becomes more sensitive to noise due to the scrambling; and they are less popular due to the inverse-square (or similar) PSD law so they are presented less frequently to the DNN’s training process. Moreover, since the DNN itself and its training routine are both highly nonlinear, there is an acute risk that any unevenness in the treatment of different spatial frequency bands may be amplified in the final result, eventually causing the lower frequencies to dominate. Ref. [38] attributed the inability of the Phase Extraction Neural Network (PhENN) [37] to resolve spatial features well within its admitted spatial bandwidth to this unequal treatment of spatial frequencies. They showed that PhENN’s resolution is approximately doubled by pre-filtering the training examples as to flatten their PSD. That is, during the training, each example $f(x, y)$ from the database was replaced with its filtered version

$$f_p(x, y) := \mathcal{F}^{-1}\left\{\mathcal{F}\{f(x, y)\} \times C(v_x, v_y)\right\}. \quad (11)$$

The transfer function was defined as the high-pass filter

$$C(v_x, v_y) = \sqrt{v_x^2 + v_y^2} \quad (12)$$

exactly compensating for the inverse-quadratic dependence (10) and flattening the spectrum. Raw images for training were correspondingly filtered as

$$g_p(x, y) = H f_p(x, y),$$

whereas, during the test, the un-filtered measurements (i.e., as received from the camera) were used to obtain the reconstructions.

Unfortunately, with this implementation amplification of high spatial frequency features, especially of artifacts caused even by weak noise, was also evident in the reconstructions. This is not surprising, since technically (11) trades off violating the prior in return for finer spatial resolution. The LS approach that we describe in the next section is meant to fix this problem.

3. METHODOLOGY OF LEARNING TO SYNTHESIZE

A. Description of the LS-DNN system

In [72], we proposed a two-step approach to tackle the difficulty of restoring high frequency components without introducing significant artifacts and distortions. Here we describe the two steps for training and execution in detail, as well as the design of the DNNs involved in the LS system. For unity in notation, we denote the input to the entire LS system as $\xi(x, y)$, to be understood as

$$\xi(x, y) = \begin{cases} g(x, y) & {\text{in the End-to-End scheme, and}} \\ f^*(x, y) & {\text{in the Approximant scheme.}} \end{cases} \quad (13)$$

We will discuss the Approximant implementation in more detail in section B below.

The two training steps are shown in block-diagram form in Figure 1. The first step consists of training two separate DNNs in parallel, as follows:

- DNN-L is trained to match unfiltered patterns $\xi^{(\text{n})}(x, y)$ at its input with the corresponding unfiltered example phase patterns $f^{(\text{n})}(x, y)$ as ground truth at its output (the superscript $\text{n}$ enumerates the examples).
- DNN-H is trained to match unfiltered patterns $\xi^{(\text{n})}(x, y)$ at its input to corresponding spectrally filtered versions $f_p^{(\text{n})}(x, y)$ of the ground truth examples $f^{(\text{n})}(x, y)$ at its output. The filter’s transfer function is chosen more generally than (12) as

$$C(v_x, v_y) = \left(\sqrt{v_x^2 + v_y^2}\right)^q. \quad (14)$$

We have investigated implementations with $q$ spanning a broad range, as we discuss in more detail below.

The output of DNN-L for a general test input $\xi(x, y)$ is denoted as $f^{\text{LF}}(x, y)$. Assuming similar training conditions, $f^{\text{LF}}$ matches the output of PhENN as presented in [37] in the End-to-End scheme or [39] in the Approximant scheme; that is, $f^{\text{LF}}$ is expected to be fairly accurate at low spatial frequencies but missing fine detail.

The output of DNN-H is denoted as $f^{\text{HF}}(x, y)$. Note that [38] required spatial pre-filtering the raw inputs $g$; here, we do not spatially pre-filter the input $\xi$ (i.e., $g$ or $f^*$ according to whether the End-to-End or Approximant scheme is used). We instead train DNN-H to produce the filtered output based on an unfiltered input. This leads to better generalization, because DNN-H is trained on the broadest set of possible images (whereas the training in [38] was on high-frequency containing images only). Moreover, using unfiltered inputs to DNN-H allows for the training process to be parallelized for better efficiency.

Depending on the value of $q$, the PSD of the patterns training DNN-H will be flat or almost flat. The output of DNN-H $f^{\text{HF}}(x, y)$ is expected to have better fidelity at fine spatial features of the phase objects. However, spectral flattening may also
generate artifacts due to over-learning the high spatial frequencies. Therefore, \( f^{\text{HF}} \) looks rather like a high-pass filtered version of the true object \( f \), which we found to be more beneficial for subsequent use in the LS scheme.

The second training step consists of combining the two partially accurate reconstructions \( f^{\text{LF}} \) and \( f^{\text{HF}} \) into a final estimate \( f(x, y) \) with uniform quality at all spatial frequencies, low and high up to the passband. To this end, we train the synthesizer DNN-S to receive \( f^{\text{LF}} \) and \( f^{\text{HF}} \) as inputs, and use the un-filtered examples \( f \) as the output. To avoid any further damage to the high-spatial frequency content in \( f^{\text{HF}} \), we bypass \( f^{\text{HF}} \) and present it intact to the last layer of DNN-S. By operating on \( f^{\text{HF}} \) alone, DNN-S learns how to treat the low frequency reconstruction so as to compensate for artifacts at all bands. The use of the synthesizer DNN-S also makes our results less sensitive to the choice of power \( q \) in the transfer function (14). We found that \( q \in [0.3, 0.7] \) can produce reconstructions of approximately equal quality, as will be presented in section 4.

![Fig. 1. LS-DNN schematic. (a) Training stage, (b) Test stage.](image)

After DNN-L, DNN-H and DNN-S have been trained, they are combined in the LS system and operated as shown in Figure 1(b). The input \( \xi(x, y) \) is passed to DNN-L and DNN-H in parallel fashion, and the respective outputs \( \hat{f}^{\text{LF}}(x, y) \) and \( \hat{f}^{\text{HF}}(x, y) \) are passed to DNN-S, which produces the final estimate \( \hat{f}(x, y) \). It is worth noting that it is not valid to lump the three networks in Figure 1(b) into a single network, due to their separate training schemes described above.

There is a wide variety of DNN structures one may choose to implement DNN-L, H and S. In this work, we use for DNN-L the same architecture in [39], a residual U-net architecture with skip connections [73]. For simplicity, DNN-H and DNN-S, are also chosen to be structures similar to DNN-L. The details of the implementations are in the Supplementary Material. We made these choices to enable specific and fair comparisons with the earlier works; alternative architectures are certainly possible within the LS scheme, though we judged a full exploration to be outside the scope of the present paper.

Training a neural network is typically implemented as a stochastic optimization procedure [74, 75] where the neural network’s internal coefficients (weights) are adjusted so as to minimize a metric of the distance between the actual output of the neural network and its desired output to a given input (training example). This distance is called training loss function (TLF). In the context of training to solve an inverse problem, the TLF is defined as

\[
\mathcal{L} = \sum_{n} D(\hat{f}^{(n)}, f^{(n)}),
\]

where the superscript \( n \) is again used to enumerate the examples, and the dilemma of choosing the appropriate metric operator \( D \) emerges.

It is generally accepted [37, 76–80] that the \( L^2 \) metric (also referred to as mean square error, MSE) is a poor choice that does not generalize well; i.e., deep neural networks trained with MSE do not perform well when presented with examples outside their training set. For image classification tasks, and in some early work on phase retrieval [37], the \( L^1 \) metric (mean absolute error, MAE) was used instead. In direct analogy with compressive sensing, the \( L^1 \) metric promotes sparsity in the internal connectivity of the neural network, and that leads to better generalization. However, [65] found that in highly ill-posed problems this benefit is eclipsed by the inability of MAE and pixel-wise metrics more generally to learn spatial structure priors about the object class that are crucial for regularization.

In this paper, we train DNN-L, H and S using the Negative Pearson Correlation Coefficient (NPCC) [65, 81, 82] as the TLF. This is defined as

\[
D_{\text{NPCC}}(a, b) \equiv \frac{(-1) \sum_{x,y} (a(x, y) - \langle a \rangle)(b(x, y) - \langle b \rangle)}{\left( \sum_{x,y} (a(x, y) - \langle a \rangle)^2 \right) \left( \sum_{x,y} (b(x, y) - \langle b \rangle)^2 \right)^{1/2}},
\]

where \( \langle a \rangle \) and \( \langle b \rangle \) are the spatial averages of the generic functions \( a(x, y), b(x, y) \). If \( a \) and \( b \) are uncorrelated, the expected value of \( D_{\text{NPCC}}(a, b) \) is zero, whereas if they are identical then \( D_{\text{NPCC}}(a, b) = -1 \). Thus, training the neural network minimizes the TLF toward \( \mathcal{L} \approx -N \), where \( N \) is the number of training examples.

The NPCC has been shown [83] to be more effective in recovering fine features than conventional loss functions such as the Mean Square Error (MSE), Mean Absolute Error (MAE) and the Structural Similarity (SSIM) index [84, 85]. However, the NPCC has the disadvantage that it is invariant to affine transformations to its arguments, i.e.

\[
D_{\text{NPCC}}(a, b) = D_{\text{NPCC}}(a_1 a + a_2, b_1 b + b_2)
\]

for arbitrary real numbers \( a_1, a_2, b_1, b_2 \). For quantitative phase retrieval, where the scale of phase difference matters, the affine ambiguity is resolved with a histogram equalization step after inversion [38].
B. Computation of the Approximant

It has been shown that, even under extreme noise conditions, just a single iteration of the Gerchberg-Saxton (GS) algorithm suffices as Approximant in scheme (9) for phase retrieval [39]. We elected to use the same approach here for the LS-DNN architecture. More recently, a comparative study [86] showed that higher iterates or regularized versions of GS do improve the appearance of the Approximant result \( \hat{f}^* \) but do not yield significant improvement in the end output \( f \) of the DNN. Similar conclusions hold for alternatives to GS, e.g. Gradient Descent. While these alternative schemes are interesting for the LS-DNN method, we chose to not pursue them here.

The general form of the \((k + 1)\)-th GS iterate from the \(k\)-th iterate is

\[
f[k + 1] = \arg \left\{ F_z^{-1} \left( \sqrt{\mathcal{F}} \exp \left\{ i \arg \{ F_z(\hat{f}^k) \} \right\} \right) \right\},
\]

where we have taken into account that \( \psi_{\text{inc}} = 1 \). Accordingly, our Approximant is

\[
\hat{f}^* = f[1] = \arg \left\{ F_z^{-1} \left( \sqrt{\mathcal{F}} \exp \left\{ i \arg \{ F_z(1) \} \right\} \right) \right\},
\]

where \( 1 \) denotes the function that is uniformly equal to one within the frame [86].

Figure 2 compares the 2D (log-scale) Fourier spectrum magnitude of a ground truth image (from ImageNet [63]), Approximant (19) computed without noise, and Approximant (19) computed from an input subject to Poisson statistics corresponding to average flux of one photon per pixel. We can see that although the single photon Approximant (which we will later use as the input to the LS-DNN) has a large support in its spectrum, it is the noise that dominates the mid-to-high frequency range. Therefore, the learning process still bears the burden of restoring the correct high frequency contents and relying heavily on high frequency priors, as our DNN-H does, is justified.

4. RESULTS

A. Experimental Apparatus and Data Acquisition

In each experiment carried out to train and test different LS-DNN schemes, 10,000 image objects from ImageNet [63] were successively projected on a phase SLM as phase objects (i.e., with phase value at each pixel proportional to the intensity of the corresponding pixel in the original from the database) and their raw images were recorded by the EM-CCD camera at an out-of-focus plane. These 10,000 ground-truth phase images and their corresponding raw intensity images were split into a training set of 9,500 images, a validation set of 450 images and a test set of 50 images. The choice of ImageNet [63] is reasonable, since the low-frequency dominance in its spatial PSD is representative of the broader classes of objects of interest, and therefore, we anticipate our results will generalize well in practical applications.

The experiments were carried out using the apparatus described in Figure 3. The light source was a continuous wave Helium-Neon gas laser at 632.8 nm. The laser beam first passed through a variable neutral density filter (VND) that served the purpose of adjusting the photon flux. The beam was then spatially filtered and expanded into a 18 mm diameter collimated pencil and sent onto a transmission spatial light modulator (SLM) with \( 256 \times 256 \) pixels, each of size \( 36 \times 36 \) \( \mu m \). Phase objects were projected on the SLM and imaged by a telescope (4F system) consisting of lenses L1 (focal length 230 mm) and L2 (100 mm). The \( 2.3 \times \) reduction factor in the 4F system was designed to reduce the spatial extent of the defocused raw image to approximately fit the size of the camera. An aperture was placed in the Fourier plane to suppress higher diffraction orders due to the periodicity of the SLM pixels. The raw intensity images were captured by a Q-Imaging EM-CCD camera with 1004 \( \times \) 1002 square-shaped pixels of size \( 8 \times 8 \) \( \mu m \) placed at a distance \( z = 400 \) mm from the image plane of the 4F system. Additional details about the implementation of the optical apparatus and its numerical simulation with digital Fresnel transforms are in Supplementary Material.

The photon flux is quantified as the number of photons \( p \) received by each pixel on average for an unmodulated beam, i.e. with no phase modulation driving the SLM. During an initial calibration procedure, for different positions of the VND filter, the photon level is measured using a calibrated silicon photodetector placed at the position of the camera. The quoted photon count \( p \) is also corrected for the quantum efficiency of the CCD (60\% at \( \lambda = 632.8 \) nm), meaning that we refer to the number of photons actually detected and not the incident number of...
Here, we report results for two levels of photon flux $p = 9.8 \pm 5\%$ and $p = 1.1 \pm 5\%$, respectively quoted in the text as “10” and “1” photons. The data acquisition, training and testing procedures of the entire LS-DNN architecture were repeated separately for each value of $p$.

**B. Reconstructions and Analysis**

Figure 4 shows the reconstructions obtained by the LS-DNN method ($q = 0.5$) and its components at fluxes $p = 1$ photon and 10 photons per pixel, as defined immediately above. As expected, the reconstructions $\hat{f}_{LF}$ by DNN-L have good fidelity at low spatial frequencies but lose fine details, as in [39]; whereas the reconstructions $\hat{f}_{HF}$ by DNN-H look like high-pass filtered versions of the true objects with some additional high-frequency artifacts due to the noise. The reconstructions $\hat{f}$ by DNN-S preserve detail at both low and high frequencies, while significantly attenuating the artifacts. The improvement of $\hat{f}$ over $\hat{f}_{LF}$ is more pronounced under severe noise, i.e. in the $p = 1$ photon/pixel case. More examples of reconstructions (obtained with $q = 0.5$) for the noisier case ($p = 1$) are given in the Supplementary Material.

In Figures 5 and 6, we compare reconstructions by LS-DNN with different values of the pre-filtering parameter $q$ for $p = 1$ photon and $p = 10$ photons per pixel, respectively. The most detail at high frequencies in the DNN-S output is preserved in the range $0.3 \lesssim q \lesssim 0.7$. At lower values of $q$, the quality of reconstructions by DNN-S does not noticeably exceed that of DNN-L. This is expected, since in the limit $q = 0$, training DNN-H becomes identical to DNN-L. On the other hand, for values $q \gtrsim 0.7$, the DNN-H output is dominated by high-frequency artifacts and again the quality of DNN-S reconstructions regresses to that of DNN-L, since the high-frequency channel is no longer contributing. These observations are valid for both values of $p$, and even stronger for the most severely noise-limited case $p = 1$.

Similar trends are evident according to various quantitative metrics averaged over the entire set of test examples compared to the true phase signals $f$, summarized in Table 1. For comparisons we used the PCC, defined as in (16) but without the minus sign; peak signal-to-noise ratio (PSNR) [87]; and structural similarity index metric (SSIM) [84, 85]. As we noted in Section 3.A, DNN-H is trained with spectrally pre-filtered version of the true object $f$ so a quantitative comparison of its output with the ground truth does not make sense.

It is noteworthy that in both visualization and quantitative comparisons of Figures Figures 5-6 and Table 1, respectively, the performance of DNN-S remains approximately the same within the range $0.3 \lesssim q \lesssim 0.7$. This is desirable as it suggests that one need not prefilter exactly with the inverse of the PSD power law. This further suggests that for datasets that do not represent natural images and may obey power laws different than (10), not knowing the exact value of $q$ may not be catastrophic. We have not tested this hypothesis exhaustively, as it is beyond the scope of this paper.

In Table 2 and in Supplementary Material, we also analyze the case of a bigger DNN (denoted DNN-L-3) with computational capacity equal to the sum of DNN-L, H and S together, but trained with un-filtered examples, and show that DNN-L-3 cannot achieve reconstructions of equal quality. Therefore, the improvements over [39] resulted from the training procedure followed in the LS-DNN method and not simply by brute force due to the use of larger computational capacity.

To further study the behavior of the LS components in the low and high spatial frequency bands, we studied the reconstructions in the Fourier domain. Figure 7 shows the spectra (2D Fourier transforms) of two randomly selected test examples. Figure 8 and Figure 5 in Supplementary Material show normalized diagonal cross-sections of the PSD averaged over all the 50 test images, for $p = 1$ and 10 photons per pixel, respectively. These plots illustrate that DNN-L and DNN-H’s outputs are depeleted at the high and low frequencies, respectively, with the losses being more severe in the noisy case $p = 1$; whereas DNN-S’s output mostly recovers the frequency content at both bands, albeit still with some minor loss at high frequencies.

Lastly, we experimentally characterized the spatial resolution of the LS-DNN reconstructions, i.e. the ability of DNN-S to resolve two pixels at nearby locations having phase delay higher than the rest of the signal. Similar analyses were carried out in [38, 65], where the methodology was also described in detail. In the work presented here, we carried out the analysis under ample illumination, i.e. not under strong Poisson statistics. We...
Fig. 4. Reconstructions by LS-DNN (top: 1 photon/pixel/frame, bottom: 10 photons/pixel/frame); from left to right: Approximant (the input to the LS-DNN system), DNN-L reconstruction [39], DNN-H reconstruction ($q = 0.5$), DNN-S reconstruction, ground truth.

made that choice because spatial resolution under highly noisy conditions becomes non-trivially coupled to the noise statistics, and a complete investigation would have been outside the scope of the present investigation. The results are shown in section 6 in Supplementary Material.

5. CONCLUDING REMARKS
The LS-DNN reconstruction scheme [72] for quantitative phase retrieval has been shown to be resilient to highly noisy raw intensity inputs while preserving high spatial frequency detail better than [39]. The reconstructions are also quite robust to variations of the pre-filtering power law $q$ around the value $\approx 1/2$ following from natural image statistics. Beyond the scope of the work reported here, further improvements may be obtained through modifying the architecture of the DNNs used to process and recombine the two spatial frequency bands. It is also possible that for highly restricted datasets, $q$ needs to be investigated further. However, our observations for natural images suggests that the LS-DNN approach is relatively insensitive to $q$ under fairly wide conditions.

Another obvious alternative strategy is to split the signals into more than two bands, then process and recombine these multiple bands with a synthesizer DNN, according to the LS scheme. While we did not investigate this approach in detail here, we expect it to present a trade-off between the improvements and the complexity of having to train multiple neural networks implying the need for more examples and the danger of poor generalization.

ACKNOWLEDGMENTS
This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) grant No. FA8650-17-C-9113 and by the SenseTime company. I. Kang was supported in part by KFAS (Korea Foundation for Advanced Studies) scholarship. We are grateful to Kwabena Arthur for useful discussions and critique of the manuscript.
Fig. 5. Comparisons of LS-DNN reconstructions under different $q'$s for $p = 1$ photon/pixel. Rows from top to bottom: DNN-L output; DNN-H output under different $q'$s; DNN-S under different $q'$s; 1D cross-section (along the dashed line in the row below) of DNN-S output under different $q'$s; ground truth.

REFERENCES

1. P. Marquet, B. Rappaz, P. J. Magistretti, E. Cuche, Y. Emery, T. Colomb, and C. Depeursinge, “Digital holographic microscopy: a noninvasive contrast imaging technique allowing quantitative visualization of living cells with subwavelength axial accuracy,” Opt. Lett. 30, 468–470 (2005).

2. G. Popescu, T. Ikeda, R. R. Dasari, and M. S. Feld, “Diffraction phase microscopy for quantifying cell structure and dynamics,” Opt. letters 31, 775–777 (2006).

3. S. C. Mayo, T. J. Davis, T. E. Gureyev, P. R. Miller, D. Paganin, A. Pogany, A. W. Stevenson, and S. W. Wilkins, “X-ray phase-contrast microscopy and microtomography,” Opt. Express 11, 2289–2302 (2003).

4. F. Pfeiffer, T. Weitkamp, O. Bunk, and C. David, “Phase retrieval and differential phase-contrast imaging with low-brilliance X-ray sources,” Nat. Phys. 2, 258–261 (2006).

5. A. Pan, Ling Xu, J. C. Petruccelli, R. Gupta, B. Singh, and G. Barbastathis, “Contrast enhancement in x-ray phase contrast tomography,” Opt. express 22, 18020–18026 (2014).

6. M. Holler, M. Guizar-Sicairos, E. H. R. Tsai, R. Dinapoli, O. Bunk, J. Raabe, and G. Aeppli, “High-resolution non-destructive three-dimensional analysis of integrated cir-
Fig. 6. Comparisons of LS-DNN reconstructions under different $q'$s for $p = 10$ photon/ pixel. Rows from top to bottom: DNN-L output; DNN-H output under different $q'$s; DNN-S under different $q'$s; 1D cross-section (along the dashed line in the row below) of DNN-S output under different $q'$s; ground truth.
Table 1. Quantitative comparison of reconstructions by Approximant, DNN-L and DNN-S. Each entry takes the form of ‘average ± standard deviation’.

|                  | Average PCC ± std.dev | Average PSNR ± std.dev (dB) | Average SSIM ± std.dev |
|------------------|------------------------|-----------------------------|------------------------|
|                  | p = 1                  | p = 10                      | p = 1                  | p = 10 |
| Approximant $\hat{f}$ | 0.149 ± 0.067           | 0.181 ± 0.082               | 8.082 ± 3.979          | 8.098 ± 3.986 | 0.220 ± 0.105 | 0.222 ± 0.106 |
| DNN-L output $\hat{f}^{LF} $ | 0.808 ± 0.099           | 0.880 ± 0.059               | 16.207 ± 2.466         | 18.299 ± 2.536 | 0.875 ± 0.071 | 0.925 ± 0.039 |
| DNN-S output $\hat{f} (q = 0.1) $ | 0.848 ± 0.075           | 0.892 ± 0.053               | 17.511 ± 2.317         | 18.426 ± 2.321 | 0.905 ± 0.054 | 0.929 ± 0.048 |
| DNN-S output $\hat{f} (q = 0.2) $ | 0.858 ± 0.074           | 0.898 ± 0.050               | 17.758 ± 2.523         | 18.469 ± 2.469 | 0.903 ± 0.056 | 0.926 ± 0.036 |
| DNN-S output $\hat{f} (q = 0.3) $ | 0.859 ± 0.071           | 0.896 ± 0.050               | 17.769 ± 2.420         | 18.487 ± 2.313 | 0.912 ± 0.048 | 0.930 ± 0.043 |
| DNN-S output $\hat{f} (q = 0.4) $ | 0.865 ± 0.067           | 0.897 ± 0.050               | 18.116 ± 2.319         | 18.634 ± 2.380 | 0.915 ± 0.061 | 0.934 ± 0.034 |
| DNN-S output $\hat{f} (q = 0.5) $ | 0.866 ± 0.069           | 0.899 ± 0.048               | 17.940 ± 2.281         | 18.902 ± 2.443 | 0.915 ± 0.051 | 0.937 ± 0.034 |
| DNN-S output $\hat{f} (q = 0.6) $ | 0.860 ± 0.071           | 0.898 ± 0.050               | 17.880 ± 2.455         | 18.940 ± 2.320 | 0.915 ± 0.062 | 0.935 ± 0.038 |
| DNN-S output $\hat{f} (q = 0.7) $ | 0.859 ± 0.088           | 0.892 ± 0.051               | 17.827 ± 2.140         | 18.487 ± 2.311 | 0.911 ± 0.070 | 0.927 ± 0.047 |
| DNN-S output $\hat{f} (q = 0.8) $ | 0.845 ± 0.081           | 0.887 ± 0.050               | 17.375 ± 2.221         | 18.445 ± 2.470 | 0.902 ± 0.059 | 0.934 ± 0.041 |
| DNN-S output $\hat{f} (q = 1) $ | 0.825 ± 0.111           | 0.883 ± 0.052               | 16.991 ± 2.008         | 18.484 ± 2.363 | 0.894 ± 0.075 | 0.934 ± 0.042 |
| DNN-S output $\hat{f} (q = 2) $ | 0.819 ± 0.077           | 0.882 ± 0.055               | 16.820 ± 2.254         | 18.391 ± 2.600 | 0.893 ± 0.058 | 0.933 ± 0.037 |

Table 2. Quantitative comparison of reconstructions by Approximant, DNN-L-3 and DNN-S (for $q = 0.5$). Each entry takes the form of ‘average ± standard deviation’.

|                  | Average PCC ± std.dev | Average PSNR ± std.dev (dB) | Average SSIM ± std.dev |
|------------------|------------------------|-----------------------------|------------------------|
|                  | p = 1                  | p = 10                      | p = 1                  | p = 10 |
| Approximant $\hat{f}$ | 0.149 ± 0.067           | 0.181 ± 0.082               | 8.082 ± 3.979          | 8.098 ± 3.986 | 0.220 ± 0.105 | 0.222 ± 0.106 |
| DNN-L output $\hat{f}^{LF}$ | 0.808 ± 0.099           | 0.880 ± 0.059               | 16.207 ± 2.466         | 18.299 ± 2.536 | 0.875 ± 0.071 | 0.925 ± 0.039 |
| DNN-L-3 output $\hat{f}^{LF}$ | 0.811 ± 0.125           | 0.880 ± 0.117               | 16.209 ± 2.522         | 18.268 ± 2.030 | 0.835 ± 0.092 | 0.926 ± 0.073 |
| DNN-S output $\hat{f} (q = 0.5) $ | 0.866 ± 0.069           | 0.899 ± 0.048               | 17.940 ± 2.281         | 18.902 ± 2.443 | 0.915 ± 0.051 | 0.937 ± 0.034 |

16. J. R. Fienup, “Phase retrieval algorithms – a comparison,” Appl. Opt. 21, 2758–2769 (1982).
17. H. H. Bauschke, P. L. Combettes, and D. R. Luke, “Phase retrieval, error reduction algorithm, and fienup variants: a view from convex optimization,” JOSA A 19, 1334–1345 (2002).
18. R. W. Gerchberg, “The lock problem in the Gerchberg-Saxton algorithm for phase retrieval,” Optik 74, 91–93 (1986).
19. J. Fienup and C. Wackerman, “Phase-retrieval stagnation problems and solutions,” JOSA A 3, 1897–1907 (1986).
20. E. Candès and T. Tao, “Decoding by linear programming,” IEEE Trans. Inf. Theory 51, 4203–4215 (2005).
21. E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” IEEE Transactions on information theory 52, 489–509 (2006).
22. D. L. Donoho, “Compressed sensing,” IEEE Transactions on information theory 52, 1289–1306 (2006).
23. E. Candès, J. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” Comm. Pure Appl. Math. 59, 1207–1223 (2006).
24. Y. C. Eldar and G. Kutyniok, Compressed Sensing: Theory and Applications (Cambridge University Press, 2012).
25. M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” IEEE Transactions on Image processing 15, 3736–3745 (2006).
26. M. Aharon, M. Elad, and A. Bruckstein, “K-Svd: An algorithm for designing overcomplete dictionaries for sparse representation,” IEEE Transactions on signal processing 54, 4311–4321 (2006).
27. R. Rubinstein, A. M. Bruckstein, and M. Elad, “Dictionaries for sparse representation modeling,” Proc. IEEE 98, 1045–1057 (2010).
28. Changlong Bao, Hui Ji, Yuhui Quan, and Zuowei Shen, “Dictionary learning and sparse coding: algorithms and convergence analysis,” IEEE Trans. Patt. Anal. Mach. Intel. 38, 1356–1369 (2016).
29. K. Gregor and Y. LeCun, “Learning fast approximations of sparse coding,” in Proceedings of the 27th International Conference on International Conference on Machine Learning, (Omnipress, 2010), pp. 399–406.
30. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, Bing Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Nets,” in Neural Information Processing Systems (NIPS), , vol. 27 (2014).
31. M. Mardani, H. Monajemi, V. Papyan, S. Vasanawala, D. L.
Fig. 7. Fourier spectra of two test examples and their reconstructions from the components of the LS scheme.

Donoho, and J. M. Pauly, “Recurrent generative adversarial networks for proximal learning and automated compressive image recovery,” CoRR. (2017).

32. C.-Y. Yang, C. Ma, and M.-H. Yang, “Single-image super-resolution: A benchmark,” in European Conference on Computer Vision, (Springer, 2014), pp. 372–386.

33. Chao Dong, Chen Loy, Kaiming He, and Xiaou Tang, “Learning a deep convolutional neural network for image super-resolution,” in European Conference on Computer Vision (ECCV) / Lecture Notes on Computer Science Part IV, vol. 8692 (2014), pp. 184–199.

34. Chao Dong, Chen Loy, Kaiming He, and Xiaou Tang, “Image super-resolution using deep convolutional networks,” Pami. 38, 295–307 (2015).

35. Kyong Hwan 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).

36. M. T. McCann, Kyong Hwan Jin, and M. Unser, “Convolutional neural networks for inverse problems in imaging,” IEEE Sig. Process. Mag. pp. 85–95 (2017).

37. A. Sinha, Justin Lee, Shuai Li, and G. Barbastathis, “Lensless computational imaging through deep learning,” Optica. 4, 1117–1125 (2017).

38. Shuai Li and G. Barbastathis, “Spectral pre-modulation of training examples enhances the spatial resolution of the phase extraction neural network (PhENN),” Opt. Express 26, 29340–29352 (2018).

39. A. Goy, K. Arthur, Shuai Li, and G. Barbastathis, “Low photon count phase retrieval using deep learning,” Phys. Rev. Lett. 121, 243902 (2018).

40. Y. Rivenson, Yibo Zhang, H. Günaydin, Da Teng, and A. Ozcan, “Phase recovery and holographic image reconstruction using deep learning in neural networks,” Light. Sci. 7, 17141 (2018).

41. Y. Wu, Y. Rivenson, Y. Zhang, Z. Wei, H. Günaydin, X. Lin, and A. Ozcan, “Extended depth-of-field in holographic imaging using deep-learning-based autofocus and phase recovery,” Optica. 5, 704–710 (2018).

42. T. Nguyen, Y. Xue, Y. Li, L. Tian, and G. Nehmetallah, “Deep learning approach for fourier ptychography microscopy,” Opt. express 26, 26470–26484 (2018).

43. Y. Xue, S. Cheng, Y. Li, and L. Tian, “Reliable deep-learning-based phase imaging with uncertainty quantification,” Op-
Fig. 8. 1D diagonal cross-sections of the average 2D power spectral density (PSD) of 50 test images for $p = 1$ photon per pixel. The case of $p = 10$ photons per pixel is in the Supplementary Material.
68. A. Van der Schaaf and J. H. van Hateren, “Modelling the power spectra of natural images: statistics and information,” Vis. Res. 36, 2759–2770 (1996).

69. B. A. Olshausen and D. J. Field, “Sparse coding with an overcomplete basis set: a strategy employed by V1?” Vis. Res. 37, 3311–3325 (1997).

70. M. S. Lewicki and B. A. Olshausen, “A probabilistic framework for the adaptation and comparison of image codes,” J. Opt. Soc. A 16, 1587–1601 (1999).

71. M. S. Lewicki and B. A. Olshausen, “A probabilistic framework for the adaptation and comparison of image codes,” J. Opt. Soc. A 16, 1587–1601 (1999).

72. M. S. Lewicki and B. A. Olshausen, “A probabilistic framework for the adaptation and comparison of image codes,” J. Opt. Soc. A 16, 1587–1601 (1999).

73. O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in Medical Image Computing and Computer-Assisted Intervention (MICCAI), vol. 9351 of LNCS (Springer, 2015), pp. 234–241. (available on arXiv:1505.04597 [cs.CV]).

74. L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in Proceedings of COMPSTAT’2010, (Springer, 2010), pp. 177–186.

75. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” CoRR abs/1412.6980 (2015).

76. G. E. Hinton, “Learning translation invariant recognition in massively parallel networks,” in Proceedings PARLE Conference on Parallel Architectures and Languages Europe, (Springer-Verlag, 1987), pp. 1–13.

77. C. M. Bishop, Pattern recognition and machine learning (Springer, 2006).

78. J. Johnson, A. Alahi, and Li Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in European Conference on Computer Vision (ECCV) / Lecture Notes on Computer Science, vol. 9906 B. Leide, J. Matas, N. Sebe, and M. Welling, eds. (2016), pp. 694–711.

79. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning (MIT Press, 2016).

80. C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Zehan Wang, and Wenshe Shi, “Photo-realistic single image super-resolution using a Generative Adversarial Network,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017), pp. 4681–4690.

81. K. Pearson, “Contributions to the mathematical theory of evolution. note on reproductive selection,” Proc. Royal Soc. 59, 300–305 (1896).

82. J. L. Rodgers and W. A. Nicewander, “Thirteen ways to look at the correlation coefficient,” The Am. Stat. 42, 59–66 (1988).

83. S. Li, G. Barbastathis, and A. Goy, “Analysis of phase-extraction neural network (phen) performance for lensless quantitative phase imaging,” in Quantitative Phase Imaging V, vol. 10887 (International Society for Optics and Photonics, 2019), p. 108870F.

84. Z. Wang, E. P. Simoncelli, and A. C. Bovik, “Multiscale structural similarity for image quality assessment,” in The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, vol. 2 (Ieee, 2003), pp. 1398–1402.

85. Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” ieeep. 13, 600–612 (2004).