Probing shallower: perceptual loss trained Phase Extraction Neural Network (PLT-PhENN) for artifact-free reconstruction at low photon budget

Citation: Deng, Mo et al. "Probing shallower: perceptual loss trained Phase Extraction Neural Network (PLT-PhENN) for artifact-free reconstruction at low photon budget." Optics Express 28, 2 (January 2020): 2511-2535 © 2020 Optical Society of America

As Published: http://dx.doi.org/10.1364/oe.381301

Publisher: The Optical Society

Persistent URL: https://hdl.handle.net/1721.1/126326

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Optics EXPRESS

Probing shallower: perceptual loss trained Phase Extraction Neural Network (PLT-PhENN) for artifact-free reconstruction at low photon budget

MO DENG,[1,6,*] ALEXANDRE GOY,[2,4,6] SHUAI LI,[2,5] KWABENA ARTHUR,[2] AND GEORGE BARBASTATHIS[2,3]

[1] *Department of Electrical Engineering and Computer Science, MIT, 77 Massachusetts Ave, Cambridge, MA 02139, USA*
[2] *Department of Mechanical Engineering, MIT, 77 Massachusetts Ave, Cambridge, MA 02139, USA*
[3] *Singapore-MIT Alliance for Research and Technology (SMART) Centre, Singapore 117543, Singapore*
[4] *Current address: Omnisens SA, Riond Bosson 3, 1110 Morges, VD, Switzerland*
[5] *Current address: Sensebrain Technology Limited LLC, 2550 N 1st Street, Suite 300, San Jose, CA 95131, USA*
[6] *Equal contribution*
[*] *modeng@mit.edu*

Abstract: Deep neural networks (DNNs) are efficient solvers for ill-posed problems and have been shown to outperform classical optimization techniques in several computational imaging problems. In supervised mode, DNNs are trained by minimizing a measure of the difference between their actual output and their desired output; the choice of measure, referred to as "loss function," severely impacts performance and generalization ability. In a recent paper [A. Goy *et al.*, Phys. Rev. Lett. 121(24), 243902 (2018)], we showed that DNNs trained with the negative Pearson correlation coefficient (NPCC) as the loss function are particularly fit for photon-starved phase-retrieval problems, though the reconstructions are manifestly deficient at high spatial frequencies. In this paper, we show that reconstructions by DNNs trained with default feature loss (defined at VGG layer ReLU-22) contain more fine details; however, grid-like artifacts appear and are enhanced as photon counts become very low. Two additional key findings related to these artifacts are presented here. First, the frequency signature of the artifacts depends on the VGG's inner layer that perceptual loss is defined upon, halving with each MaxPooling2D layer deeper in the VGG. Second, VGG ReLU-12 outperforms all other layers as the defining layer for the perceptual loss.

1. Introduction

In the last few years, the importance of deep learning in the field of computational imaging has been rapidly growing [1]. Deep neural networks (DNNs) [2] are used in a variety of tasks such as denoising [3], super-resolution imaging [4–7], imaging through scattering media [8],and optical and X-ray tomography [9–13]. The success of DNNs comes from their versatility and execution speed once they have been trained. They have proven particularly suitable for underdetermined and ill-posed problems, which earlier had been classically solved using iterative optimization methods involving a regularization scheme in the functional. In classical methods, the regularizer's role is to incorporate prior knowledge about the class of objects expected to appear in the particular scenario under consideration. The regularizer has to be designed to favor solutions that match properties of the object known to be true: for example, smoothness at continuous surfaces, sharpness at edges, positivity, real-valuedness, geometrical support, etc. The task of designing the optimal regularizer is difficult for two reasons: first, one may

#381301 https://doi.org/10.1364/OE.381301
Journal © 2020 Received 31 Oct 2019; revised 24 Dec 2019; accepted 29 Dec 2019; published 17 Jan 2020

not know exactly which object features really matter for the problem under consideration; and second, even if these features were known, designing the proper regularization operator that favors them might not always be trivial. Alternatively, the regularizer can be *learnt* in the form of dictionaries [14,15] that exploit known examples to generate a set of basis functions where sparse representations of valid objects are favored. DNNs offer a similar advantage as dictionaries in that they learn the prior from the data, while also being open to discover priors different than sparsity.

One computational imaging problem of significant importance, also chosen as testing ground for this paper, is phase retrieval. Generally, the problem is stated as retrieving a complex function from the modulus of its Fourier or Fresnel transform. A popular variation is to assume the unknown function to be pure phase, *i.e.* the object to be absorption-free, because this assumption is often valid for biological cells. Traditionally, various approaches have been applied to phase retrieval, including interferometric/holographic [16,17], iterative [18–21] , ptychographic [22,23], and transport-based [24,25]. Recently, phase retrieval was among the first computational imaging problems where DNNs were successfully used [26–31]. The original method in [26] was dubbed “Phase Extraction Neural Network” (PhENN) and it is the backbone of the architecture investigated here as well.

As in any other imaging problem, phase retrieval becomes increasingly difficult to solve as the photon budget available for the measurement is reduced. In a recent paper [32], we demonstrated a method for phase retrieval in extremely low light conditions that combines a DNN with a physics-inspired preprocessing step based on the Gerchberg-Saxton algorithm [18] on the Fresnel propagated field. The preprocessing step projects the measurement back to the object plane so as to obtain a first guess of the object, hereafter called the “Approximant.” Note that an exact projection is not possible, neither is convexity guaranteed [20,33]. In any case, since the intensity input is very noisy, the quality of the Approximant reconstruction is very poor. The role of the subsequent DNN is to both denoise the Approximant and to correct the distortion left over by the approximate projection. For the noisiest raw inputs, this Approximant scheme is proven to be much more effective than the "End-to-End" scheme [26], where the intensity images are input directly to the DNN, thereby not taking advantage of known physics—in other words, in the End-to-End scheme the DNN carries the double burden of learning the prior *and* the physical model.

In both End-to-End and Approximant schemes, reconstructions display uneven fidelity across spatial frequency bands: the low frequency band tends to be reliable, while the high frequencies are suppressed, resulting in an over-smoothening effect [8,26,32]. In [34], Li *et al* attributed this uneven fidelity to the uneven treatment of spatial frequencies during DNN training. High spatial frequencies are under-represented in the training examples and therefore less likely to survive the nonlinearity of the training process. The authors proposed to mitigate this by *ad hoc* spectral pre-filtering to artificially amplify the high spatial frequencies in the training examples. Though spatial resolution was improved significantly, spectral pre-filtering violated the true prior of the training examples and thus artifacts and distortions emerged. Subsequently, we proposed the Learning to Synthesize by DNN (LS-DNN) scheme [35], which takes the more principled approach of splitting and processing low and high frequencies separately by two DNNs, followed by a synthesizing DNN trained specifically to recombine the two bands reliably into a reconstruction with even performance across the entire spectrum. We also found the LS-DNN method to be resilient to noise [35], yielding reconstructions with much finer detail than [32].

In this work, we take an altogether different approach to attack the uneven spatial frequency problem. We use what is essentially a cognitive metric, the perceptual loss function [7], to train PhENN. In perceptual loss training, as we describe in detail in the next Section 2, PhENN is trained based on the distance of internal representations (feature maps) from a *pre-trained* natural image classification network, e.g. VGG [36]. The intuitive argument for this approach is as follows: in the primate visual system, the hierarchy of classification circuits is concordant with the power-spectral density of natural images [37,38]; that is, low and high spatial frequencies are all given their fair share of processing. It is not clear whether something similar occurs in artificial image classification, e.g. VGG, even though the present paper presents some evidence corroborating in favor of this hypothesis. Arguably, by forcing the reconstructions produced by PhENN to match the feature maps of the classification network VGG, we are causing the distribution of spatial frequencies in PhENN's reconstructions to match that of the original natural objects. Thus, we restore the balance in the treatment of high and low frequencies. Our results, Section 3, show that this hypothesis is true, albeit also producing certain periodic artifacts, whose nature and mitigation we address in Section 4. The key outcome is that probing the VGG at ReLU-12, shallower than ReLU-22 recommended in [7], seems to offer the best compromise between noise-resilient and artifact-free reconstructions.

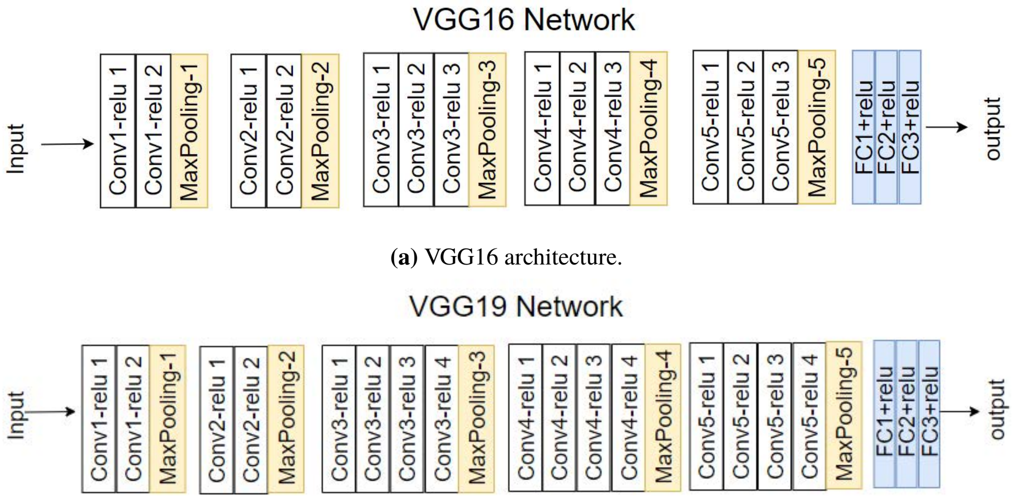
2. Training for inverse problems with VGG based perceptual loss

In [32], PhENN was trained by minimizing the negative Pearson correlation coefficient (NPCC) between the ground truth and the DNN output. The NPCC was earlier proven to work better than the pixel-wise loss functions, e.g. Mean Square Error (MSE), Mean Absolute Error (MAE), as well as the frequently used similarity metric SSIM [39,40] in retrieving fine features [8,41] of the objects. This is because pixel-wise losses (*e.g.* MSE, MAE) generally encourage finding pixel-wise averages of plausible solutions which tend to be oversmooth [39,40,42].

Perceptual loss, based instead on high-level image feature representations extracted from CNNs pre-trained for image-classification tasks, can be used as the loss function to generate images with good visual quality. It was first applied to various image processing tasks, including feature inversion [43], feature visualization [44,45], and texture synthesis and style transfer [46]. Later, in [7], Johnson *et al* first combined the advantages of the feed-forward neural networks and perceptual loss for style transfer and super-resolution. Subsequently, perceptual loss has been applied to many image-formation applications, including [47,48], *etc*. However, to our knowledge, perceptual loss has not yet been successfully applied to phase retrieval, neither has it been applied to inverse problems under extremely low light condition in general, as this paper is concerned.

The VGG network [36], whose versions include VGG16 and VGG19, is a class of deep convolutional neural networks (CNN) that has been highly successful with classification tasks on ImageNet [49]. As in Figs. 1(a) and 1(b), respectively, VGG16 and VGG19 each consists of 5 Convolutional Blocks (CBs), followed by 3 Dense layers. In each CB, there are a few 2D convolutional layers, with Rectified Linear Units (ReLU) as the nonlinearity, followed by a MaxPooling2D with factor 2. Each CB, by its MaxPooling2D layer specifically, reduces the size of each feature map by 2×2. Each subsequent CB doubles the depth of feature maps (the number of feature maps) and within each CB, the number of feature maps remains constant. Each convolutional layer has kernel size 3×3. Subsequent to the CBs, three Dense layers with ReLU as the nonlinear activations, sequentially map the feature maps to the final output, which is compared with the ground truth labels for classification.

As the signals from input objects propagate through the VGG network layers, features relevant for classification become progressively identified as components in sparse representations of the objects. Therefore, given an inverse problem on ImageNet [49] (or objects similar to natural images), if the DNN is trained not by the conventional pixel-wise loss between the reconstructions and their corresponding ground truth examples, but instead by the loss of their corresponding feature maps at a certain layer, then it would generally encourage the reconstructions generated to be ones that are more likely to be correctly classified by the VGG. If we assume that human visual perception is fine tuned for recognition and classification tasks, then we can expect that a DNN trained with the perceptual loss function will produce images that are of a better visual quality as



(a) VGG16 architecture.

(b) VGG 19 architecture.

Fig. 1. VGG architecture.

perceived by a human observer. In fact, a recent study [50] discovered that similarity in deep features, including those generated from VGG networks, corresponds well to perceptual similarity and outperforms any known low-level quantitative metrics (*e.g.* PSNR, SSIM, etc.) These arguments justify the popularity of VGG-based perceptual loss methods for inverse problems. In a representative work [7], Johnson *et al* demonstrated that, in general, the distance on low-level (shallower) feature maps of VGG network corresponds well to disparity of content fidelity between input examples, while that on high-level (deeper) feature maps corresponds to style disparity. Stated differently, cognitive prior information learned from classifying pre-trained examples is transferred into the image transformation and image inversion problems, compensating for ill-posedness.

Loss based on VGG feature maps has been used both on its own [7], *i.e.* formed on the feature maps extracted at a particular layer of VGG network; and with the image-domain loss to form a mixed loss [51]. In the former case, hereafter referred to as the Feature Loss, layer ReLU-22 was commonly believed as the ideal choice, as it seemed to best compromise between visual quality and image-content accuracy. In the latter case, hereafter referred to as the Mixed Loss, with image-domain loss governing image-content fidelity, feature maps at deeper layers of pre-trained classification networks were believed to be ideal, as they supposedly compensate for style information that pixel-wise loss was incapable of reconstructing.

Let n_{feat} denote the number of feature maps and $N_x \times N_y$ the size of each feature map at the respective layer of the VGG network [36]. Let also ReLU-ij denote the j^{th} convolutional layer in the i^{th} CB. Then the Feature Loss between the ground truth f and the reconstruction $\hat{f}$ at layer ReLU-ij is

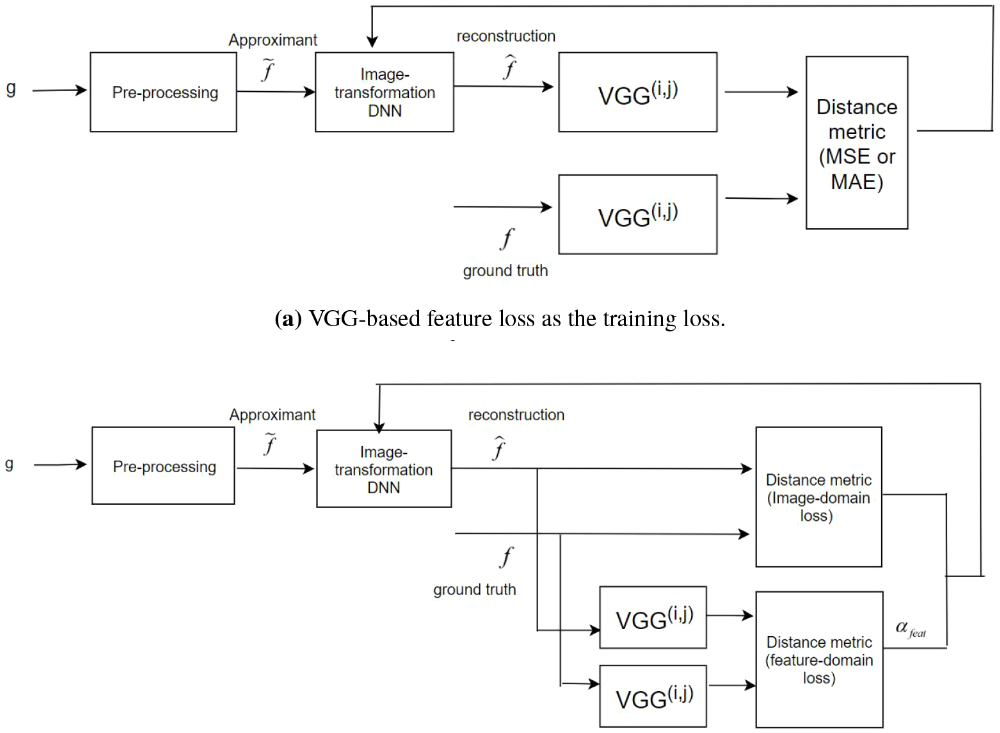
$$\mathscr{L}_{\mathrm{feat}}^{(i,j)}(f,\hat{f}) = \frac{1}{n_{\mathrm{feat}} N_x N_y} \sum_{k=1}^{n_{\mathrm{feat}}} \left\| \mathrm{VGG}_k^{(i,j)}(f) - \mathrm{VGG}_k^{(i,j)}(\hat{f}) \right\|_2^2, \tag{1}$$

where $\|.\|_2$ denotes the L^2 norm, and $\mathrm{VGG}_k^{(i,j)}(f)$ denotes the k^{th} feature map generated when passing image f up to layer ReLU-ij of VGG.

For the Mixed Loss scheme, left α_{feat} denote the parameter controlling relative strength between image-domain loss and feature loss. DNN training minimizes

$$\begin{aligned}\mathscr{L}^{(i,j)}_{\text{mixed}}(f,\hat{f}) &= \mathscr{L}_{\text{image}}(f,\hat{f}) + \alpha_{\text{feat}}\mathscr{L}^{(i,j)}_{\text{feat}}(f,\hat{f}) \\ &= \left\|f-\hat{f}\right\|_2^2 + \alpha_{\text{feat}}\frac{1}{n_{\text{feat}}N_xN_y}\sum_{k=1}^{n_{\text{feat}}}\left\|\text{VGG}_k^{(i,j)}(f) - \text{VGG}_k^{(i,j)}(\hat{f})\right\|_2^2,\end{aligned} \tag{2}$$

Figures 2(a) and 2(b) show how VGG-based loss can be used to, either on its own or in collaboration with image-domain loss, train the image-transformation DNNs. The pre-processing step, which will be discussed in more detail in Section 3.1, is optional but can enable significantly better reconstructions in the most ill-posed cases [32].



(a) VGG-based feature loss as the training loss.

(b) VGG-based feature loss used as part of the mixed training loss.

Fig. 2. VGG-based feature loss as the training loss.

In both cases, to cope with the dynamic range of the pre-trained VGG network, images need to be normalized to [-1,1] before entering into the pre-trained VGG network. Also, a histogram matching step as post-processing is necessary to calibrate the scale of the reconstructions. We will discuss this later in Section 3.4.

Despite its advantages in terms of fidelity to spatial details in the reconstructions, the perceptual loss method has a significant defect. In [7], Johnson *et al* already noticed a "cross-hatch pattern" that appeared in Feature Loss based reconstructions. They attributed the reconstructions' inferiority in quantitative metrics, *e.g.* PSNR and SSIM, to these artifacts. Neither an interpretation nor a strategy to remove these artifacts has yet been provided, to our knowledge. One possible reason is that when no noise is present, as in [7], such artifacts are perceptible only under magnification, so it is tempting to ignore them.

Optics EXPRESS

When strong noise is present, as in our present investigation, the artifacts become much more pronounced. The artifacts display clear spatial periodicity, offsetting the visual quality advantage brought upon by the perceptual loss. Therefore, some effort into removing the artifacts is warranted. We now proceed with formulating the phase retrieval problem using perceptual loss, followed by investigation of the frequency signature (power spectral density) of the artifacts as function of the VGG ReLU-*ij* feature map used for training. The latter leads to a mitigation strategy for the artifacts.

3. PhENN implementation using perceptual loss

3.1. Phase retrieval as an inverse problem and computation of the approximant

Let

$$\psi_{\mathrm{obj}}(x,y) = t(x,y)\mathrm{e}^{if(x,y)}$$

be the complex transmittance of an optically thin object, with modulus response $t(x,y)$ and phase response $f(x,y)$. Here, we are only interested in weakly absorbing objects, *i.e.* $t(x,y) \approx 1$. Moreover, let $\psi_{\mathrm{inc}}(x,y)$ be the coherent incident illumination of wavelength λ on the object plane. Subject to the scalar and paraxial approximations, the noiseless intensity measurement on the detector plane located at distance z away, $g_0(x,y)$ can be written as:

$$g_0(x,y) = \left|\mathbf{F}_z\left[\psi_{\mathrm{inc}}(x,y)\psi_{\mathrm{obj}}(x,y)\right]\right|^2 \equiv H_0 f(x,y), \tag{3}$$

where $\mathbf{F}_z[\cdot]$ is the Fresnel propagation operator for distance z, and $H_0(\cdot)$ is the noiseless overall nonlinear forward operator. In this paper, we shall limit the choice of illumination to a normally incident plane wave, so that $\psi_{\mathrm{inc}}(x,y) = 1$. Therefore,

$$g_0(x,y) = |\mathbf{F}_z[\exp\{if(x,y)\}]|^2 = H_0 f(x,y), \tag{4}$$

The measurement is subject to a mixture of shot noise and readout noise, following Poisson and Gaussian statistics, respectively. Thus, to a good approximation, the measurement $g(x,y)$ captured on the detector is

$$g(x,y) = \mathscr{P}\left\{p\,\frac{H_0 f(x,y)}{\langle H_0 f\rangle}\right\} + \mathscr{N} \equiv Hf(x,y), \tag{5}$$

where $\mathscr{P}\{\theta\}$ denotes the Poisson random variable with mean θ, $\mathscr{N}$ the zero-mean Gaussian random variable with variance σ^2, and H the noisy forward operator. The term $\langle H_0 f\rangle$ is the mean of noiseless measurement, and is necessary to normalize the measurement so that p carries the physical meaning of average photon count per pixel.

In this work, we are particularly interested in phase retrieval under extremely low light conditions, *i.e.* very small values of p. This amounts to "inverting" the extremely ill-posed H to find the best $\hat{f}$ from g, so that (6) approximately holds. One way to achieve this is by minimizing a regularized functional as

$$\hat{f} = \underset{f}{\operatorname{argmin}}\left\{D\left(H_0 f, g\right) + \beta\Phi(f)\right\}. \tag{6}$$

Here, $\Phi(f)$ is the regularizer penalizing reconstructions that do not match the class of objects of interest, $D(H_0 f, g)$ is the data-fidelity term that matches the measurement to the forward operator for the assumed object, and β is the parameter expressing our belief on the relative importance of the measurement fitness versus the prior knowledge.

DNNs are desirable solvers of inverse problems such as Eq. (6) because they *learn* the prior from the training data leaving no need for sparsity notions to specify the applicable $\Phi(f)$.

Moreover, DNNs solve Eq. (6) fast, whereas proximal gradient methods used traditionally to minimize Eq. (6) are iterative and, thus, time consuming. The original End-to-End PhENN scheme [26] suffered the double burden of learning the prior *and* the forward operator, yet it proved to be adequate for the low-noise case. For noisy intensity data, Goy *et al* proposed the Approximant scheme [32], where the first iterate of the Gerchberg-Saxton algorithm [18]

$$\tilde{f} = \arg\left\{F^{-1}\left(\sqrt{g}\arg\left\{F\left(u_{\mathrm{inc}}\right)\right\}\right)\right\}, \tag{7}$$

is computed first, and is then used as input to the DNN to produce the final estimate

$$\hat{f} = \mathrm{DNN}\left(\tilde{f}\right). \tag{8}$$

In Eq. (7), $\tilde{f}$ is referred to as the Approximant, and ‘arg’ denotes the argument (phase) of the complex field. A comparative study [52] showed that using more iterations of Gerchberg-Saxton to produce better Approximants $\tilde{f}$ does not necessarily create a better estimate for the DNN reconstruction but does make the overall computation slower. Thus, the single iterate of Eq. (7) followed by Eq. (8) seems to be the best compromise.

3.2. Training PhENN with perceptual loss

Our use of PhENN [26,32] in this work was necessary for a fair comparison between our results and those in [32]. However, the approach described here is applicable to other architectures with the appropriate modifications. PhENN is essentially a deep U-net with residual connections [53]. Its input is the phase approximant $\tilde{f}$ (or the raw data f in the End-to-End scheme) and it generates the phase estimate $\hat{f}$ as its output.

In the perceptual training scheme, $\hat{f}$ is passed into the *pre-trained* VGG16 or VGG19 network up to a particular layer. The layer is either a ReLU-*ij* (see Section 2) or an i-Pooling, the MaxPooling2D layer in the i^{th} layer’s convolutional block. In our initial effort, Section 3.5, we used ReLU-22 as recommended by [7]; we subsequently expanded the investigation to other layers and we describe the results in Section 4. The feature maps are compared with those generated from the ground truth examples f at the same layer, either on its own (Feature Loss scheme, Fig. 2(a)), or collaborating with the image-domain loss (Mixed Loss scheme, Fig. 2(b)). Minimizing this loss optimizes the weights in the perceptual loss trained PhENN (PLT-PhENN). For testing, the phase estimate $\hat{f}$ is retrieved directly from the output of PLT-PhENN, so the VGG is not necessary; however, we still observe the feature maps of test objects through the VGG for analysis purposes.

Numerical computations are carried out on a Nvidia GTX1080 GPU using the open source Tensorflow platform [54], which allows us to reuse the pre-trained VGG network. The Adam optimizer [55] is used to train the neural network for 20 epochs, which takes approximately 2 hours. During the test stage, inference of the input (the Approximant in this case) corresponding to the test set is carried out on the trained neural network, taking only a few seconds.

3.3. Experimental apparatus and data acquisition

The examples used for training are extracted from the benchmark ImageNet [49] database. A set of 10,000 examples are randomly chosen and split into a training set of 9,500 examples, a validation set of 450 examples, and a test set of 50 examples. The latter is used to display the results and quantify performance.

The experimental apparatus, identical to that in [32], is depicted in Fig. 3 to which the optical components abbreviations refer. We use a coherent light source (continuous wave He-Ne laser at 632.8nm), which is first passed through a calibrated variable neutral density filter (VND). The light is then spatially filtered and collimated into a beam with a diameter of 18mm that serves as input illumination to the imaging setup. The incident illumination is sent through a transmissive spatial light modulator (SLM) with 256 × 256 36 × 36μm pixels. Phase objects are displayed on the SLM and imaged by a telescope (4F system) consisting of lenses L1 (focal length 230mm) and L2 (100mm). The 2.3× reduction factor in the 4F system is designed to match the size of the image with that of the camera. The image is spatially filtered in the Fourier plane in order to suppress higher diffraction orders by the SLM. The camera (Q-Imaging EM-CCD with 1004 × 1002 8×8μm pixels) is placed z = 400mm away from the image plane in order to introduce defocus—a necessary step for phase retrieval from pure phase objects. Subsequently, the raw images are fed to the computational pipeline described in Section 3.2.

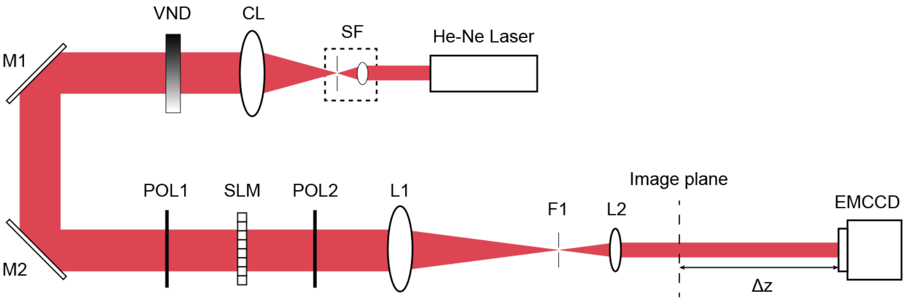


Fig. 3. Experimental Apparatus.

As in [32], the photon flux is quantified as the number of photons p incident on each detector pixel on average for an unmodulated beam, *i.e.* with no phase modulation displayed on 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 λ = 632.8nm). We refer to the number of photons actually detected and not the incident number of photons.

Here, we report results primarily for $p = 1.1 \pm 5\%$ (quoted as “1” photon/pixel), as it represents the extremely low light conditions that we are most interested in. Some results under higher photon levels, including $p = 10, 100, 1000$ are also presented for comparison.

3.4. Calibration of reconstructions

The reconstruction by the perceptual loss trained neural networks is generally a nonlinear function of the corresponding ideal object. For phase retrieval, we aim to produce a quantitatively accurate estimate of the phase. To that end, we perform a polynomial fit between the raw reconstructions and the ground truths from the validation set and use the optimized polynomial to calibrate reconstructions of the test objects. We empirically tested polynomials with degrees ranging from 1 to 10 and found that polynomials with degree 6 were the best for this fit, as any degree beyond 6 would not further reduce the validation error, only to increase the computational burden. We present results from this calibration step later in Section 3.5 and 4.

3.5. PLT-PhENN reconstructions with feature loss from ReLU-22

In Fig. 4, we show comparisons of phase retrieval results for $p = 1$ and $p = 10$, respectively, where PhENN is trained with the default feature loss at VGG16 ReLU-22, against those produced by the negative Pearson Correlation Coefficients (NPCC) trained PhENN, as in [32]. The feature-loss reconstructions display sharper details in general and, as can be seen in the scaled up images in Fig. 4, show particular details such as the vertical posts in the scene, that are completely blurred out in the NPCC reconstruction at 1-photon level. This observation is consistent with our intuition that the fine details (features) are necessary to semantically improve the accuracy of classification and, hence, are imposed upon PhENN by the perceptual loss training scheme. In Fig. 10, we show a similar comparison at four different photon levels ($p = 1, 10, 100, 1000$) and find the improvement in richness of recognizable details that the feature-loss introduces is most significant in the noisiest case $p = 1$.

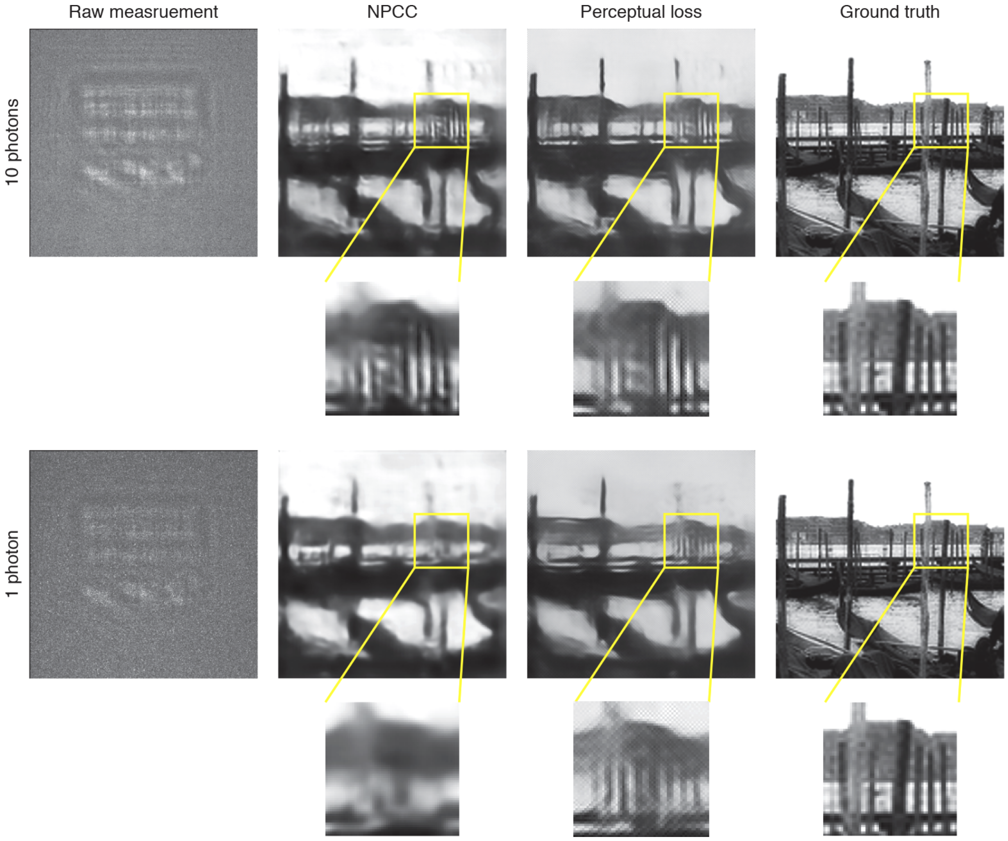


Fig. 4. Comparison of reconstructions from PhENN trained with perceptual loss *vs.* NPCC for 1 and 10-photon levels. The scaled up images show that some details are not rendered by the NPCC-trained PhENN whereas they become clearly identifiable with the perceptual loss function.

Also in Fig. 4, at low photon levels, the grid-like artifacts pointed out earlier as resulting from perceptual loss training (Section 2) become noticeable, both before and after the calibration step of Section 3.4. In ReLU-22 reconstructions, the artifacts are always centered at the same spatial frequency corresponding to a spatial period of 4 pixels, which we will refer to as the fundamental frequency ν_{f}.

In Fig. 5(a), we show an example severely affected by the artifact. Figure 5(b) shows the log-scale magnitude of same reconstruction's 2D Fourier transform, where the artifact is clearly visible at frequencies $(\nu_x, \nu_y) = (\pm\nu_{\mathrm{f}}, \pm\nu_{\mathrm{f}})$. In Fig. 5(c), we also compare the cross sections of the log-scale magnitude of the 2D Fourier Transform of the ground truth, the NPCC reconstruction, and the perceptual loss (feature loss) reconstruction, respectively, clearly indicating the same artifact at ν_{f}. In Fig. 6(a), we show the 2D power spectral density (PSD) of the entire set of test reconstructions and find this same frequency signature pronounced horizontally, vertically and diagonally.

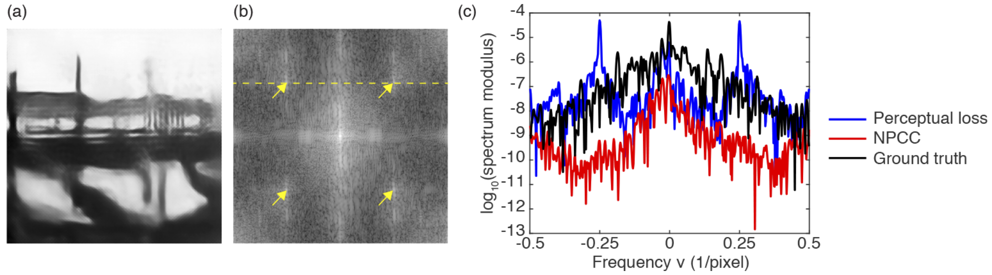


Fig. 5. (a) Reconstruction by the VGG16 ReLU-22 feature-loss trained DNN for the 1-photon level. (b) Log-scale magnitude of the 2D Fourier Transform of the reconstruction shown in (a). The artifact contributes in the modes indicated by the arrows. (c) Cross sections of the log-scale magnitude of the Fourier Transform of the perceptual loss reconstruction (blue), corresponding to image (b), ground truth (black) and the NPCC-trained DNN reconstruction (red).

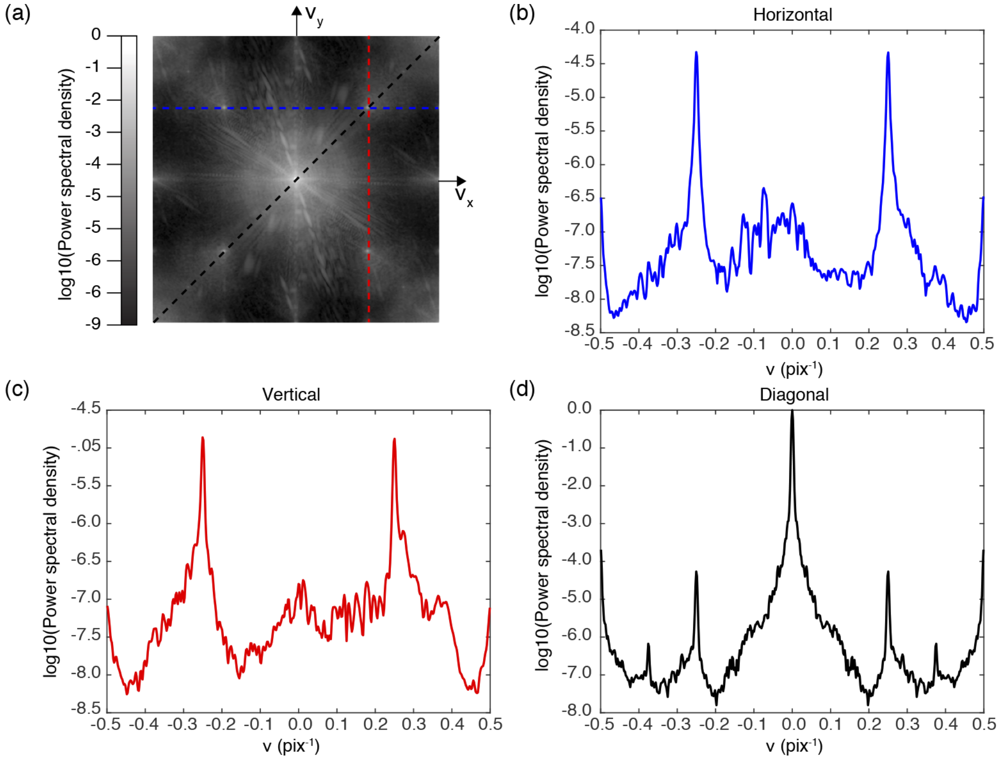


Fig. 6. (a) Log-magnitude of the power spectral density of the test set of reconstructions $\hat{f}$ clearly showing the signature of the artifact, which is perceived in the reconstructions as a prominent network of horizontal and vertical strips, e.g. Figs. 4 and 5. (b) Horizontal profile of (a). (c) Vertical profile of (a). (d) Diagonal profile of (a).

Figures 4–6 substantiate the increasing prevalence of that artifact at spatial frequency ν_f as the noise level worsens. Perhaps this is why earlier literature, concerning itself primarily with low-noise cases, paid only scarce attention to this issue. More auxiliary investigations in VGG16's (up to layer ReLU-22) particular behavior at the fundamental frequency ν_f can be found in Appendices B and C. We now turn to demonstrating that the artifact frequency depends on the

VGG layer where the feature maps are drawn from, and to using this discovery to mitigate the problem.

4. Probing shallower or deeper: finding an optimal ReLU for artifact-free, high-quality reconstructions

4.1. Frequency signature of the artifacts at different layers

The artifacts, at first glance, may be thought of as either content disparity or style disparity between the reconstructions and the ground truth. Such disparities are supposed to be handled by perceptual loss, but evidently the recipe malfunctions in this case. This motivated us to investigate whether the depth of the perceptual probe might influence image reconstruction quality. Alternatively, this strategy essentially alters the semantic depth of the content used to force PhENN to create realistic reconstructions: to the degree that the VGG bears any similarity to the primate visual cortex, shallow depths in VGG would correspond to low-level features such as elemental orientations and textures, whereas deeper layers would process concepts. Might finding just the right depth for perceptual probing be the key to PhENN's preserving fine detail without artifacts?

To test this hypothesis, we trained PhENN with perceptual loss drawn not from the standard ReLU-22 layer recommended by previous works, but from shallower and deeper layers. Representative results are shown in Figs. 7–9. First, we discuss the behavior of the grid-like artifacts as function of perceptual probing depth. We observe that the fundamental frequency of the artifacts in the reconstructions halves with each encounter with the MaxPooling2D layer going deeper into the network; however, the fundamental frequency of the artifacts remains constant until encountering the next MaxPooling2D layer, even though the artifacts could intensify as the defining layer progresses deeper. This is starkly illustrated in Fig. 9, where all PSDs in each of row (ii) to row(v) have a common fundamental frequency and this fundamental frequency is half of that from the previous row.

Next, we discuss overall reconstruction quality. We observe that going deeper does not pay off—PhENN's function is too low level for VGG concepts to have any use. (This reinforces the notion that networks like PhENN *do not* themselves perform any "hidden" classification or other cognitive processing; rather, they reconstruct the images based on physical and geometrical priors learnt from the examples.) On the other hand, using shallower layers, ReLU-12 in particular, seems to perform well, recovering the high-frequency features in our sample. At the same time, the grid-like artifacts are strongly suppressed because their spatial frequency has been pushed to the edge of the Nyquist window at this layer depth (see Fig. 9).

Going even shallower does not pay off either—the ReLU-11-trained reconstructions are too blurry. This is perhaps because this VGG layer only processes very coarse features. Therefore, ReLU-12 is found to be the optimal VGG layer for PhENN's perceptual loss training.

4.2. Quantitative assessment of reconstructions by PLT-PhENN

We assessed reconstruction accuracy quantitatively according to several commonly used metrics: Peak Signal to Noise Ratio (PSNR) [42], Structural Similarity Index Metric (SSIM) [39,40], and Pearson Correlation Coefficient (PCC), which is defined as the NPCC [8,32] without the minus sign. From Table 1 and visual inspection of the samples in Figs. 7–8, we see that performance in the quantitative metrics and better observed visual quality are monotonic (this is not generally the case.) Reconstructions from ReLU-12 trained PhENN also achieve the best quantitative accuracy according to all three quantitative and overperform the NPCC-trained PhENN [32].

We also investigated reconstructions produced by PhENN trained with feature loss based on VGG19. The results, including PSDs of the test set, are in Appendix D and the quantitative assessment is contained in Table 2. As VGG19 is fundamentally similar to VGG16, we decided

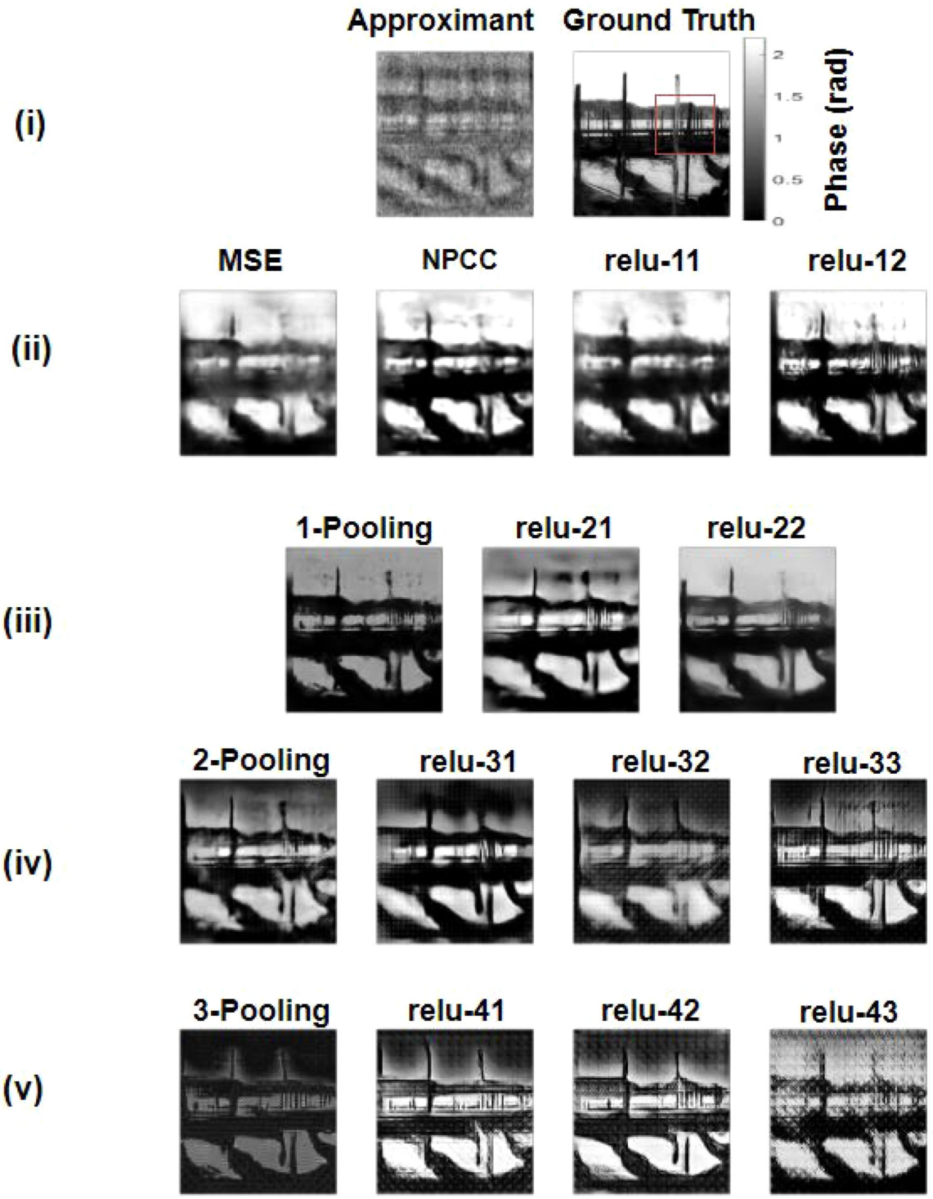


Fig. 7. Reconstructions with feature loss defined at various layers of VGG16 for $p = 1$. Row (i): Approximant and ground truth from a representative sample; rows (ii) to (v) each contain layers before 1-Pooling, after 1-Pooling, 2-Pooling and 3-Pooling, respectively.

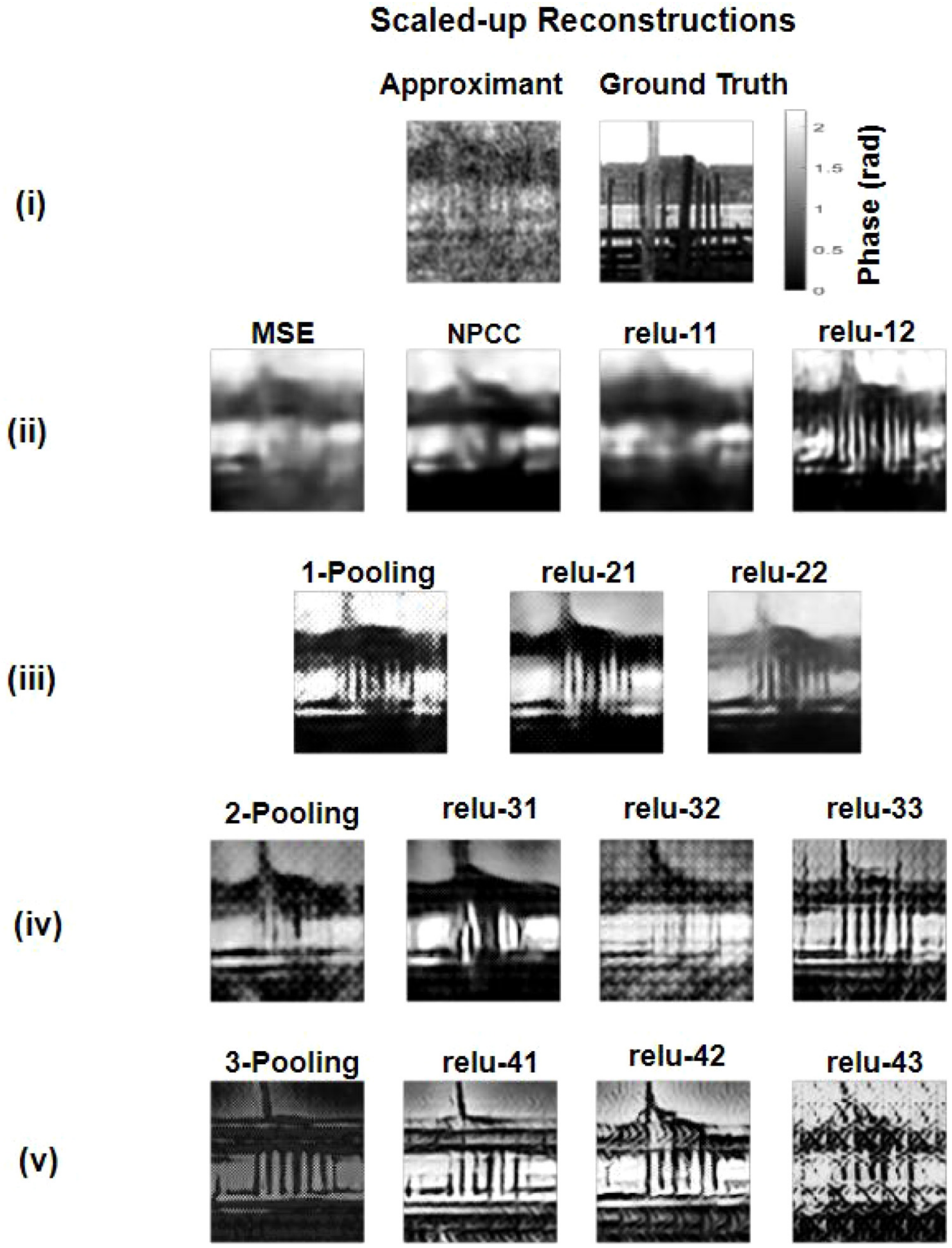


Fig. 8. Scaled-up reconstructions; the region is indicated by the red square in the ground truth image of Fig. 7, row (i). The rows correspond to those in Fig. 7.

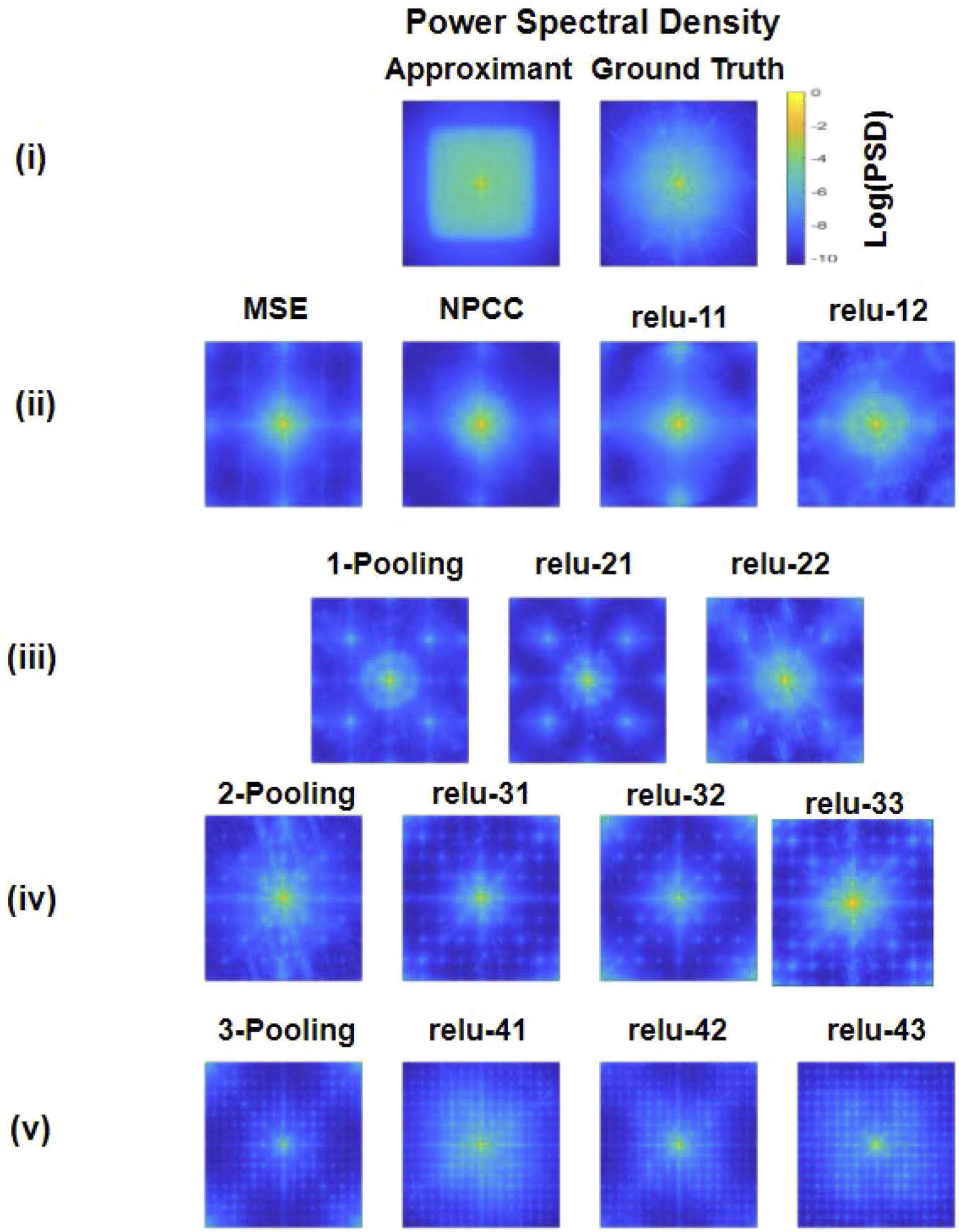


Fig. 9. Log-scale of PSDs of reconstructions, based on the entire test set of 50 randomly drawn samples. The rows correspond to those in Fig. 7.

that investigation of feature loss based on layers after the third convolutional block was unnecessary. Similar to VGG16, ReLU-12 in VGG19 produced reconstructions with both best quantitative metrics and visual quality. Compared to their counterparts in VGG16, the VGG19 ReLU-12 reconstructions have similar visual quality but slightly worse quantitative performance.

Table 1. Quantitative assessment of reconstructions by feature loss PLT-PhENN defined at various VGG16 layers. Each entry takes the form of average ± standard deviation.

	Average PSNR	Average PCC	Average SSIM
	± std. dev (dB)	± std. dev	± std. dev
image-MSE	11.523 ± 2.639	0.577 ± 0.237	0.687 ± 0.184
image-NPCC	16.207 ± 2.466	0.808 ± 0.099	0.875 ± 0.071
ReLU-11	15.943 ± 2.622	0.765 ± 0.109	0.866 ± 0.059
ReLU-12	16.719 ± 2.045	0.822 ± 0.094	0.891 ± 0.064
1-Pooling	14.569 ± 1.729	0.736 ± 0.131	0.837 ± 0.087
ReLU-21	12.853 ± 2.449	0.636 ± 0.149	0.755 ± 0.102
ReLU-22	13.589 ± 2.561	0.703 ± 0.129	0.800 ± 0.081
2-Pooling	12.610 ± 2.918	0.633 ± 0.142	0.742 ± 0.100
ReLU-31	12.236 ± 2.652	0.583 ± 0.178	0.713 ± 0.120
ReLU-32	11.578 ± 2.789	0.476 ± 0.203	0.636 ± 0.125
ReLU-33	11.867 ± 2.418	0.526 ± 0.162	0.684 ± 0.116
3-Pooling	10.823 ± 2.413	0.300± 0.141	0.553 ± 0.091
ReLU-41	11.874 ± 2.839	0.496 ± 0.210	0.668 ± 0.119
ReLU-42	12.441 ± 2.984	0.572 ± 0.203	0.714 ± 0.122
ReLU-43	12.890 ± 2.199	0.596 ± 0.142	0.755 ± 0.077
4-Pooling	11.429 ± 2.250	0.410 ± 0.169	0.635 ± 0.108

Table 2. Quantitative assessment of reconstructions by feature loss PLT-PhENN defined at various VGG19 layers. Each entry takes the form of average ± standard deviation.

	Average PSNR	Average PCC	Average SSIM
	± std. dev (dB)	± std. dev	± std. dev
image-MSE	11.523 ± 2.639	0.577 ± 0.237	0.687 ± 0.184
image-NPCC	16.207 ± 2.466	0.808 ± 0.099	0.875 ± 0.071
ReLU-11	16.627 ± 2.524	0.814 ± 0.087	0.883 ± 0.051
ReLU-12	16.663 ± 2.256	0.821 ± 0.107	0.886 ± 0.066
1-Pooling	15.687 ± 2.645	0.776 ± 0.125	0.857 ± 0.079
ReLU-21	13.690 ± 2.626	0.688 ± 0.124	0.800 ± 0.073
ReLU-22	12.461 ± 2.619	0.608 ± 0.153	0.732 ± 0.122
2-Pooling	10.514 ± 2.401	0.176 ± 0.121	0.489 ± 0.074
ReLU-31	12.847 ± 2.628	0.631 ± 0.166	0.752 ± 0.109
ReLU-32	12.595 ± 2.580	0.597 ± 0.172	0.736 ± 0.106
ReLU-33	12.027 ± 2.915	0.531 ± 0.209	0.678 ± 0.118
ReLU-34	12.336 ± 2.631	0.572 ± 0.188	0.714 ± 0.115

The discussion so far concerned using VGG-based loss as Feature Loss, see Section 2. We have also investigated the Mixed-loss scheme at various layers of VGG19, but we did not find it to be promising. The results are available in Appendix E.

5. Conclusions and future work

The main emphasis of this work has been on highly noisy raw intensity images and the value of using cognitive information, in the form of perceptual loss, to train a neural network to reconstruct phase images of high quality despite the noise. A standard image classification network, such as VGG16 or VGG19 is used to provide the cognitive information, and the image reconstruction neural network PhENN is trained to match the perceptual representations within the VGG. We discovered that the effectiveness of this strategy depends on the depth where VGG is probed for perceptual loss—equivalently, the degree of abstraction in the VGG's internal representations.

The optimal layer for perceptual probing was found to be ReLU-12, shallower than earlier recommendations. The new optimum proffers two advantages: recovery of fine-detail features and effective suppression of grid-like artifacts that had been observed earlier as well. Both aspects are adversely affected by noise, but the strategy we propose here is immune, down to the level of a single photon per pixel.

Another discovery reported in this paper is that the frequency of the artifacts depends on the MaxPooling2D layer within the VGG where the perceptual loss is drawn from. We also verified that increasing the level of (Poisson) noise in the input results in stronger artifacts. However, the actual origin of the artifacts remains unclear, inviting further investigation.

We expect the conclusions and intuition drawn here to apply to a great variety of inverse problems. However, it is important to note that exact quantitative guidelines, such as the ReLU depth for optimal perceptual loss, may turn out to be slightly different for inverse problems other than phase retrieval. This topic also warrants more extensive study.

Appendix A: Reconstructions produced by ReLU-22 trained PhENN at various photon levels

In Fig. 10, we compare reconstructions by Feature Loss-trained PhENN and NPCC-trained PhENN, at photon incidence levels of $p = 1, 10, 100, 1000$. From the comparison, we see that certain spatial details are well rendered by the Feature Loss-trained PhENN, but are not rendered by the NPCC-trained PhENN.

Optics EXPRESS

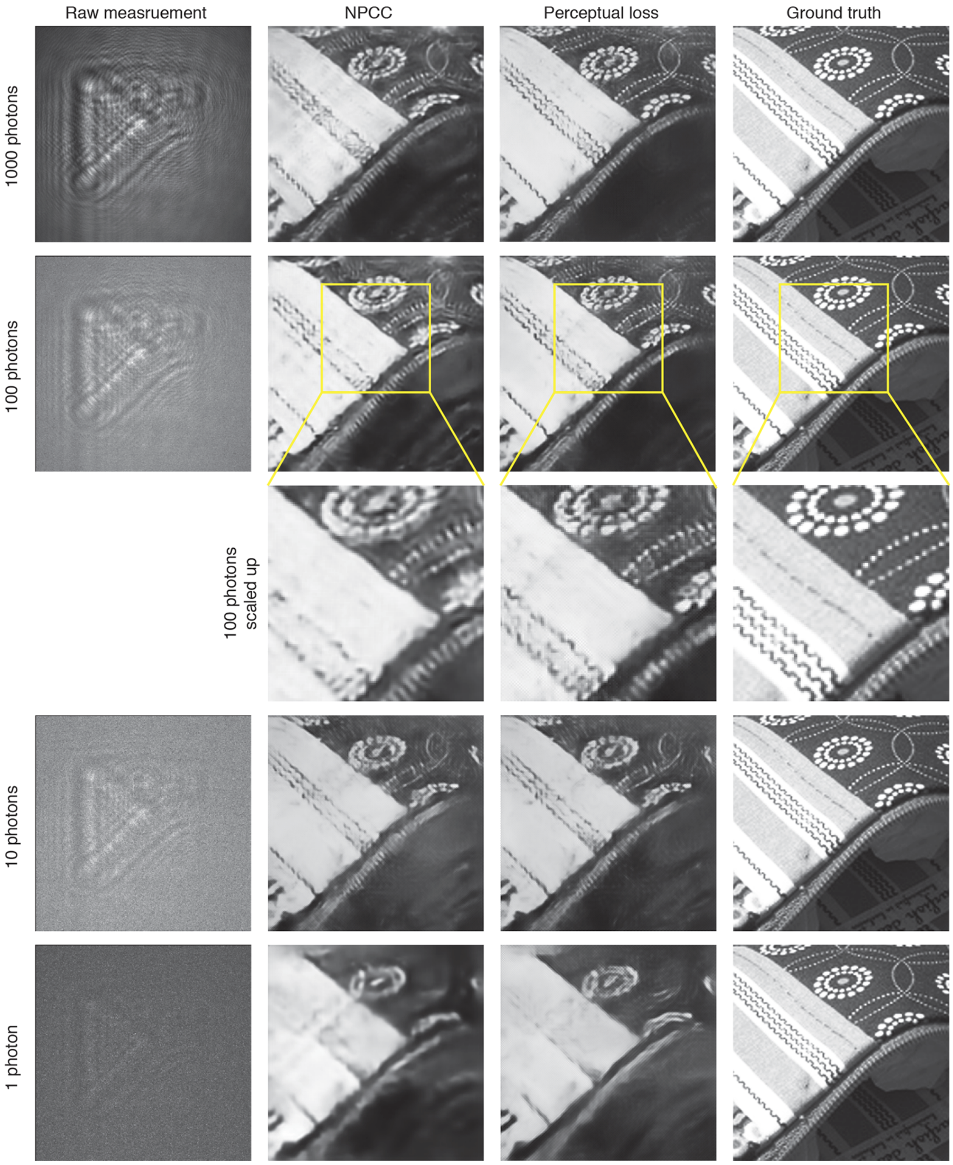


Fig. 10. comparison of reconstructions from PhENN trained with feature loss *vs.* NPCC for 1, 10, 100 and 1000-photon levels. In some areas, as shown by the scaled up images, some details are only visible in the feature loss reconstruction.

Appendix B: VGG16's effect on the fundamental frequency

To investigate the formation of grid-like artifacts and whether the pre-trained VGG treats the fundamental frequency any differently from others, we conduct the following test: for each ground truth image f_n in the test set, we generate a noisy version of it by adding noise at a particular spatial frequency (ν_{x0}, ν_{y0}) with amplitude A, empirically pre-defined to be 0.1. Thus, the strength of the artifacts in the noisy images are visually comparable with those in the reconstructions at 1 photon.

We define the noise signal as

$$\begin{aligned}\xi(A, x, y, \nu_{x0}, \nu_{y0}) = A\mathscr{F}^{-1}\{&e^{ia}\delta(\nu_x - \nu_{x0}, \nu_y - \nu_{y0}) + e^{-ia}\delta(\nu_x + \nu_{x0}, \nu_y + \nu_{y0}) \\ &+e^{ib}\delta(\nu_x + \nu_{x0}, \nu_y - \nu_{y0}) + e^{-ib}\delta(\nu_x - \nu_{x0}, \nu_y + \nu_{y0})\},\end{aligned} \tag{9}$$

where $\mathscr{F}$ is the Fourier transform, δ the Dirac impulse, and a and b two random real numbers uniformly distributed in $[-\pi, \pi]$. The noisy and clean images satisfy:

$$f_{\text{noisy},n}(A, x, y, \nu_{x0}, \nu_{y0}) = f_n + \xi(A, x, y, \nu_{x0}, \nu_{y0}) \tag{10}$$

We then submit the clean set $F = \{f_n, n = 1, \ldots, N_{\text{test}}\}$ and the corresponding noisy set, $F_{\text{noisy}}(A, \nu_{x0}, \nu_{y0}) = \{f_{\text{noisy},n}(A, \nu_{x0}, \nu_{y0}), n = 1, \ldots, N_{\text{test}}\}$, into the pre-trained VGG16 up to layer ReLU-22 and compute the sum of the losses for all examples n:

$$\mathscr{L}(f, f_{\text{noisy}}) = \frac{1}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} \mathscr{L}(f_n, f_{\text{noisy},n}) \tag{11}$$

The loss, $\mathscr{L}(f, f_{\text{noisy}})$ characterizes how the pre-trained VGG16 (up to ReLU-22) reacts to disparity at frequency (ν_{x0}, ν_{y0}). We scan the whole Fourier plane and compute a loss according to each frequency to understand responses of pre-trained VGG16 to disparities at all possible frequencies.

Here, we only show frequency scans along three representative directions, *i.e.* horizontal, vertical and diagonal (Fig. 11(a)). In Figs. 11(b)-(e), we show the corresponding profiles, for five randomly picked examples. We see that in all three scanning directions, for the majority of examples, there is a periodic artifact at frequency 0.25pixel^{-1}. The horizontal and vertical profiles display significantly different shapes (and magnitudes of the non-smooth artifacts), which indicates that the convolution filters in the pretrained VGG are not symmetric. This can be expected from the fact that, in the classification task for which VGG was trained, invariance to image orientation is important. In Fig. 11, we circled the portion of the loss curves where strong non-smoothness occurs. The magnitude and sign (*i.e.*, whether it is a positive of negative fluctuation) of the artifact vary from example to example. Therefore, we consider the mean absolute derivative of the loss function as a suitable metric to detect the non-smoothness (Fig. 11(e)).

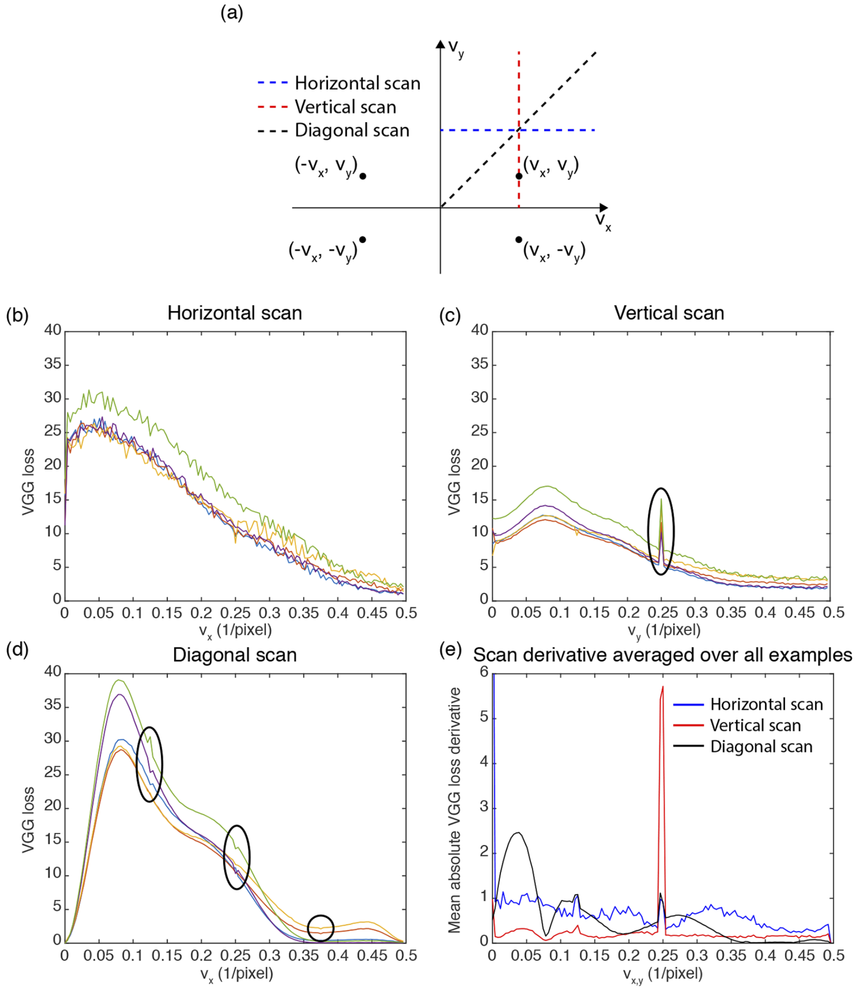


Fig. 11. Dependence of VGG16 loss on the frequency of the noise. (a) Diagram showing the scanning scheme in the Fourier domain. The noise n is added on at a single frequency and made Hermitian, *i.e.* $n(v_x, v_y) = n(-v_x, -v_y)^*$. (b) Loss as a function of frequency for the horizontal scan and five examples from the test set, for a noise amplitude of $A = 0.1$. (c) Loss as a function of frequency for the vertical scan for the same five examples. (d) Loss as a function of frequency for the diagonal scan for the same five examples. (e) Absolute value of the derivative of the loss with respect to frequency. The values are averaged over the 50 examples of the test set and plotted for the horizontal, vertical and diagonal scans. The ellipses in (c) and (d) indicate where strong non-smoothness can be observed in the loss curves. The position of the spikes correspond to artifact features observed in the spectrum of the average reconstruction.While we would not expect a perfect match between the reconstruction spectrum of Figs. 6(b)-(d) and the VGG frequency response in Figs. 11(b)-(d), we still expect that VGG displays a particular behavior at the fundamental frequency and that is, indeed, what we observe. Knowing that, in what follows, we investigate more deeply below in Appendix C the mechanism of why the perceptual loss based training leads to the survival of the artifact at the fundamental frequency.

Appendix C: Minimization of the perceptual loss

The results in the previous section suggest that the VGG network is primarily responsible for the appearance of the artifact. A common way to investigate the internal mechanism of a neural network is to compute so-called maximally activated patterns (MAPs) [56]. The idea is to find, through optimization, the input to the network that would maximize some metric defined on a given layer within the network. MAPs are thus functions of the particular layer on which they are defined. For the layer of interest, the MAP represents what the layer is most sensitive to. In a classification network, such as VGG, MAPs suggest what patterns may contribute most to the success of the classification. In the default feature loss training, we consider layer ReLU-22 and we are thus interested in the MAP defined for that particular layer.

Formally, we use the following definition of the MAP, based on the norm of the feature maps at layer ReLU-22:

$$\mathrm{MAP} = \underset{\eta}{\operatorname{argmax}} \left\{ \|\mathrm{VGG}(\eta)\| \right\} \text{ such that } \eta_p \in [0, 1] \tag{12}$$

where VGG stands for the mapping from an image to the VGG ReLU-22 layer, and η_p the pixels of η.

MAPs provide a methodology to study the response of DNNs to their inputs, and can suggest what input patterns may get amplified or suppressed through the network. Because of the possibly strong nonlinearity of the network, we suggest to consider the response of the ReLU-22 not with respect to the whole input itself (which would be the MAP defined in Eq. (12)), but rather with respect to perturbations added on top of input images from the ImageNet dataset. We propose to find out what perturbations are left over after a minimization of the perceptual loss $\mathscr{L}$ defined in Eq. (11). We expect that the artifact typically observed in the perceptual loss trained PhENN reconstructions lie in the set of perturbations only weakly affected by the minimization of the perceptual loss.

To that end, we perform numerical tests by initializing the minimization algorithm with a noisy version $f_{\text{noisy}}(\nu_{x0}, \nu_{y0})$ of the ground truth on which noise at a particular spatial frequency (as defined in Eq. (10)) has been added. That is:

$$\hat{f} = \underset{\eta}{\operatorname{argmin}} \left\{ \mathscr{L}\left(\eta, f\right) \right\}. \tag{13}$$

This minimization problem, because it is initialized with $f_{\text{noisy}}(\nu_{x0}, \nu_{y0})$, implicitly defines an operator from the noise $\xi(\nu_{x0}, \nu_{y0})$ to $\hat{f}$, which we can write as:

$$\hat{f}(\nu_{x0}, \nu_{y0}) = G_f[\xi(\nu_{x0}, \nu_{y0})]. \tag{14}$$

One may think that problem Eq. (13) necessarily converges to the ground truth (*i.e.* $\hat{f}(\nu_{x0}, \nu_{y0}) = f$ for all (ν_{x0}, ν_{y0})); however, due to the non-linearity in VGG16, it is not expected to be convex and may converge instead to a local minimum that depends on (ν_{x0}, ν_{y0}). We are interested in the following: at what frequency (ν_{x0}, ν_{y0}) does the noise get most reduced by minimizing VGG loss? In Fig. 12, we show of this test. Consistent with the observations in Section 5, the noise at the fundamental frequency undergoes the strongest suppression, which indicates that the disparity at the fundamental frequency would give rise to higher VGG loss than its neighbors.

The numerical experiments we conducted on VGG show that its frequency response share commonalities with the spectrum of the artifact, notably by the fact that non-smoothness is observed in the VGG spectrum at ν_{f}, $\frac{1}{2}\nu_{\mathrm{f}}$ and $\frac{3}{2}\nu_{\mathrm{f}}$. Moreover, we showed that minimization of the perceptual loss *per se* has uneven effect on the different frequencies of an image and that the typical artifact observed in the feature loss reconstruction survives the minimization process.

Therefore, when PhENN is trained to minimize the VGG-based feature loss, the training implicitly tends to match the reconstructions and the ground truth examples at the fundamental

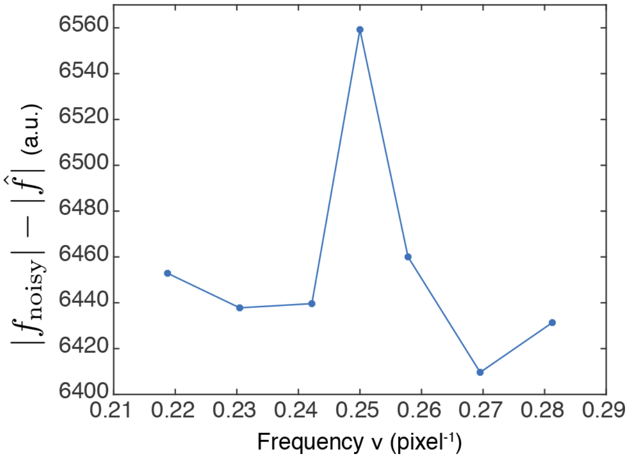


Fig. 12. Change in the frequency components of an image through the minimization operation of Eq. (13). The frequency ν refers to position $(\nu_x, \nu_y) = (\nu, \nu)$ in the Fourier domain (diagonal scan). The value plotted is the difference of the modulus of the spectrum of the noisy image $\tilde{f}$ (defined in Eq. (10)) and the spectrum of $\hat{f}$, the result of optimization Eq. (13) starting from $\tilde{f}$.

frequency more than its neighbors. Other frequencies may not need to be perfectly matched to achieve a low VGG loss, thus the training stagnates at some local minimum. At such local minimum, the fundamental frequency stands out due to deficiencies of its neighboring frequencies, manifesting as the artifacts centered at the fundamental frequency.

Appendix D: More constructions by VGG19 based feature loss

In this Appendix, we present reconstructions produced by VGG19-based feature loss, defined at various layers. From Fig. 13, we see although each MaxPooling2D layer does not reduce the

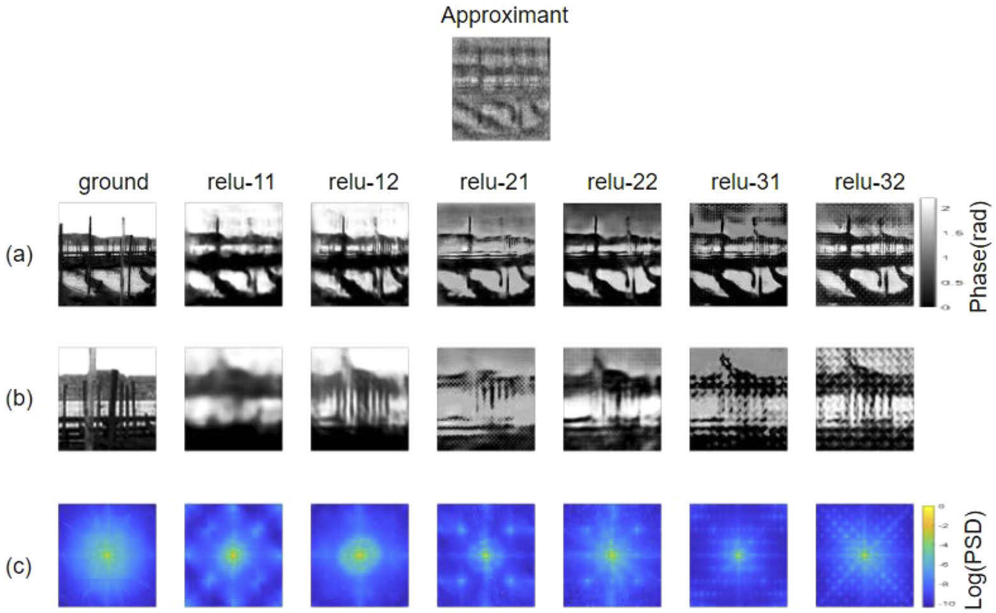


Fig. 13. Reconstructions by VGG19 feature loss trained PhENN

fundamental frequency of the artifacts exactly by half, as is the case in VGG16, the general trend that the deeper the defining layer is, the smaller the frequency is, still holds. From the PSDs and the reconstructions themselves, ReLU-12 produces reconstructions with highest visual quality, consistent with the quantitative superiority of ReLU-12 reconstructions in Table 2.

Appendix E: Reconstructions from the mixed loss scheme

As mentioned in Section 2, an alternative strategy is to use the weighted sum of the image-domain loss and VGG based loss. The success of such a method can be attributed to the addition of style information that the VGG loss, especially based on deeper layers, implants to the reconstruction, to the favor of the image-domain loss. Sometimes, a second neural network, called the discriminator, can be introduced and trained in turn with the image-transformation neural networks, forming the Generative Adversarial Networks (GANs) [57,58]. In GANs. the discriminator is trained to progressively better discriminate between the reconstructions that the generator produces, and the ground truth; the generator is subsequently trained to make the job of the discriminator increasingly more difficult, thereby improving the quality of reconstructions. GANs are proven efficient in producing high quality reconstructions [4,59], however, in some cases, it could also backfire, as its convergence highly depends on the network's hyper-parameters. In this severely ill-posed case, we anticipate that the advantage it brings is likely outweighed by its difficulty to converge. As such, we limit the scope of this paper by using mixed loss as defined in (2) and leaving the use of discriminator for future investigations.

Using the mixed loss strategy, we face the dilemma of choosing the parameter α_{feat}. Intuitively, α_{feat} being too small will lead the training to be indistinguishable from training with MSE,

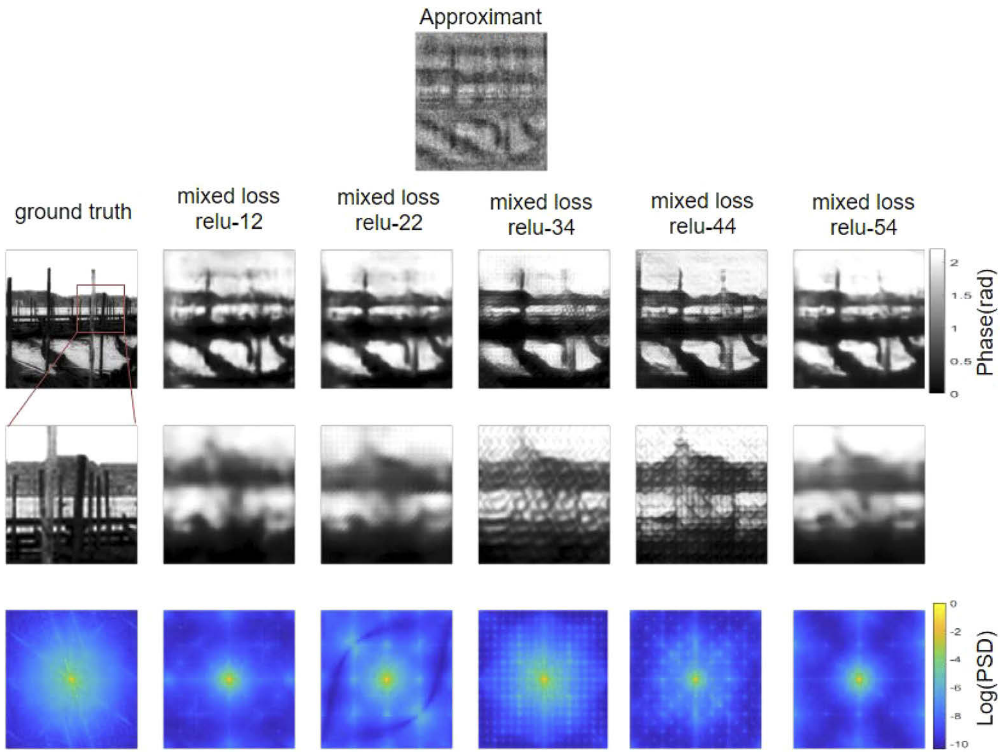


Fig. 14. Reconstructions and PSDs produced by mixed loss defined at various layers of VGG19

whereas, it being too large, the training would approach training with feature loss discussed earlier, hence losing potential advantage of the mixed-loss scheme. In fact, the rule of thumb for choosing α_{feat} is to make the two loss components in (2), the image domain loss and the VGG-based loss, be at the same order of magnitude during the training. However, finding the optimal α_{feat} in principle requires an impractically exhaustive scanning of the range of feasible α_{feat}, *i.e.* the ones that keeps the two components of loss in the same order of magnitude. Therefore, in what follows, for each layer, we sample a few representative values of such feasible α_{feat}'s and only report results with the best quantitative metrics and/or visual quality.

Here, anticipating the artifacts due to severe noise could affect the ideal choice of defining layer in the mixed loss setup, we expanded our investigation into mixed loss defined at the last ReLU layer of each CB. Representative results are presented in Fig. 14 and Table 3. We find that, although reconstructions from the mixed loss defined at ReLU-34, ReLU-44 and ReLU-54, have superior quantitative performances over the feature loss at ReLU-12, they are visually worse – either from oversmooth images with attenuated artifacts (*e.g.* ReLU-54) or from appearance of severe artifacts (*e.g.* ReLU-22,ReLU-34 and ReLU-54). In noiseless inverse problems, another common variant of mixed-loss is a composite loss using the *five layers*[50,60] of VGG19, *i.e.* ReLU-12, ReLU-22, ReLU-34, ReLU-44, ReLU-54. Anticipating facing a similar dilemma to the mixed-loss strategy just discussed, we decided that this strategy is not advisable in extremely noisy settings.

Table 3. Quantitative assessment of reconstructions by mixed-loss PLT-PhENN defined at various VGG19 layers. Each entry takes the form of average ± standard deviation.

	Average PSNR	Average PCC	Average SSIM
	± std. dev (dB)	± std. dev	± std. dev
image-MSE	11.523 ± 2.639	0.577 ± 0.237	0.687 ± 0.184
image-NPCC	16.207 ± 2.466	0.808 ± 0.099	0.875 ± 0.071
ReLU-12, $\alpha_{\text{feat}} = 0.0025$	15.562 ± 2.253	0.762 ± 0.121	0.856 ± 0.069
ReLU-22,$\alpha_{\text{feat}} = 0.0025$	17.090 ± 2.521	0.819 ± 0.096	0.890 ± 0.056
ReLU-34,$\alpha_{\text{feat}} = 0.0025$	17.404 ± 2.700	0.828 ± 0.095	0.901 ± 0.050
ReLU-44,$\alpha_{\text{feat}} = 0.04$	17.396 ± 2.595	0.828 ± 0.092	0.903 ± 0.050
ReLU-54, $\alpha_{\text{feat}} = 0.3$	17.891 ± 2.692	0.846 ± 0.086	0.910 ± 0.048

Funding

Intelligence Advanced Research Projects Activity (FA8650-17-C-9113).

Disclosures

The authors declare no conflicts of interest.

References

1. G. Barbastathis, A. Ozcan, and G. Situ, "On the use of deep learning for computational imaging," Optica **6**(8), 921–943 (2019).
2. Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature **521**(7553), 436–444 (2015).
3. T. Remez, O. Litany, R. Giryes, and A. M. Bronstein, "Deep convolutional denoising of low-light images," ArXiv:1701.01687v1 (2017).
4. C. Ledig, L. Theis, F. Huczar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. 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.
5. C. Dong, C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," IEEE Trans. on Pattern Analysis Mach. Intell. **38**, 295–307 (2015).

6. C. Dong, C. Loy, K. He, and X. Tang, "Learning a deep convolutional neural network for image super-resolution," in *European Conference on Computer Vision (ECCV), Part IV / Lecture Notes on Computer Science*, vol. 8692 (Springer, 2014), pp. 184–199.
7. J. Johnson, A. Alahi, and L. 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. (Springer, 2016), pp. 694–711.
8. S. Li, M. Deng, J. Lee, A. Sinha, and G. Barbastathis, "Imaging through glass diffusers using densely connected convolutional networks," Optica **5**(7), 803–813 (2018).
9. 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).
10. U. S. Kamilov, I. N. Papadopoulos, M. H. Shoreh, A. Goy, C. Vonesch, M. Unser, and D. Psaltis, "Optical tomographic image reconstruction based on beam propagation and sparse regularization," IEEE Trans. Comput. Imag. **2**(1), 59–70 (2016).
11. K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," IEEE Trans. Image Process. **26**(9), 4509–4522 (2017).
12. H. Gupta, K. H. Jin, H. Q. Nguyen, M. T. McCann, and M. Unser, "Cnn-based projected gradient descent for consistent ct image reconstruction," IEEE Trans. Med. Imag. **37**(6), 1440–1453 (2018).
13. T. C. Nguyen, V. Bui, and G. Nehmetallah, "Computational optical tomography using 3-d deep convolutional neural networks," Opt. Eng. **57**(4), 043111 (2018).
14. M. Elad and M. Aharon, "Image denoising via learned dictionaries and sparse representation," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1 (IEEE, 2006), pp. 895–900.
15. R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," Proc. IEEE **98**(6), 1045–1057 (2010).
16. J. W. Goodman and R. Lawrence, "Digital image formation from electronically detected holograms," Appl. Phys. Lett. **11**(3), 77–79 (1967).
17. K. Creath, "Phase-shifting speckle interferometry," Appl. Opt. **24**(18), 3053–3058 (1985).
18. R. W. Gerchberg, "A practical algorithm for the determination of phase from image and diffraction plane pictures," Optik **35**, 237–246 (1972).
19. J. R. Fienup, "Reconstruction of an object from the modulus of its Fourier transform," Opt. Lett. **3**(1), 27–29 (1978).
20. J. R. Fienup, "Phase retrieval algorithms: a comparison," Appl. Opt. **21**(15), 2758–2769 (1982).
21. J. Fienup and C. Wackerman, "Phase-retrieval stagnation problems and solutions," J. Opt. Soc. Am. A **3**(11), 1897–1907 (1986).
22. G. Zheng, R. Horstmeyer, and C. Yang, "Wide-field, high-resolution fourier ptychographic microscopy," Nat. Photonics **7**(9), 739–745 (2013).
23. 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).
24. M. R. Teague, "Deterministic phase retrieval: a Green's function solution," J. Opt. Soc. Am. A **73**(11), 1434–1441 (1983).
25. N. Streibl, "Phase imaging by the transport equation of intensity," Opt. Commun. **49**(1), 6–10 (1984).
26. A. Sinha, J. Lee, S. Li, and G. Barbastathis, "Lensless computational imaging through deep learning," Optica **4**(9), 1117–1125 (2017).
27. C. Metzler, P. Schniter, A. Veeraraghavan, and R. Baraniuk, "Prdeep: Robust phase retrieval with flexible deep neural networks. arxiv 2018," arXiv preprint arXiv:1803.00212 (2018).
28. Z. D. C. Kemp, "Propagation based phase retrieval of simulated intensity measurements using artificial neural networks," J. Opt. **20**(4), 045606 (2018).
29. L. Boominathan, M. Maniparambil, H. Gupta, R. Baburajan, and K. Mitra, "Phase retrieval for fourier ptychography under varying amount of measurements," CoRR abs/1805.03593 (2018).
30. Y. Jo, H. Cho, S. Y. Lee, G. Choi, G. Kim, H.-S. Min, and Y. Park, "Quantitative phase imaging and artificial intelligence: a review," IEEE J. Sel. Top. Quantum Electron. **25**(1), 1–14 (2019).
31. Y. Xue, S. Cheng, Y. Li, and L. Tian, "Reliable deep-learning-based phase imaging with uncertainty quantification," Optica **6**(5), 618–629 (2019).
32. A. Goy, K. Arthur, S. Li, and G. Barbastathis, "Low photon count phase retrieval using deep learning," Phys. Rev. Lett. **121**(24), 243902 (2018).
33. H. H. Bauschke, P. L. Combettes, and D. R. Luke, "Phase retrieval, error reduction algorithm, and Fienup variants: a view from convex optimization," J. Opt. Soc. Am. A **19**(7), 1334–1345 (2002).
34. S. Li and G. Barbastathis, "Spectral pre-modulation of training examples enhances the spatial resolution of the phase extraction neural network (PhENN)," Opt. Express **26**(22), 29340–29352 (2018).
35. M. Deng, S. Li, A. Goy, I. Kang, and G. Barbastathis, "Learning to synthesize: Robust phase retrieval at low photon counts," arXiv preprint arXiv:1907.11713 (2019).
36. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556 (2014).
37. A. Van der Schaaf and J. H. van Hateren, "Modelling the power spectra of natural images: statistics and information," Vision Res. **36**(17), 2759–2770 (1996).

38. M. S. Lewicki and T. J. Sejnowski, “Learning overcomplete representations,” Neural Comput. **12**(2), 337–365 (2000).
39. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” IEEE Trans. on Image Process. **13**(4), 600–612 (2004).
40. Z. Wang, E. P. Simoncelli, and A. C. Bovik, “Multiscale structural similarity for image quality assessment,” in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2 (IEEE, 2003), pp. 1398–1402.
41. S. Li, G. Barbastathis, and A. Goy, “Analysis of phase-extraction neural network (phenn) performance for lensless quantitative phase imaging,” in *Quantitative Phase Imaging V*, vol. 10887 (International Society for Optics and Photonics, 2019), p. 108870T.
42. P. Gupta, P. Srivastava, S. Bhardwaj, and V. Bhateja, “A modified psnr metric based on hvs for quality assessment of color images,” in *Communication and Industrial Application (ICCIA), 2011 International Conference on* (IEEE, 2011), pp. 1–4.
43. A. Mahendran and A. Vedaldi, “Understanding deep image representations by inverting them,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2015), pp. 5188–5196.
44. K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” arXiv preprint arXiv:1312.6034 (2013).
45. J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, “Understanding neural networks through deep visualization,” arXiv preprint arXiv:1506.06579 (2015).
46. L. Gatys, A. S. Ecker, and M. Bethge, “Texture synthesis using convolutional neural networks,” in *Advances in Neural Information Processing Systems*, (2015), pp. 262–270.
47. S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, “Toward convolutional blind denoising of real photographs,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2019), pp. 1712–1722.
48. S. S. Khan, V. Adarsh, V. Boominathan, J. Tan, A. Veeraraghavan, and K. Mitra, “Towards photorealistic reconstruction of highly multiplexed lensless images,” in *Proceedings of the IEEE International Conference on Computer Vision*, (2019), pp. 7860–7869.
49. J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” *2009 IEEE Conference on Computer Vision and Pattern Recognition*, (2009), pp. 248–255.
50. R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2018), pp. 586–595.
51. A. Dosovitskiy and T. Brox, “Generating images with perceptual similarity metrics based on deep networks,” in *Advances in neural information processing systems*, (2016), pp. 658–666.
52. A. Goy, K. Arthur, S. Li, and G. Barbastathis, “The importance of physical pre-processors for quantitative phase retrieval under extremely low photon counts,” in *Quantitative Phase Imaging V*, vol. 10887 (International Society for Optics and Photonics, 2019), p. 108870S.
53. O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention* (Springer, 2015), pp. 234–241.
54. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” (2015). Software available from tensorflow.org.
55. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980 (2014).
56. M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *European conference on computer vision* (Springer, 2014), pp. 818–833.
57. 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).
58. M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *International conference on machine learning*, (2017), pp. 214–223.
59. H. Wang, Y. Rivenson, Y. Jin, Z. Wei, R. Gao, H. Günaydın, L. Bentolila, C. Kural, and A. Ozcan, “Deep learning enables cross-modality super-resolution in fluorescence microscopy,” Nat. Methods **16**(1), 103–110 (2019).
60. Q. Chen and V. Koltun, “Photographic image synthesis with cascaded refinement networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, (2017), pp. 1511–1520.