Review Article Vol. 6, No. 8 / August 2019 / Optica 921

On the use of deep learning for computational imaging

George Barbastathis,[1,2,*] Aydogan Ozcan,[3] and Guohai Situ[4,5]

[1]Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139-4301, USA
[2]Singapore-MIT Alliance for Research and Technology (SMART) Centre, 1 Create Way, Singapore 138602, Singapore
[3]Department of Electrical & Computer Engineering, and Department of Bioengineering, University of California at Los Angeles, Engineering IV Building, Los Angeles, California 90095-1594, USA
[4]Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, Shanghai 201800, China
[5]University of the Chinese Academy of Sciences, Beijing 100049, China
*Corresponding author: gbarb@mit.edu

Received 13 March 2019; revised 16 May 2019; accepted 16 May 2019 (Doc. ID 362339); published 25 July 2019

Since their inception in the 1930–1960s, the research disciplines of computational imaging and machine learning have followed parallel tracks and, during the last two decades, experienced explosive growth drawing on similar progress in mathematical optimization and computing hardware. While these developments have always been to the benefit of image interpretation and machine vision, only recently has it become evident that machine learning architectures, and deep neural networks in particular, can be effective for computational *image formation*, aside from interpretation. The deep learning approach has proven to be especially attractive when the measurement is noisy and the measurement operator ill posed or uncertain. Examples reviewed here are: super-resolution; lensless retrieval of phase and complex amplitude from intensity; photon-limited scenes, including ghost imaging; and imaging through scatter. In this paper, we cast these works in a common framework. We relate the deep-learning-inspired solutions to the original computational imaging formulation and use the relationship to derive design insights, principles, and caveats of more general applicability. We also explore how the machine learning process is aided by the physics of imaging when ill posedness and uncertainties become particularly severe. It is hoped that the present unifying exposition will stimulate further progress in this promising field of research.

https://doi.org/10.1364/OPTICA.6.000921

1. INTRODUCTION

Computational imaging (CI) is a class of imaging systems that, starting from an imperfect physical measurement and prior knowledge about the class of objects or scenes being imaged, deliver estimates of a specific object or scene presented to the imaging system [1–7]. This is shown schematically in Fig. 1. The physical measurement, typically of light intensity on a digital camera, i.e., sampled on a pixel grid, we will refer to as raw measurement, or raw intensity image. The analytical relationship between the object and raw intensity image is the forward operator, whereas the analytical expression of prior knowledge is the regularizer, or simply prior. The premise on the existence of the regularizer is that, in general, any object that we measure in a natural scene obeys patterns that introduce correlations in the object's spatial features [8]; these patterns, if discovered, should help to reduce uncertainty in the recovery of the object, i.e., to reduce ill posedness. As we will review in some detail in Section 2, in a CI system, the estimate of the object is generally obtained by solving an optimization problem. The functional to be optimized promotes competition between fitting the raw measurement according to the forward operator, and observing the prior. We refer to the optimization outcome as object estimate, scene estimate, solution to the inverse problem or, simply, image.

The motivation for employing computation instead of using the raw intensity image directly is that the latter may not be readily interpretable. The information may be there, but incomplete or hidden; so further processing is needed to complement and reveal it. The CI way of thinking also opens up new directions in optical design because it removes the traditional requirement that the raw image itself must observe some metric of spatial conformity and fidelity with respect to the object. Thus, it is no longer necessary to fully correct all the aberrations before forming the raw image; some of the correction may be assigned to the computational components of the system. Perhaps more importantly, with CI, it becomes possible to create optical systems producing raw images that are deliberately spatially non-conforming to their respective scenes. The intent would then be after computation to form representations that are not readily available in the traditional spatially conforming intensity images: tomography, i.e., reconstructing the interior of a volume from projections; and

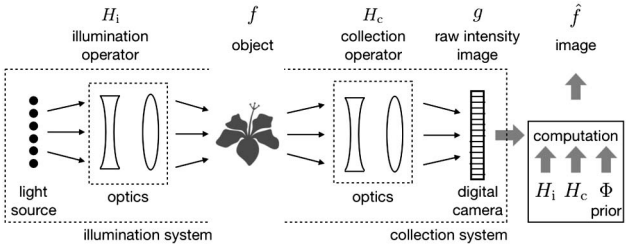

Fig. 1. General computational imaging system. The illumination source or source array is shaped by the condenser optics according to the operator H_i before reaching the object f. The radiation is subsequently shaped again by the imaging optics with collection operator H_c, and the intensity g is sampled by the digital camera. The signal g is then processed by the CI algorithm, which takes into account prior knowledge Φ about the class of objects being imaged (regularizer) in addition to the physical models H_i, H_c. The result of the computation is the image $\hat{f}$. For detailed notation and description, please see Section 2.

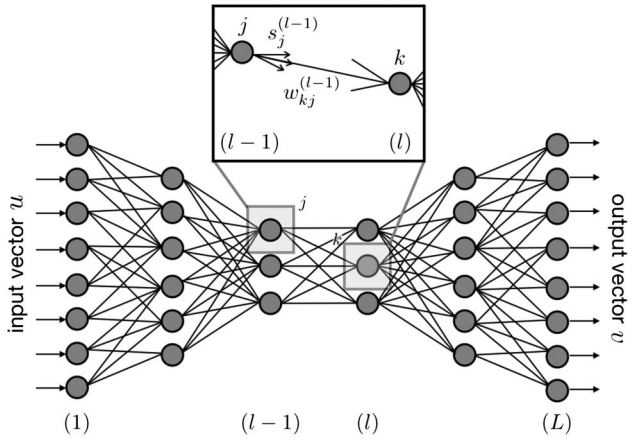

Fig. 2. Simplified schematic of a generic deep neural network. The input layer consists of the components of the input vector u. The dark circles denote the activation (nonlinear) elements, whereas $w_{kj}^{(l)}$ is the weight connecting the j-th activation element in layer (l) with the k-th activation element in layer $(l+1)$. The last (L)-th layer produces an estimate of the output vector v. The quality of the approximation relating input to output depends on the training; please see Section 3.

quantitative phase retrieval, i.e., extracting the phase of the optical field from the raw intensity, are both classical examples of this CI mode of operation.

Machine learning (ML) is a class of function interpolation algorithms that are informed by examples of the function being interpolated [9–13]. ML architectures are very general, e.g., they may be layered, recursive, etc. Later in this review, we will see that it is even possible to include explicit physical models as part of the computation. The specific architecture of interest here is based on the neural network (NN), a multilayered computational geometry. Each layer is composed of simple nonlinear processing units, also referred to as activation units (or elements); and each unit receives its inputs as weighted sums from the previous layer (except the very first layer, whose inputs are the quantities we wish the NN to process.) Until about two decades ago, students were advised to design NNs with up to three layers: the input layer, the hidden layer, and the output layer. Recent progress in ML has demonstrated the superiority of architectures with many more than three layers, referred to as deep NNs (DNNs) [14–17]. Figure 2 is a simplified schematic diagram of the multi-layered DNN architecture.

The process of adjusting the connection weights is known as training and is generally implemented as an optimization routine. In the supervised training mode, which is of most interest here, the weights are adjusted by minimizing the training loss function (TLF), a metric of the difference between the actual DNN output and its desired output. The former is determined by known DNN input–output pairs, the training examples. Thus, already the use of optimization emerges as a broad common ground between CI and ML. However, there is an important difference: in traditional CI, the optimization routine must be computed for each imaging operation, whereas in ML, the optimization is executed only during the training phase of the DNN. Imaging operations using ML, thus, are generally very fast because the trained DNN is a feed-forward computational architecture. Typical computation times may be from a few minutes to a few hours per frame for typical CI optimizations; several hours or days for DNN training; and milliseconds per frame for a trained DNN. Thus, as a rule, the ML approach yields images at rates approaching real time, even in common laptop computer or smartphone processors.

During the past few years, a number of researchers have shown convincingly that the ML formulation is not only computationally efficient, but it also yields high-quality solutions in several CI problems. In this approach, shown in Fig. 3, the raw intensity image is fed into a computational engine specifically incorporating ML components, i.e., multilayered structures as in Fig. 2 and trained from examples—taking the place of the generic computational engine in Fig. 1. CI problems so solved have included lensless imaging, imaging through scatter, bandwidth- or sampling-limited imaging (also referred to as "super-resolution"), and extremely noisy imaging, e.g., under the constraint of very low photon counts. With few early exceptions [18], the ML architectures used in these CI problems have relied on DNNs. The main purpose of this paper is to review these developments, providing a common framework for understanding and improving upon them as well as developing underlying design principles and intuition.

It is important early on to make a clear distinction between image interpretation versus image formation (or image transformation, as it is referred to in the computer vision research

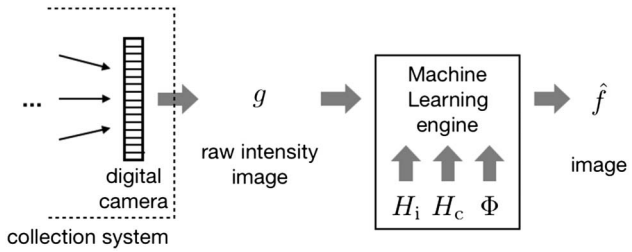

Fig. 3. CI architecture of Fig. 1 with an ML engine producing the image. As we see in Section 4, the ML engine generally includes a multilayered architecture such as the one shown in Fig. 2 and is informed on the physics of the illumination and collection optics, H_i, H_c, respectively, and the prior Φ. The three components H_i, H_c, Φ or their combinations are incorporated in the ML engine either explicitly as approximants (Section 3.F) or implicitly through training with examples.

community), and emphasize that the work presented here is exclusive to the latter. Methods for the former typically are *agnostic to the physics of imaging*, instead assuming that the input data to the ML algorithm are a vector or matrix of pixel values produced by a generic imaging system. These data are then used to arrive at a scalar or low-dimensional vector representation expressing classification or prediction outcomes e.g., identification of objects or faces in the scene, translating written text found in the scene, etc. The images used as input to the ML algorithm are assumed to meet the same spatial conformity and fidelity criteria that a human performing the image interpretation task would have demanded. Traditional image interpretation, in other words, aims to directly replace the human's function with an algorithm, with the algorithm operating on input similar to what the human's visual system would (or should) be receiving.

In contrast, the ML algorithms we are concerned with here are cognizant of imaging physics but *agnostic to the contents* of the object or scene, except to the degree that the contents provide regularizing priors. ML executes a regression operation: given an incomplete representation of the scene in the raw intensity measured by the camera, ML replaces the raw scene with one of higher spatial fidelity. In the process, information such as the phase of the field, which in the measured raw intensity may have been hidden or incomplete, is revealed. The user of the ML-produced images may well be a human demanding a guarantee that the images meet certain standards of fidelity to the true scene; or the images could be fed to another ML algorithm of the more common image interpretation type designed for a similar kind of spatial fidelity as the human would have required.

When the raw intensity data on the camera are not readily usable by a human or a traditional image interpretation algorithm for the purposes of recognition, an interesting dilemma occurs: does it make sense to first form an image of high spatial fidelity and then feed it to the image interpretation algorithm? Or should one design the image interpretation algorithm anew, operating directly on the incomplete raw intensity data? (For the human user, there is of course no choice: spatial fidelity is imperative.) The second approach has often been proven to be productive, e.g., for digital holographic particle localization and tracking [19,20] and diseased cell identification [21,22]; and recognition of handwritten characters through multimode fibers [23]. In this paper, we will not take a position on this dilemma or discuss it any further; yet, we wish to point out that, in our opinion, the dilemma is one of the least explored and most fascinating in the rapidly progressing field of merging ML into CI.

After these introductory remarks, in Section 2, we briefly summarize the principles of CI to establish notation and a common framework. With similar goals, we summarize ML architectures and design principles in Section 3, emphasizing those that have proven to be most useful for CI to-date and the principles for including physical knowledge of the forward operator into the ML computation. In Section 4, we first describe how ML principles apply specifically to CI, and then in Sections 4.A–4.D, respectively, we review the related literature on the following canonical CI problems: super-resolution, i.e., recovery of lost or suppressed spatial frequencies; lensless phase retrieval from intensity; imaging of dark scenes; and imaging through scatter. The use of ML in magnetic resonance imaging (MRI) and x-ray tomography [24–27] has been reviewed elsewhere in detail [28]; and overviews specific to the use of ML in computer vision-related reconstruction problems [29] and quantitative phase imaging [30] have also been published. Concluding remarks and further suggestions for future work are given in Section 5.

2. OVERVIEW OF COMPUTATIONAL IMAGING

A. General Formulation

Referring to Fig. 1, let f denote the object or scene that the imaging system's user wishes to retrieve. To avoid complications that are beyond the scope of this review, we will assume that even though objects are generally continuous, a discrete representation suffices [31–33]. Therefore, f is a vector or matrix matching the spatial dimension where the object is sampled. Light–object interaction is denoted by the illumination operator H_i, whereas the collection operator H_c models propagation through the rest of the optical system.

How f, H_i, and H_c are constructed depends on the state of the illumination and how light is scattered by the object and optical elements in the system. Possible models for H_c are thin transparency, Born or Rytov expansions, beam propagation, transmission matrix, etc. If multiple scattering is present in the optical system, e.g., a strong diffuser with f denoting the index of refraction distribution, then H_c becomes random and nonlinear. In Section 4.D, we discuss how using ML improves image quality in the latter most challenging case.

The illumination operator H_i is often designed to improve the condition of the inverse problem. Structured and coded illumination have been important for CI since the 1960s for improving resolution and sectioning contrast [34–51], and more recently for phase retrieval [22,52–54]. Another interesting instance is ghost imaging, whose ML implementation we discuss in Section 4.C. Similarly, coded aperture approaches are effective for designing the pupil function in H_c to improve performance [6,55–58].

If the thin object approximation is valid, $H_\mathrm{i}f$ is simply a multiplication whose outcome serves as input to the collection optics. For quasi-monochromatic spatially incoherent illumination, f is the modulus-squared of the complex transmittance as a function of lateral position, and $H_\mathrm{i}f$ is the intensity immediately after the object plane. For monochromatic spatially coherent light, f is the thin object's complex transmittance, and $H_\mathrm{i}f$ is the optical field. Sections 4.A and 4.B describe the ML implementations of super-resolution and quantitative phase retrieval, respectively, where these assumptions hold.

The output of the collection optics is optical intensity g, sampled and digitized at the output (camera) plane. After aggregating the illumination and collection models into the forward operator $H = H_\mathrm{c}H_\mathrm{i}$, the noiseless measurement model is

$$g = Hf. \tag{1}$$

Since the measurements are by necessity discrete, g is arranged into a matrix of the appropriate dimension or rastered into a one-dimensional vector. For a single raw intensity image, g may be up to two dimensional; however, if scanning is involved (as, e.g., in computed tomography where multiple projections are obtained with the object rotated at various angles), then g must be augmented accordingly. In ghost imaging, g is a one-dimensional intensity signal whose length equals the number of illumination patterns applied on the object.

Uncertainty in the measurements and/or the forward operator is the main challenge in inverse problems. Typically, an optical measurement is subject to signal-dependent Poisson statistics due to the random arrival of signal photons, and additive signal-independent statistics due to thermal electrons in the detector circuitry. Thus, the deterministic model (1) should be replaced by

$$g = \mathcal{P}\{Hf\} + \mathcal{T}. \tag{2}$$

Here, $\mathcal{P}$ generates a Poisson random process with arrival rate equal to its argument; and $\mathcal{T}$ is the thermal random process often modeled as additive white Gaussian noise (AWGN). In realistic sensors, noise may originate from multiple causes, such as environmental disturbances. For large photon counts, signal quantization is also modeled as AWGN. The nature of the noise underlies many choices in our subsequent developments, e.g., the form of the Tikhonov functional (3) and the related Wiener filter [1,2]. Extremely noisy imaging conditions where ML can be of value are discussed in Section 4.C.

B. Linear Inverse Problems, Regularization, and Sparsity

For linear forward operators H, the image is obtained by minimizing the Tikhonov [3,4] functional

$$\hat{f} = \underset{f}{\operatorname{argmin}}\{\|Hf - g\|_2^2 + \alpha\Phi(f)\}, \tag{3}$$

where $\|\cdot\|_2$ denotes the L^2 norm. The first term expresses fitness, i.e., matching in the least-squares sense the measurement to the forward model for the assumed object. The fitness term is constructed for AWGN errors, even though it is often used with more general noise models (2). The regularization parameter α expresses our relative belief in the measurement fitness versus our prior knowledge. Setting $\alpha = 0$ to obtain the image from the fitness term yields only the pseudo-inverse solution, or its Moore–Penrose improvement [59,60]. The results are often prone to artifacts and seldom satisfactory, due to ill posedness in the forward operator H. To improve, the second regularizing term $\Phi(f)$ is meant to compete with the fitness term, by driving the estimate $\hat{f}$ to also match prior knowledge about the class of objects being imaged.

Andrey Tikhonov [3,4] proposed the regularizer

$$\Phi(f) = \|f\|_2^2. \tag{4}$$

The solution to the inverse problem is then obtained explicitly as

$$\hat{f} = (H^T H + \alpha\mathbf{1})^{-1} H^T g, \tag{5}$$

where H^T is the transpose of H, and $\mathbf{1}$ is the unit tensor matching the dimension of $H^T H$. For $\alpha = 0$, this expression reduces to the pseudo-inverse solution, in imaging also known as direct deconvolution. Equation (5) is quite similar to the Wiener filter [1,2] if α is set to equal the noise-to-signal ratio. In the Tikhonov–Wiener solution, improvement over the pseudo-inverse is expected because $\hat{f}$ includes amplified noise, and keeping the L^2 norm to a low value competes with the amplification. Unfortunately, due to the action of H^T, the estimate $\hat{f}$ is typically low-pass filtered.

Instead of (4), modern regularizers $\Phi(\cdot)$ promote sparsity in the object f or its transformed version Sf. The linear transform S that converts the object to a sparse representation is called the sparsifier. Sparse or compressive sensing and imaging schemes [61–65] are intuitively justified because we know that most signals, especially images and video, are compressible; therefore, the degrees of freedom expressed by most images of interest are far fewer than the raw number of pixels implies. Moreover, if the raw images are corrupted by noise, S most likely will sparsify the signal but not the noise; as a result, the noise should be penalized and vulnerable to removal by the regularizer.

Compressed sensing is realized with either a fixed or learned set of basis functions. A common example of a fixed set is wavelet decomposition, which is highly redundant [66–70] and, therefore, typically results in a highly sparse representation of the signal. This is also the reason that wavelets are the basis of the JPEG compressed image format [71]. Another popular alternative is the class of nonlinear diffusion and Weikert operators [72–76]. The special case of

$$\partial_{\mathrm{TV}} \equiv \|\nabla_{x,y} f\|_2, \tag{6}$$

known as total variation (TV), rests on the notion that most images become sparse when edge enhanced. TV was used in one of the earliest demonstrations of compressed sensing in tomography [77], and subsequently for other CI problems including confocal microscopy [78], phase retrieval [75,79], holographic superlocalization [80,81], and phase tomography [82].

Early instances of the idea that sparse representations may instead be *learned* from examples were motivated by the power spectral density of natural images and analogies with the primate visual cortex [83–87]. This led to the concept of a dictionary operator D, ideally chosen such that the transformation

$$\tilde{f} = Df \tag{7}$$

results in a representation whose L^0 norm $\|\tilde{f}\|_0$ is minimized [88–91], i.e., a representation that is sparsest. Dictionaries have been used extensively in CI and inverse problems, e.g., for image upsampling [92,93], a form of super-resolution—see Section 4.A.

Sparsity-promoting regularizers typically involve non-differentiable metrics, e.g., L^1 [94], so their iterative optimization requires the use of proximal gradients [95]. Common algorithms include lasso [96], iterative shrinkage and thresholding algorithm (ISTA) [97] and its variants fast ISTA (FISTA) [98] and TwIST [99], message passing [100], coordinate descent (CoD) [101], alternating direction method of multipliers (ADMM) [102], and Adam [103]. Shortcomings in the iterative procedures specified by these algorithms, ISTA and CoD especially, inspired the idea of training the nonlinearities in a DNN to approximate the sparse regularizer with increased flexibility and computational efficiency [104–106]. We will pick up on this point in Section 3.F.

3. OVERVIEW OF NEURAL NETWORKS

A. Neural Network Fundamentals

The generic computational architecture of a NN is shown in Fig. 2. Its purpose is to learn and implement the map relating the input vector u to the output vector v. The standard NN consists of layers, and each layer consists of several simple activation units that are connected to the units in the preceding and succeeding layers. The number of layers L is the depth of the NN. The input vector is fed into the input layer $l = 1$, and the output vector emerges from the output layer $l = L$; the counter l is used to enumerate the layers. We will use the notation

$$v = \mathrm{NN}(u) \tag{8}$$

as the input–output relationship of the NN. The networks considered in this review for computational image formation range in

depth from $L \sim 10$ to 20. For pattern recognition tasks, e.g., ImageNet, it is not uncommon to have L in excess of 50.

The number of units in each layer is the layer's width. Unit width varies across the layers in various progressions. Classification tasks generally produce representations of much lower dimension than that of the input images; therefore, the width decreases progressively toward the output, following the contracting architecture in Fig. 4(a). Up-sampling tasks, as in the image super-resolution examples that we discuss in Section 4.A, require output dimension larger than the input, so expanding architectures such as Fig. 4(b) may be considered. The concatenation of the two is the encoder–decoder architecture in Fig. 4(c). The unit widths progressively decrease, forming a compressed (encoded) representation of the input near the waist of the structure, and then progressively increase again to produce the final reconstructed (decoded) image. In the encoder–decoder structure, skip connections are also used to transfer information directly between layers of the same width, bypassing the encoded channels. The U-net architecture [107] (see also Section 3.D and Fig. 7) was an early successful convolutional implementation of the encoder–decoder principle, and has been the most common implementation of ML approaches for CI to-date.

The layers are implemented as arrays of identical activation units. Each activation unit expresses a simple nonlinear function, typically the same for all units, which we will denote as $r(\cdot)$. With the exception of the first layer, the inputs to the units are weighted sums of the outputs of the units in the previous layer. Let $s_k^{(l)}$ denote the output of the k-th unit in the l-th layer, for $l = 2, \ldots, L$. This is obtained as

$$s_k^{(l)} = r\left(\sum_j w_{kj}^{(l-1)} s_j^{(l-1)} + w_{k0}^{(l-1)}\right), \tag{9}$$

where the summation takes place over all the units j of the previous layer, and $w_{k0}^{(l-1)}$ is included as a bias term. For the first layer, simply

$$s_k^{(1)} = u_k, \tag{10}$$

where u_k denotes the k-th element of the input vector u. Similarly, at the last layer,

$$s_k^{(L)} = v_k. \tag{11}$$

The coefficients $w_{kj}^{(l)}, l = 1, \ldots L-1$ are the weights, and the procedure of specifying the weights' values is referred to as training the NN. It involves an optimization process over the training examples, as we will see below. We will denote the collection of all weights, arranged in the appropriate data structure, as W.

The nonlinearity $r(\cdot)$ in modern NNs is most commonly implemented as the function

$$r(\xi) = \max(0, \xi) = \begin{cases} 0, & \text{if } \xi \le 0, \\ \xi, & \text{otherwise.} \end{cases} \tag{12}$$

This is known as the rectifying linear unit (ReLU). Sigmoidal functions, e.g., $\tanh(\cdot)$, used to be popular but have the problem of stagnating derivatives for large arguments, hampering training progress, as we will see later. This realization [108–110] motivated the adoption of ReLU in modern practice and, arguably, is one of the most significant factors making deep networks trainable.

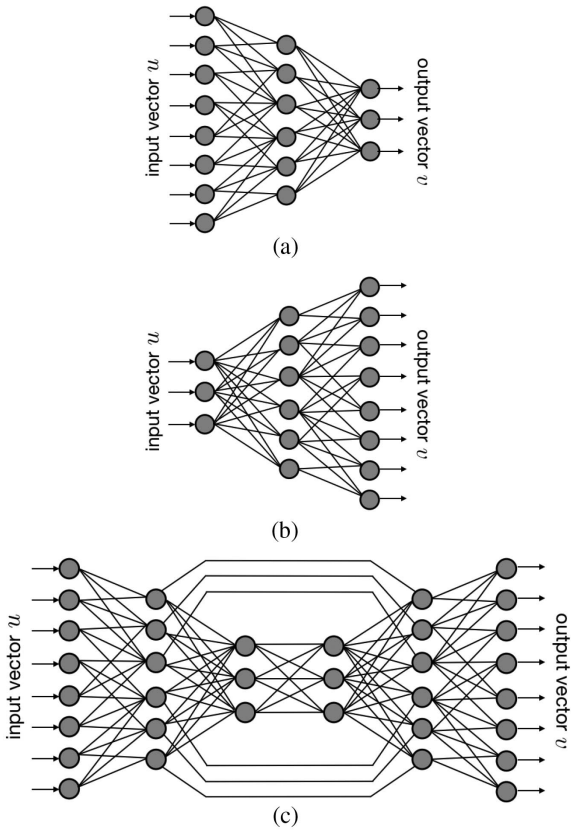

Fig. 4. Simplified schematics of layer width progression strategies for DNNs. (a) Contracting architecture. (b) Expanding architecture. (c) Encoder–decoder architecture.

B. Training and Testing Neural Networks

The power of NNs to perform demanding computational tasks is drawn from the complex connectivity between very simple activation units. The training process determines the connectivity from examples, and can be supervised or unsupervised. The supervised mode has generally been used for CI tasks, though unsupervised training has also been proposed [111–113]. After training, performance is evaluated from test examples that were never presented during training.

The supervised training mode requires examples of inputs u and the corresponding precisely known outputs $\tilde{v}$. In practice, one starts from a database of available examples and splits them to training examples, validation examples, and test examples. The training examples are used to specify the network weights; the validation examples are used to determine when to stop training; and the test examples are never to be used during the training process, only to evaluate it.

Suppose N such pairs of examples $\{u_{(n)}, \tilde{v}_{(n)}\}, n = 1, \ldots, N$ are available for training. The TLF is defined as the distance between the desired outputs $\tilde{v}_{(n)}$ and the actual outputs $v_{(n)}$ summed over all training examples as

$$\mathcal{L}\{u_{(n)}, \tilde{v}_{(n)}\}_{n=1,\ldots,N} \equiv \sum_n \mathcal{E}(\mathrm{NN}(u_{(n)}), \tilde{v}_{(n)}), \tag{13}$$

where $\mathcal{E}(\cdot, \cdot)$ is the distance metric. Common TLF choices for CI problems are discussed in Section E.

Training specifies the network weights W so as to minimize the TLF. Formally,

$$\tilde{W} = \underset{W}{\operatorname{argmin}}\, \mathcal{L}\{u_{(n)}, \tilde{v}_{(n)}\}_{n=1,\ldots,N}, \tag{14}$$

where $\tilde{W}$ is the optimal set of weights. $\mathcal{L}$ depends on the weights W implicitly through the input–output relationships of the NN and its nonlinear units, Eqs. (8) and (9), respectively. Since the activation functions are often non-differentiable, as in the ReLU (12), the optimization is carried out iteratively as a proximal gradient descent procedure. The iteration step t is called the training epoch, and the weights w evolve from each training epoch t to the next as

$$w(t+1) = w(t) + \eta\left[\frac{\partial \mathcal{L}}{\partial w}\right]. \tag{15}$$

Here, $[\cdot]$ denotes the proximal gradient, and η is the learning rate. If η is too small, convergence is too slow; if it is too large, the descent may oscillate around the minimum, also slowing progress down. Thus, an ideal value for the training rate exists, typically found by experimentation. At the beginning of training $t = 0$, the weights may be initialized randomly or according to other schemes, if the problem justifies it; see [114], Section 8.4.

In Fig. 2 and Eq. (9), it is evident that the input–output relationship NN$(\cdot)$ involves the computation of nested (implicit) instances of the nonlinearity $r(\cdot)$, and therefore, implementing the derivative in (15) will require multiple applications of the chain rule. This results in the computational procedure known as back propagation [115] (or back-prop.) To avoid excessive computational cost when dealing with large networks and training datasets, it is common to follow stochastic optimization approaches [116–120] using small, randomly chosen subsets of the dataset in each iteration. The optimizations are implemented in multidimensional numerical platforms such as TensorFlow [121] or Theano [122], and standard optimization libraries such as Adam [103] or ML-specific libraries, e.g., Caffe [123], CNTK [124], and Keras [125].

If M pairs of test examples $\{u_{(m)}, \tilde{v}_{(m)}\}, m = 1, \ldots, M$ are available, the test error is computed as

$$\mathcal{L}_{\text{test}}\{u_{(m)}, \tilde{v}_{(m)}\}_{m=1,\ldots,M} \equiv \sum_m \mathcal{E}_{\text{test}}(\text{NN}(u_{(m)}), \tilde{v}_{(m)}). \tag{16}$$

The test error metric $\mathcal{E}_{\text{test}}(\cdot,\cdot)$ quantifies the ability of the ML engine to *generalize*, i.e., to produce correct outputs v even for inputs u it has never seen before. $\mathcal{E}_{\text{test}}(\cdot,\cdot)$ need not be the same as the training metric $\mathcal{E}$; in fact, interesting conclusions are often drawn by comparing performance in different metrics. Of course, if the test metric is different from the training metric, there is no guarantee that test performance will be monotonic with training performance.

Even if the test metric is the same as the training metric, generally the two do not evolve in the same way during training. Recall that test examples are not supposed to be used *in any way* during training; however, the test error may be monitored and plotted as a function of training epoch t, and typically its evolution compared to the training error is as shown in Fig. 5. The reason test error begins to increase after a certain training duration is that overtraining results in overfitting: network function becomes so specific to the training examples that it damages generalization. It is tempting to use the test error evolution to determine the optimum training duration t_{opt}; however, that is not permissible because it would contaminate the integrity of the test examples. This is the reason we set aside the third set of validation examples; their only purpose is to monitor error on them, and stop training just before this validation error starts to increase. Assuming that all three sets of training, test, and validation examples have been drawn so that they are statistically representative of the class of objects of interest, there is a reasonable guarantee that t_{opt} for validation and test error will be the same.

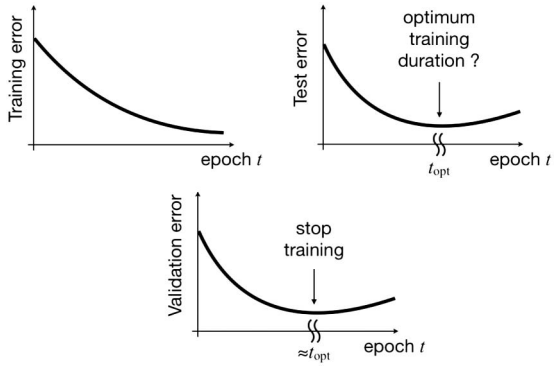

Fig. 5. Behavior of the training, test, and validation error as function of training epoch t.

The generalization ability of a computational learning architecture depends on many factors, including: the network architecture, i.e., the depth and the width of each layer; the choice of nonlinearity (ReLU generally performs better than sigmoids, as we saw); the quality of examples, i.e., if they were drawn representatively enough from the distribution of interest; and on the accuracy of the numerical algorithm used to implement Eq. (14). The design of NNs involves a certain degree of intuition and art, and is beyond the scope of the present review. Detailed strategies are given in textbooks, e.g., [114,126–132].

C. Weight Regularization

Overtraining and overfitting relate to the complexity of the model being learned *vis-à-vis* the complexity of the NN. Here, we use the term complexity in the context of degrees of freedom in a computational model [133,134]. For learning models, in particular, model complexity is known as Vapnik–Chervonenkis (VC) dimension [135–138], and it should match the complexity of the computational task. Unfortunately, the VC dimension itself is seldom directly computable except for the simplest ML architectures.

In practice, degrees of freedom in deep learning models are controlled by placing restrictions on the weights during the training process. This strategy is called weight regularization, and it is motivated by arguments similar to those in Section 2.B in the context of inverse problems. For example, weight decay [139] modifies the training functional (14) as

$$\tilde{W} = \underset{W}{\operatorname{argmin}}\, \mathcal{L}\{u_{(n)}, \tilde{v}_{(n)}\}_{n=1,\ldots,N} + \alpha \|W\|_2^2. \tag{17}$$

We recognize this as formally identical to the Tikhonov regularizer (4). The L_2 term suppresses weight values overall. This restricts the degrees of freedom the NN has to learn and improves generalization. It can be shown from (17), e.g., [127]–section 9.2, that in the absence of error, i.e., if TLF $= 0$, then the weights decay exponentially:

$$w[t] \sim w[0]\mathrm{e}^{-\eta\alpha t}; \tag{18}$$

hence, the name of the method.

More drastic is weight pruning: degrees of freedom are restricted by setting all weights below a certain threshold to zero, or by eliminating the activation units associated with weak weights [140,141]. Recent improvements are Dropout [142,143] and DropConnect [144], where instead multiple "thinned" versions of the network with shared weights are trained, each with subsets of activations or weights, respectively, randomly removed. The subnetworks are then recombined into a single network for testing. More complicated Bayesian strategies [145] sample models according to their posterior distributions given the training data. Alternatively, generalizability can be improved with a sparsity-promoting TLF, e.g., the minimum absolute error (MAE); see Section 3.E. One way to restrict weights appropriately for many CI circumstances is to exploit invariances in the object class, as in the convolutional model that we discuss in Section 3.D.

D. Convolutional Neural Networks

Certain tasks, such as speech and image processing, are naturally invariant to temporal and spatial shifts, respectively. This may be exploited to regularize the weights through convolutional architectures [146,147]. The convolutional NN (CNN) principle limits the spatial range on the next layer, i.e., the neighborhood where each unit is allowed to influence, and make the weights spatially repeating. Thus, (9) is replaced by

$$\zeta_j^{(l-1)} = r\left(\sum_{m=-M}^{M} w_m^{(l-1)} s_{j-m}^{(l-1)}\right), \tag{19}$$

where now $\zeta_j^{(l-1)}$ is an intermediate output of the $(l-1)$-th layer, r is the ReLU nonlinearity Eq. (12), and M is a small integer. Typically, $M = 1$, i.e., each unit influences only three immediate neighbors in the intermediate layer. Equation (19) needs trivial modification to remain valid at the layer edges.

In the encoder–decoder DNN architecture in Fig. 4(c), convolutions may be used to reduce the dimension of signal representation on the encoder side, i.e., to the left of the DNN waist, and increase it on the decoder side. The convolutional encoding principle is shown in Fig. 6. The intermediate outputs $\zeta_j^{(l-1)}$ of the $(l-1)$-th layer are first fed into a pooling layer, which combines neighboring ζ's to reduce the size of the l-th layer, typically by a factor of two. To avoid overcomplicating the notation, we will describe only the 1D case here; notation in higher dimensions becomes tensorial without altering the concept. Common choices for the pooling function are

$$\text{stride: } s_k^{(l)} = \zeta_{2k-1}^{(l-1)}, \tag{20}$$

$$\text{max pooling: } s_k^{(l)} = \max_k(\zeta_{2k-1}^{(l-1)}, \zeta_{2k}^{(l-1)}), \tag{21}$$

$$\text{average pooling: } s_k^{(l)} = \frac{\zeta_{2k-1}^{(l-1)} + \zeta_{2k}^{(l-1)}}{2}. \tag{22}$$

Stride is the simplest, and computationally most efficient, since half the convolutions need not even be carried out. Max pooling allows dominant features to propagate downstream without being influenced by their neighbors or by small shifts within a spatial domain. A word of caution on terminology: in the ML literature, this property is referred to as "invariance," whereas what standard

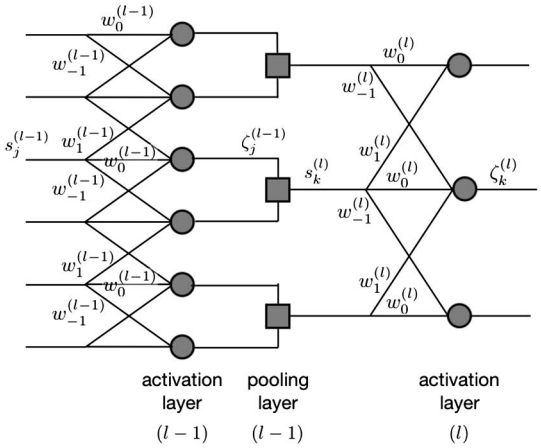

Fig. 6. Convolutional encoding principle between layers $(l-1)$ and l of decreasing width. Shift-invariant weights of limited range feed into activation units, e.g., ReLU, followed by a pooling layer.

optics textbooks [148] define as shift invariance, the ML literature refers to as "equivariance;" see [114], Sections 9.2 and 9.3.

A realistic implementation of an encoder–decoder type convolutional network from the first U-net realization [107] is shown in Fig. 7. The original purpose of U-net was image segmentation, but similar ML architectures have found wide use in CI problems, as we will see in Section 4. The layer width contracts through 3 × 3 convolutions and pooling (max, in this case) in the earlier encoding stages of the network. Multiple channels of convolutions are added to ensure information is not throttled by the contraction. Starting from input image size 572 × 572, by the deepest layer, the signal has contracted to 30 × 30 but with 1024 parallel convolution channels. In the subsequent decoding stages, upsampling is achieved through 2 × 2 convolutions and concatenating with same-width layers from the contracting path through the skip connections.

Subsequently, residual learning [149], or ResNet, was introduced to facilitate learning in very deep NNs, including U-nets. In the example illustrated in Fig. 8, let $s^{(l)}$ denote the vector of inputs (activation pattern) arriving at the l-th activation (ReLU) layer, and let $\mathcal{W}$ denote the (nonlinear) map between $s^{(l-1)}$ and

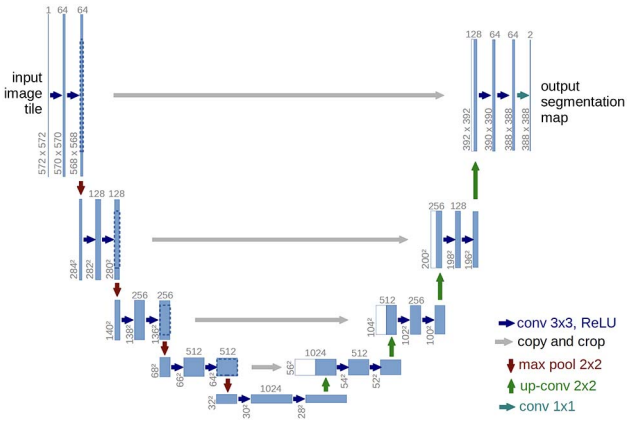

Fig. 7. Detailed implementation of the encoder–decoder architecture, Fig. 4(c), and the convolutional principle, Fig. 6, in the original U-net that was built for image segmentation (after [107], reprinted with permission from Springer).

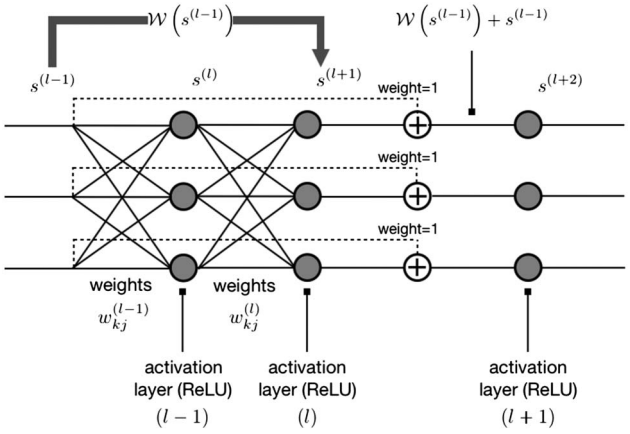

Fig. 8. Simplified schematic of residual learning through bypass connections with unit weight (dashed lines) [149].

$s^{(l+1)}$. Since $s^{(l-1)}$ is added through the unit-weight (dashed line) connection before the $(l+2)$-nd activation layer, $\mathcal{W}$ need learn only the difference $s^{(l+1)} - s^{(l-1)}$ rather than the complete map $s^{(l-1)} \to s^{(l+1)}$. With the residual method, it became possible for networks of depth in excess of ~50 to be trained and generalize well, whereas previously there had been a tendency in very deep networks to get "stuck" in local TLF minima with large training error.

E. Training Loss Functions

The most obvious TLF choices are the L^2 (minimum square error, MSE) and L^1 (MAE) metrics. They are defined, respectively, as

$$\mathcal{E}_{\mathrm{MSE}}(v,\tilde{v}) = \sum_p (v(p) - \tilde{v}(p))^2, \qquad \text{and} \tag{23}$$

$$\mathcal{E}_{\mathrm{MAE}}(v,\tilde{v}) = \sum_p |v(p) - \tilde{v}(p)|. \tag{24}$$

The summations take place over all the pixels p required to describe the object, and the equations are often also normalized for the number of pixels P^2 and/or the peak or average signal energies. As noted in the compressive imaging discussion in Section 2.B, the MAE promotes *sparsity* in the solution that minimizes the functional. Thus, the MAE acts effectively as a weight regularizer.

These metrics are pixel-wise, i.e., they do not take into account what happens in each pixel's neighborhood. For images, especially of natural objects, local correlations are prevalent, and neglecting them may lead to unnatural results. Instead, correlative metrics such as the negative Pearson correlation coefficient (NPCC) and the structural similarity image measure (SSIM) take local structure explicitly into account.

Given two arbitrary signals a, b of length P, let us define the mean

$$\langle a \rangle = \frac{1}{P}\sum_p a(p)$$

and covariance

$$C_{ab} = \frac{1}{P-1}\sum_p (a - \langle a \rangle)(b - \langle b \rangle).$$

Here, as in (23) and (24), the summations are over all pixels. The NPCC is defined as

$$\mathcal{E}_{\mathrm{NPCC}}(v,\tilde{v}) = -\frac{C_{v\tilde{v}}}{\sqrt{C_{vv}C_{\tilde{v}\tilde{v}}}}. \tag{25}$$

The negative sign maintains the optimization process as a minimization. Since Eq. (25) is essentially a (negative) dot product, it preserves structural similarity between the operands better than the MSE or MAE metrics. The Pearson metric originates from statistics, where it was proposed for measuring evolutionary correlations between subpopulations [150,151]. It has been disparaged as a metric for secure image comparison [152], yet it still has found some use in robotic vision [153,154]. In ML-based computational image inversion, it was found to do a better job than MAE to promote simultaneously weight sparsity and objects' spatial sparsity [155,156].

The (N)PCC may be viewed as a simplification of the more general SSIM index [157]:

$$\mathcal{E}_{\mathrm{SSIM}}(v,\tilde{v}) = \frac{(2\langle v\rangle\langle\tilde{v}\rangle + c_1)(2C_{v\tilde{v}} + c_2)}{(\langle v\rangle^2 + \langle\tilde{v}\rangle^2 + c_1)(C_{vv}^2 + C_{\tilde{v}\tilde{v}}^2 + c_2)}. \tag{26}$$

The SSIM index is often used in evaluating DNN performance on test examples after training. The coefficients c_1, c_2 stabilize the denominators and are chosen proportionally to the square of the dynamic range of the signals.

Sparsity is also promoted by the cross-entropy metric, inspired by information theory–see, e.g., [158] Sections 2.4 and 8.1–8.7, and also known for its utility in solving combinatorial optimization problems [159–162]. Typically, cross-entropy in ML is computed by first binarizing the training ground truth f and reconstruction data $\hat{f}$ and then penalizing deviations from one of the probability χ of agreement between f and $\hat{f}$ according to [107,144,163]

$$\mathcal{E}_{\mathrm{CE}} = \sum_p [\sigma(f)\log\chi + (1-\sigma(f))\log(1-\chi)], \qquad \text{where} \tag{27}$$

$$\sigma(f) = \frac{e^{f_k}}{\sum_j e^{f_j}} \quad \forall \text{ available instances } k \text{ of the data } f_k. \tag{28}$$

Modifications, e.g., focal loss [164], have also been proposed.

A related metric is the generative adversarial gain (payoff) [165]. In this case, we assume that outputs v and inputs u are drawn from probability distributions p_{data} and p_{model}, respectively. For CI, these should be interpreted as the prior distribution according to which objects f are generated and the probability of the optical system generating raw intensity images g, respectively. Then, two DNNs are used: the generator G and the discriminator D. The generator is given inputs u and produces samples v, while the discriminator emits a probability $d(v)$ that v belongs to the training set. The two networks are then set to compete against each other as the generator tries to minimize and the discriminator to maximize the discriminator's payoff:

$$\mathcal{E}_{\text{Disc-P}} = \mathrm{EV}_{v\sim p_{\mathrm{data}}}\log d(v) + \mathrm{EV}_{v\sim p_{\mathrm{model}}}(1 - \log d(v)). \tag{29}$$

Here, EV(·) denotes the expectation value, estimated over the available training examples $(u,\tilde{v})$. The adversarial model was originally proposed for embedding physics into ML engines for inversion, as discussed relating to Fig. 10(b) in Section 3.F.

Unfortunately, it is often the case that while TLFs or their combinations decrease during training, the visual fidelity of reconstructed images becomes worse. This phenomenon is well known in inverse problems literature [5,166,167]. The frustrating

disagreement between numerical metrics and image quality perception by human observers is not easy to resolve. Recently, computer vision researchers have started using Amazon's Mechanical Turk platform for that purpose [168,169].

To make image evaluation content-aware, a perceptual metric [170,171] has been proposed. The principle is shown in Fig. 9. During training, each training object f is accompanied by a content target f_c and/or a style target f_s. The training architecture consists of two DNNs: DNN_1 for the image reconstruction task, i.e., for producing an image $\hat{f}$ given the raw measurement g; and DNN_2, whose job is to compute the perceptual loss function $\mathcal{L}$. DNN_2 is structured as 16-layer VGG [172]. The VGG-16 is pre-trained to recognize objects in images, e.g., ImageNet [173]; during the training of DNN_1, DNN_2 remains fixed.

For an image reconstruction task, such as super-resolution (Section 4.A) the content target to be used in Fig. 9 is the object itself, i.e., $f_c = f$. Let $s^{(l)}(u)$ denote the vector of activation outputs of the l-th layer when the input to DNN_2 is u. The vectors $s^{(l)}(f)$ are encoded representations of objects f; indeed, it has been shown that using a regularized optimization routine similar to (3) these encoded representations may be "inverted" to reproduce estimates of their respective inputs f [174]. The key insight [170] is that these encoded representations also capture the perceptual difference between the ideal reconstruction f and an imperfect reconstruction $\hat{f}$ produced by DNN_1. Accordingly, the perceptual feature loss is defined as

$$\mathcal{E}_{\text{PFL}}(\hat{f}, f) = \sum_l \mu_l \|s^{(l)}(\hat{f}) - s^{(l)}(f)\|_2^2, \tag{30}$$

where the coefficients μ_l are for weighing the features from different layers (deeper layers tend not to preserve texture or fine shape details) and for normalizing according to the layer size (feature maps from deeper layers are vectors of smaller dimension in the VGG-16 network.) The training architecture in Fig. 9 can also account for style losses, such as texture and color [175]. The content-specific perceptual loss derived from activation layers within a VGG network may also be combined with an adversarial loss, used by a discriminator network that evaluates the fit of the reconstruction with natural images [171].

It is also common to use mixtures of error metrics for training, i.e., of the form

$$\mathcal{E} = \mu_1 \mathcal{E}_1 + \mu_2 \mathcal{E}_2 + \cdots$$

with coefficients $\mu_1, \mu_2, \ldots$ to be chosen according to the relative confidence the designer wishes to assign to the respective metrics $\mathcal{E}_1, \mathcal{E}_2, \ldots$ For example, Eq. (30) is typically modified to mix content- and style-sensitive perceptual losses, as well as other metrics such as MAE. There are no universal strategies for constructing the mixtures, and fine-tuning is often required.

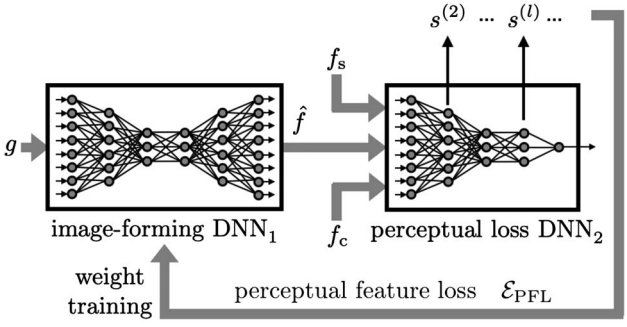

Fig. 9. Training an image-forming DNN_1 with a perceptual loss function generated by a content- and style-sensitive DNN_2 (after [170]).

F. Physics Priors

Unlike abstract classification, e.g., face recognition and customer taste prediction, in CI, the input g and intended output $\hat{f} \approx f$ of the NN are related by the known physics of the imaging system, i.e., by the operator H. Physical knowledge should be useful; how then to best incorporate it into an ML engine for imaging?

One possibility is to not incorporate it at all, as depicted in Fig. 10(a). In this end-to-end engine, the raw intensity measurement g is input directly to the DNN, and the image $\hat{f}$ is the direct output; in the notation of Section 3.A, Eq. (8):

$$\hat{f} = \text{NN}(g). \tag{31}$$

The burden then is on the training examples to teach the NN the forward operator H as well as the prior Φ that, as we saw in Section 2.B, is needed to regularize the problem. Despite this high burden, the end-to-end approach has been successful in several CI problems except when ill posedness or uncertainty become too high; see Section 4.C.

A definitive approach to incorporate the physical model (forward operator) into the ML engine originated from sparse representations, Section 2.B. Learned ISTA (LISTA), in particular,

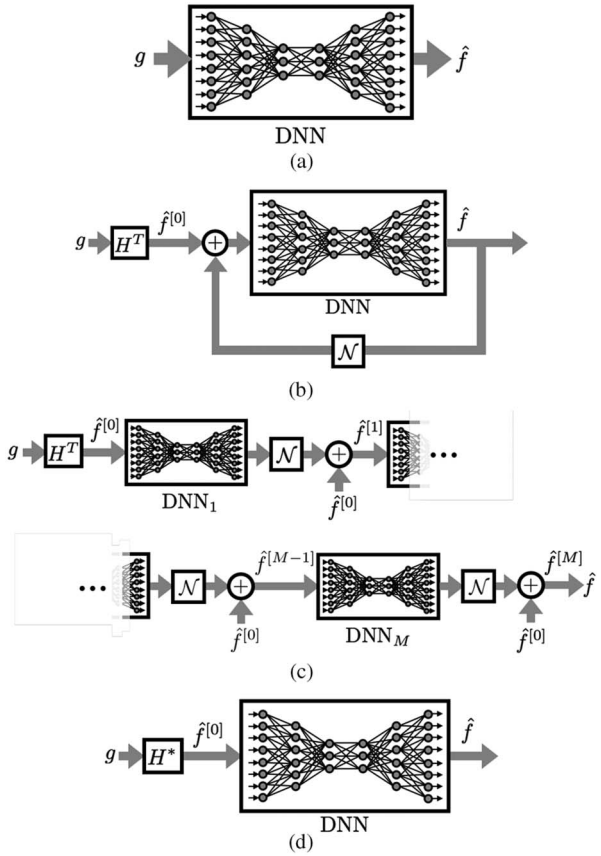

Fig. 10. Ways to implement the ML engine in Fig. 3, with or without physical priors. (a) End-to-end ML engine. (b) Recurrent physics-informed ML engine [176]. (c) Cascaded physics-informed ML engine [176]. (d) Single-pass physics-informed ML engine. Here, $H^* = H^T$ if the forward operator H is linear; otherwise, even a crude approximation to the nonlinear inverse has often been sufficient.

was an early proposal to incorporate a DNN into dictionary training for a Tikhonov–Wiener functional (3) [104–106,176,177]. Let us denote as H^T the operator conjugate to H and P_Φ the proximal gradient operator corresponding to $\alpha\Phi$. Following Mardani *et al.* [176], the proximal gradient descent approach is expressed as

$$\hat{f}^{[m+1]} = P_\Phi\left\{\hat{f}^{[m]} + \alpha H^T\left(g - H\hat{f}^{[m]}\right)\right\} \tag{32}$$

$$= P_\Phi\left\{\alpha H^T g + (\mathbf{1} - \alpha H^T H)\hat{f}^{[m]}\right\}. \tag{33}$$

Here, $f^{[m]}$ is the m-th iterate, and α is a (small) step size for the iterative update (32). Expression (33) suggests the recurrent architecture shown in Fig. 10(b), where

$$\mathcal{N} = \mathbf{1} - \alpha H^T H \tag{34}$$

denotes the projection operator onto the null space of H, and the proximal gradient operator P_Φ has been replaced by a recurrent DNN. In principle, if the available examples succeed in training the DNN to the priors and the parameter α, then the DNN can replace the job of the regularizer and its proximal gradient. In the original recurrent implementation, the DNN in Fig. 10(b) was the generator, the output $\hat{f}$ was fed into a discriminator DNN, and the two DNNs were trained according to the adversarial model [165] (see also Section 3.E.).

In practice, instead of the recurrent architecture in Fig. 10(b), it is easier to implement the unfolded, or cascaded ML engine shown in Fig. 10(c). In this version, the measurement g first produces an estimate $\hat{f}^{[0]}$ according to the conjugate operator H^T only. We will refer to $\hat{f}^{[0]}$ as the zeroth-order approximant, or simply the approximant. This approximant result is correct within the conjugate operator's range. To recover information outside the range, i.e., from the conjugate operator's null space, the approximant is passed consecutively through M cascaded DNNs. The output at the m-th stage is projected onto the null space by the operator $\mathcal{N}$, and the zeroth-order approximant is added [see Eq. (33)] to produce the m-th-order approximant. The final image is the M-th-order approximant $\hat{f}^{[M]}$. Note also that in this architecture, there is no discriminator; the DNN cascade is trained on an appropriately chosen loss function (see Section 3.E.).

The advantage of the cascaded ML engine is that it is faithful to the original Tikhonov–Wiener formulation. The forward operator is explicitly included while the NNs, in a sense, are replacing the regularizer, and the prior knowledge is now learned from examples rather than explicitly defined. One disadvantage is that training M DNNs increases the complexity of the learning task and the associated risks of undertraining (not enough training examples) or overtraining (too many degrees of freedom).

A compromise is the single-pass ML engine in Fig. 10(d). Here, an approximate inverse operator H^* produces the single approximant $f^{[0]}$. The single DNN is trained to receive $f^{[0]}$ as input and produce the image $\hat{f}$ as output directly, rather than its projection onto the null space. In practice, the single-pass approach has proven to be robust and reliable even for CI problems with high ill posedness or uncertainty, as we will see in Sections 4.A–4.C. Last, it is important to note that H must be linear for Eqs. (32) and (33) to have a fixed point; only then are Figs. 10(b) and 10(c) valid. If this is the case, then the single-pass approximant should be $f^{[0]} = H^* g = H^T g$. For nonlinear inverse problems, such as phase retrieval and imaging through strong scatter, Sections 4.B and 4.D, respectively, we will see that end-to-end or single-pass engines work well even with crude approximations to H^*.

4. COMPUTATIONAL IMAGING REALIZATIONS WITH MACHINE LEARNING

The strategy for using ML for computational image formation is broadly described as follows:

(1) Obtain a database of *physical realizations* of objects and their corresponding raw intensity images through the instrument of interest. For example, such a physical database may be built by using an alternative imaging instrument considered accurate enough to be trusted as ground truth; or by displaying objects from a publicly available abstract database, e.g., ImageNet [178] on a spatial light modulator (SLM) as phase or intensity; or by rigorous simulation of the forward operator and associated noise processes.

(2) Decide on an ML engine, regularization strategy, TLF, and physical priors according to the principles of Sections 3.C–3.F, and then train the NN from the training and validation subsets of the database, as described in Section 3.B.

(3) Test the ML engine for generalization by measuring a TLF, same as training or different, for on the test example subset of the database.

Generating the physical database for CI tasks can be hardware intensive and time consuming. Moreover, the quality of training depends on the examples chosen; therefore, care must be taken that they be representative of the class of objects the ML architecture would be used on after training. For highly ill-posed and noisy tasks, it is generally better to restrict the class so that stronger priors are learned. In such cases, it is also a good strategy to adopt an ML engine including the forward operator explicitly, rather than end-to-end (Section 3.F and Fig. 10).

This section is organized according to the type of forward operator the ML algorithm is called upon to invert, and the severity of the ill-posed problem: super-resolution in photography, machine vision, and optical microscopy in Section 4.A; complex amplitude retrieval from intensity in Section 4.B; attenuation and phase retrieval under extremely low photon counts in Section 4.C; and imaging through diffusers in Section 4.D. We also wish to remind the reader of three recent and more focused reviews on tomography [28], computer-vision related reconstruction problems [29], and quantitative phase imaging [30].

A. Super-Resolution

The two-point resolution problem was first posed by Airy [179] and Lord Rayleigh [180]. In modern optical imaging systems, resolution is understood to be limited by mainly two factors: undersampling by the camera, whence super-resolution should be taken to mean upsampling; and blur by the optics or camera motion, in which case super-resolution means deblurring. Both situations or their combination lead to a singular or severely ill-posed inverse problem due to suppression or loss of entire spatial frequency bands; therefore, they have attracted significant research interest, including some of the earliest uses of ML in the CI context.

A comprehensive review of methods for super-resolution in the sense of upsampling, based on a single image, is in [181]. To our knowledge, the first-ever effort to use a DNN in the same context was by Dong *et al.* [182,183]. The key insight, as with LISTA

(Section 3.F), was that dictionary-based sparse representations for upsampling [92,93] could equivalently be learned by DNNs. Both approaches similarly start by extracting compressed feature maps and then expanding these maps to a higher sampling rate. The difference is that sparse coding solvers are iterative; whereas, as we also pointed out in Section 1, with the ML approach, the iterative scheme takes place during training only; the trained ML engine operation is feed-forward and, thus, very fast. To combine super-resolution with motion compensation, a spatio-temporal CNN has been proposed, where, rather than simple images, the inputs are blocks consisting of multiple frames from video [184].

The ML approach to the super-resolution problem also served as motivation and testing ground for the perceptual TLF [170,171] (Section 3.E). The structure of the downsampling kernel was exploited in [177] using the cascaded ML engine architecture in Fig. 10(c) with $M = 4$. Figure 11 is a representative result showing the evolution of the image estimates along the ML cascade, as well as their spatial spectra. It is interesting that, by the final stage, the ML engine has succeeded in both suppressing high-frequency artifacts due to undersampling and boosting low frequency components to make the reconstruction appear smooth.

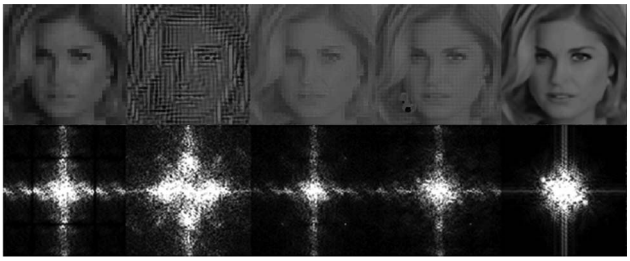

Fig. 11. Evolution of the image estimate along the cascaded ML engine in Fig. 10(c) (reprinted with permission from [177], Fig. 10). Top row, left to right, and in the notation used in the present review: undersampled raw intensity image g, $\hat{f}^{[0]}$, $\hat{f}^{[1]}$, $\hat{f}^{[2]}$, and $\hat{f} \equiv \hat{f}^{[3]}$. Bottom row: spatial Fourier transform magnitude of the corresponding images from the top row.

Turning to inverse problems dominated by blur, early work [185] used a perceptron network with two hidden layers and a sigmoidal activation function to compensate for static blur caused by Gaussian and rectangular kernels, as well as motion blur [186]. Two years later, Sun Jiao *et al.* [187] showed that a CNN can learn to compensate even when the motion blur kernel across the image is non-uniform. This was accomplished by feeding the CNN with rotated patches containing simple object features, such that the network learned to predict the direction of motion.

In optical microscopy, blur is typically caused by aberrations and diffraction [188]. More than 100 years of research, tracing back to Airy and Rayleigh's observations, have been oriented toward modifying the optical hardware—in our language, designing the illumination and collection operators—to compensate for the blur and obtain sharp images of objects down to sub-micrometer size. Thorough review of this literature is beyond the present scope; we just point out the culmination of optical super-resolution methods with the 2014 Nobel Prize in Chemistry [189–192]. Stochastic optical reconstruction microscopy (STORM) and fluorescence photoactivation localization microscopy (PALM) for single molecule imaging [193,194] and localization [195] are examples of co-designing the illumination operator H_i and the computational inverse to achieve performance vastly better than an unaided microscope could do.

Computationally, the blur kernel can be compensated for through iterative blind deconvolution [196,197] or learned from examples [198]. A DNN-based solution to the inverse problem was proposed for the first time, to our knowledge, by Rivenson *et al.* [199] in a wide-field microscope. The approach and results are summarized in Fig. 12. For training, the samples were imaged twice, once with a 40 × 0.95 NA objective lens and again with a

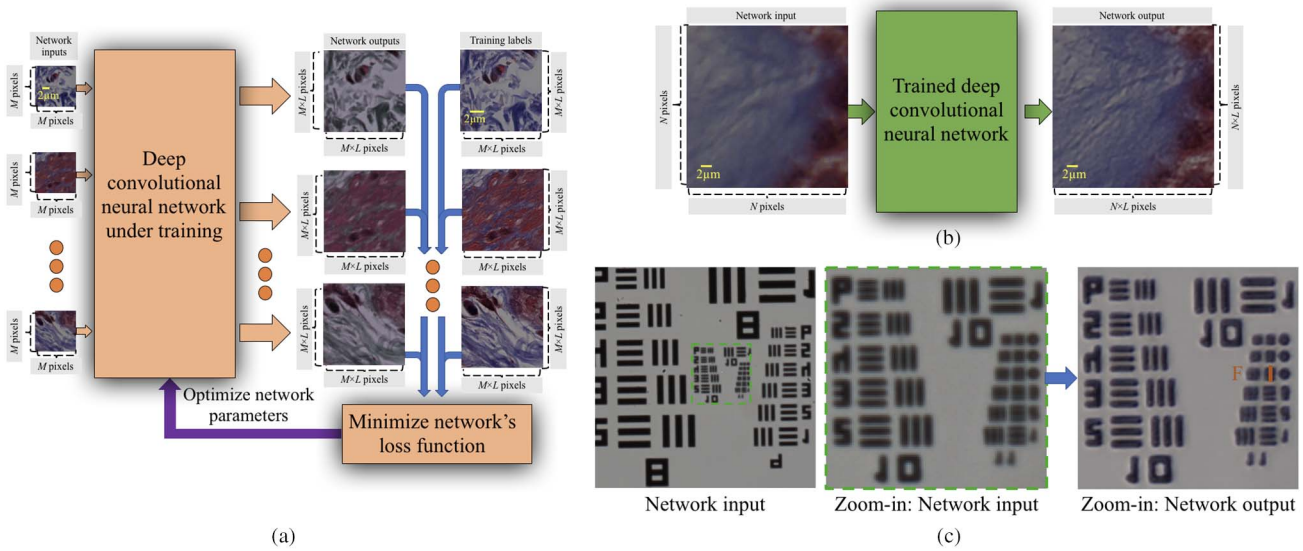

Fig. 12. Deep learning microscopy (adapted from [199], Figs. 1 and 5, with permission). (a) Training of the end-to-end ML engine with low-resolution (downsampled and blurred) inputs to the DNN produced by a 40 × 0.95 NA objective lens and high-resolution outputs produced by a 100 × 1.4 NA objective lens. (b) Operation of the trained DNN engine, with the test outputs successfully upsampled and deblurred. (c) Resolution test demonstrating the DNN with inputs from the 40 × 0.95 NA objective lens achieving in testing performance comparable to the raw performance of the 100 × 1.4 NA objective lens the DNN was trained with.

100 × 1.4 NA objective lens. The training goal was such that with the 40 × 0.95 NA raw images as input g, the DNN would produce estimates $\hat{f}$ matching the 100 × 1.4 NA images, i.e., the latter were taken to approximate the true objects f. The number of pixels in the high-resolution images was $(2.5)^2$ × the number of pixels in the low-resolution representation. Of course, the low-resolution images were also subject to stronger blur due to the lower-NA objective lens. Therefore, the inverse algorithm had to perform both upsampling and deblurring in this case. The ML engine was of the end-to-end type, as in Fig. 10(a), implemented as convolutional DNN with pyramidal progression for upsampling. The TLF was a mixture of the MSE metric (23) and a TV-like ∂^2_{TV} [Eq. (6)] penalty. Since then, ML has been shown to improve the resolution of fluorescence microscopy [200], as well as single-molecule STORM imaging [201] and 3D localization [202].

B. Quantitative Phase Retrieval and Lensless Imaging

The forward operator relating the complex amplitude of an object to the raw intensity image at the exit plane of an optical system is nonlinear. Classical iterative solutions are the Gerchberg–Saxton algorithm [203,204]; the input–output algorithm, originally proposed by Fienup [205] and subsequent variants [206–208]; and the gradient descent [209] or its variants, steepest descent and conjugate gradient [210]. This inverse problem has attracted considerable attention because of its importance in retrieving the shape or optical density of transparent samples with visible light [211,212] and x rays [213,214].

In the case of weak scattering, the problem may be linearized through a quasi-hydrodynamic approximation leading to the transport of intensity equation (TIE) formulation [215,216]. Alternatively, if a reference beam is provided in the optical system, the measurement may be interpreted as a digital hologram [217], and the object may be reconstructed by a computational back-propagation algorithm [218,219] (not to be confused with the back-propagation algorithm for NN training, Section 3.B.) Ptychography captures measurements effectively in the phase (Wigner) space, where the problem is linearized, by modulating the illumination with a quadratic phase and structuring it so that it is confined and scanned in either space [220–224] or angle [225–227]. Due to the difficulty of the phase retrieval inverse problem, compressive priors have often been used to regularize it in its various linear forms, including digital holography [228,229], TIE [82,230], and Wigner deconvolution ptychography [231,232].

When the linearization assumptions do not apply or regularization priors are not explicitly available, an ML engine may instead be applied directly on the nonlinear inverse problem. To our knowledge, this investigation was first attempted by Sinha *et al.* with binary pure phase objects [233], and subsequently with multi-level pure phase objects [234]. Representative results are shown in Fig. 13. The phase objects were displayed on a reflective SLM, and the light propagated in free space until intensity sampling by the camera. The ML engine of the end-to-end type [Fig. 10(a)] was of the convolutional DNN type with residuals. Training was carried out by drawing objects from standard databases, Faces-LFW, and ImageNet, converting each object's grayscale intensity to a phase signal in the range $[0, \pi)$, and then displaying that signal on the SLM. Because of the relatively large range of phase modulation, linearizing assumptions would have been invalid in this arrangement.

Retrieval of the complex amplitude, i.e., of both the magnitude and phase, of biological samples using ML in the digital holography (DH) arrangement was reported by Rivenson *et al.* [240]; see Fig. 14. The samples used in the experiments were from breast tissue, Papanicolaou (Pap) smears, and blood smears. In this case, the ML engine used a single-pass physics-informed preprocessor, as in Fig. 10(d), with the approximant H^* implemented as the (optical) backpropagation algorithm. The DNN was of the convolutional type. Training was carried out using

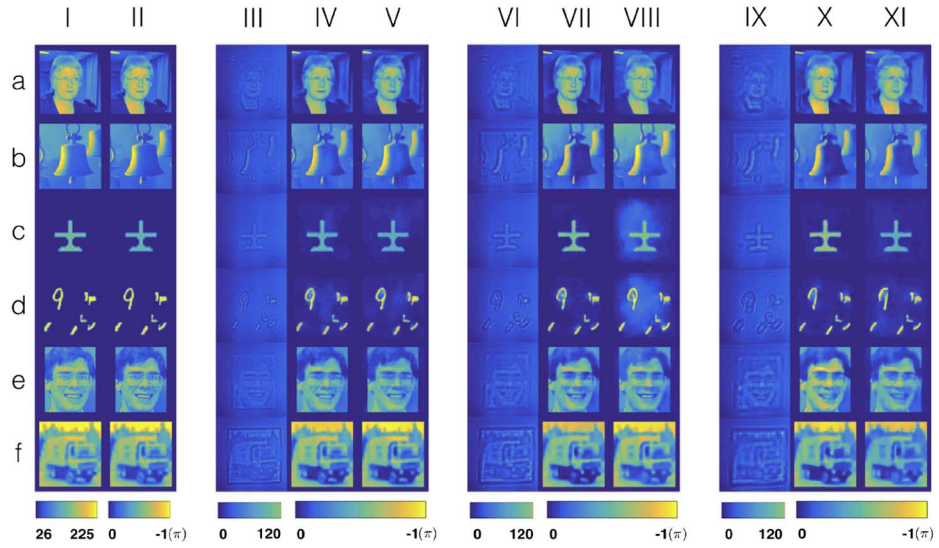

Fig. 13. Phase extraction neural network (PhENN) (reprinted from [234], Fig. 5, with permission). Columns I and II are the ground truth pixel values driving the SLM and corresponding phase images f produced by the SLM, according to calibration. Groups of columns III-V, VI-VIII, and IX-XI are different propagation distances $z = 37.5$ cm, 67.5 cm, and 97.5 cm, respectively. Columns III, VI, IX are the raw images g captured by the camera. Columns IV, VII, X are reconstructions $\hat{f}$ when PhENN was trained with Faces-LFW [235], whereas columns V, VIII, XI are estimates $\hat{f}$ when PhENN was trained with ImageNet [173]. The rows represent different databases the PhENN was tested on as a, Faces-LFW (disjoint from training); b, ImageNet (disjoint from training); c, Characters [236]; d, MNIST [237]; e, Faces-ATT [238]; f, CIFAR [239].

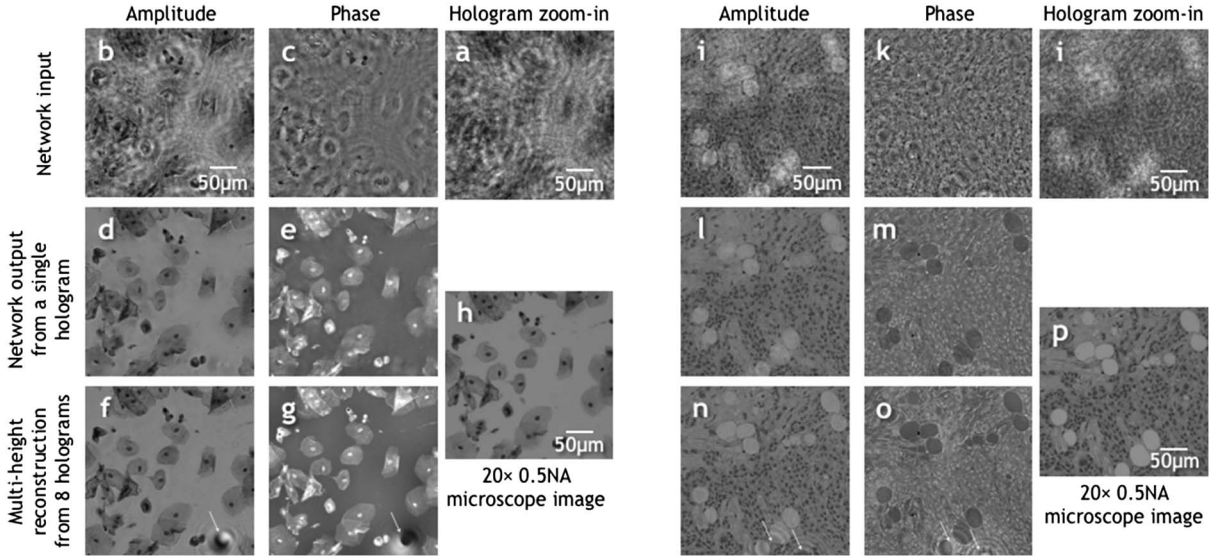

Fig. 14. Retrieval of phase and amplitude with a physics-informed ML engine (reprinted with permission from Springer [240], Fig. 2). (a)–(h) Pap smear and (j)–(p) breast tissue section reconstructions. (a), (i) Zoom-ins to the (optical) backpropagation results from a single intensity image; (h), (p) corresponding bright-field microscopy images, shown for comparison. (b), (c), (j), (k) (Optical) backpropagation results showing artifacts due to twin image and self-interference effects; (d), (e), (l), (m) corresponding ML engine reconstructions from a single hologram, showing quality comparable to (f), (g), (n), (o) traditional reconstructions from eight holograms.

up to eight holograms to produce accurate estimates of the samples' phase profiles. After training, the ML engine was able, with a single hologram input, to match imaging quality, in terms of SSIM (Section 3.E) of traditional algorithms that would have required two to three times as many holograms, and was faster as well by a factor of three to four times.

The 3D image formation properties of phase retrieval have also been investigated as depth prediction with robust automatic focusing [241,242], image reconstruction with a single-shot in-line hologram [243], extended depth of field [244,245], and transparent 3D sample reconstruction from diffraction images obtained at multiple angles (phase tomography) [246]. In the results shown in Fig. 15 from some early work [244], the single-pass physics-assisted ML engine was trained with physical objects at various depths, and out-performed both single-height and multi-height (optical) back propagation in terms of depth invariance. Resolution improvements in digital holographic microscopy were obtained through aberration correction and background rejection [247] instead of the compressive approach in [79]. Adversarial payoff [165] was recently used in holographic microscopy with resolution below the nominal pixel size [248] and ptychography 249]] using SSIM and MAE, respectively, as TLFs.

In the same context as strongly scattering phase objects, ML was used early on in an intriguing alternative to our general architecture in Fig. 3 where the object's 3D refractive index distribution, instead of output by the ML engine, is *stored as the weights* in a multilayered deep network [250,251]. This is possible because of the formal analogy between the beam-propagation method [252] and the NN back-propagation weight training method, and is implemented by training the DNN to match the illumination fields from multiple angles as input and the corresponding raw intensity images as outputs. This technique recently inspired a similar solution to the simpler problem of a thin phase object with phase delays stored in a single layer and raw intensity measurements obtained according to the Fourier ptychography scheme [253].

The problem of lateral resolution in phase retrieval algorithms is interesting, because performance depends on the optics as well as the inverse algorithm. In regularized algorithms, e.g., Gerchberg–Saxton, gradient descent, or compressed holography,

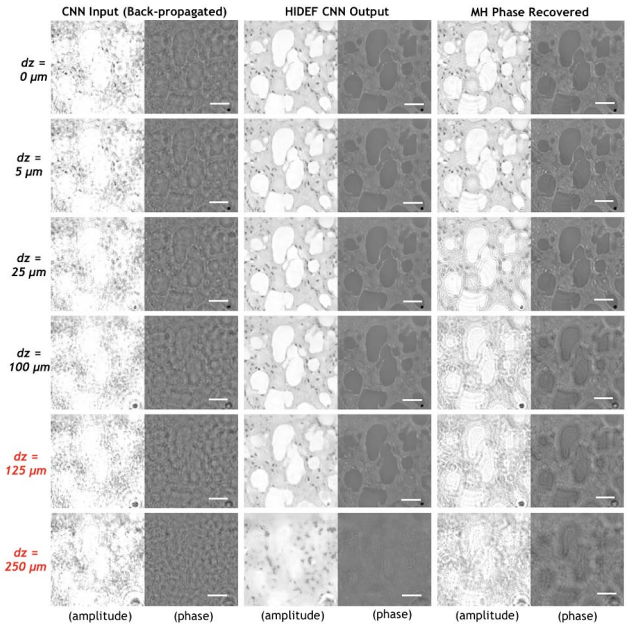

Fig. 15. Physics-informed DNN trained with samples at different propagation distances of up to ±100 μm from the focal plane, exhibiting improved depth invariance compared to (optical) backpropagation and multi-height phase retrieval (MH-PR) (reprinted with permission [244], Fig. 3). The results are from a human breast tissue sample, captured at different propagation distances dz. Distances marked in red exceed the training range of the DNN. Scale bar = 20 μm.

the effectiveness of the computation depends on how faithful the prior is to the relevant class of phase objects. Similarly, in ML-based phase retrieval, one would expect qualities of the example database, such as the spatial power spectral density (PSD), to have a similar effect. In fact, due to the nonlinearities of the training and reconstruction processes, spatial frequencies exhibiting higher prevalence in the training database may become so dominant that they inhibit successful reconstruction of less prevalent spatial frequencies. Invariably, lower spatial frequencies are more popular, especially in datasets of natural images due to the well-known inverse-square PSD law [254]. The result is that the ML-reconstructed phase estimates are strongly low-pass filtered [156]. One way around that problem is to violate the training prior, e.g., by spectrally pre-modulating the examples to flatten the PSD. This results in resolution improvement, at the expense of some edge-enhancement artifacts, as shown in Fig. 16. Alternatively, the low- and high-frequency bands can be processed by two separate NNs [255] and recombined by a third NN, all three networks trained separately [256]. The idea can be thought of as a computational analogue to the HiLo super-resolution method [46,47,49,50], and in [256], it was applied to both super-resolution and phase retrieval.

C. Imaging of Dark Scenes

The challenges associated with super-resolution and phase retrieval become much exacerbated when the photon budget is tight or other sources of noise are strong. This is because deconvolutions, in general, tend to amplify noise artifacts [5]. In standard photography, histogram equalization and gamma correction are automatically applied by modern high-end digital cameras and even in smartphones; however, "grainy" images and color distortion still occur. In more challenging situations, a variety of more sophisticated denoising algorithms utilizing compressed sensing and local feature representations have been investigated and benchmarked [257–262]. What these algorithms exploit, with varying success, is that natural images are characterized by the prior of strong correlation structure, which should persist even under noise fluctuations that much exceed the signal. Understood in this sense, ML presents itself as an attractive option to learn the correlation structures and then recover high-resolution content from the noisy raw images.

The first use of a CNN for monochrome Poisson denoising, to our knowledge, was by Remez *et al.* [263]. More recently, a convolutional network of the U-net type was trained to operate on all three color channels under illumination and exposure conditions that, to the naked eye, make the raw images appear entirely dark while histogram- and gamma-corrected reconstructions are severely color distorted [169]; see Fig. 17. The authors created a see-in-the-dark (SID) dataset of short-exposure images, coupled with their respective long-exposure images, for training and testing; and used Amazon's Mechanical Turk platform for perceptual image evaluation by humans [168]. They also report that, unlike other related works, neither skip connections in U-net nor generative adversarial training led to any improvement in their reconstructions.

ML has also been shown to be effective for phase retrieval from intensity data with extremely low photon counts [265]. The recurrent and cascaded engines in Figs. 10(b) and 10(c) are not, strictly speaking, applicable because this inverse problem is nonlinear. An end-to-end DNN, as in Fig. 10(a), yielded acceptable results only with moderate photon flux. For lower flux, the single-pass engine in Fig. 10(d) was used. The approximant H^* was obtained by running the standard Gerchberg–Saxton algorithm up to a single iteration. With noisy data, as expected, the estimates $\hat{f}^{[0]}$ produced by H^* were rather poor, as seen in Figs. 18(c) and 18(g); yet with this as input, the DNN was capable of significantly improving its output; compare Figs. 18(d), 18(h) and 18(e), 18(i), respectively. It is also worth noting that running the Gerchberg–Saxton algorithm to more iterations produced discernible improvements neither to its own reconstructions nor as an alternative approximant to the DNN.

A radically different approach to imaging under extremely low light flux is computational ghost imaging. Interferometric ghost imaging originated in quantum optics [266], but it was soon discovered theoretically [267,268] and experimentally [269,270] that it has a classical analogue with pseudo-thermal light. The raw intensity image is obtained not by a camera but by a spatially

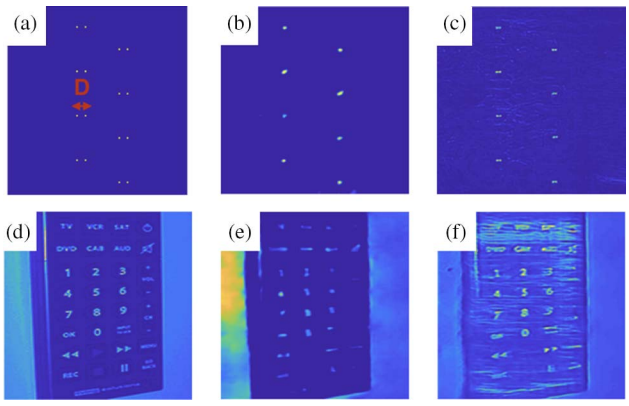

Fig. 16. Resolution improvement by spectral premodulation of PhENN (from [156], Figs. 5, 8, and 9; reprinted with permission). (a) Two-phase point-object resolution target, implemented on the SLM; (b) PhENN image, not resolving phase point-objects separated by three pixels; (c) with spectral pre-modulation, the three pixel-separated phase-point objects are resolved; (d) sample test image from the ImageNet database; (e) PheNN reconstruction; (f) PhENN reconstruction with spectral pre-modulation, showing sharper features but also edge-enhancement artifacts.

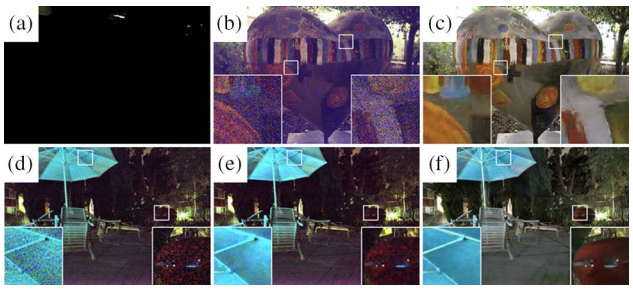

Fig. 17. ML applied to reconstruct a severely photon-limited scene (after [169], Fig. 5; reprinted with permission). (a) Signal captured by a Fujifilm X-T2 camera with ISO 800, aperture f/7.1, and exposure of 1/30 s, with illuminance of approximately 1 lux; (b) reconstruction using denoising, deblurring, and enhancement, which the authors refer to as "traditional image processing pipeline"; (d) reconstruction obtained using the radiational image processing pipeline on a Sony α7SII camera; (e) reconstruction obtained using BM3D [264], considered as benchmark for denoising; (c), (f) ML engine reconstructions of the respective dark images.

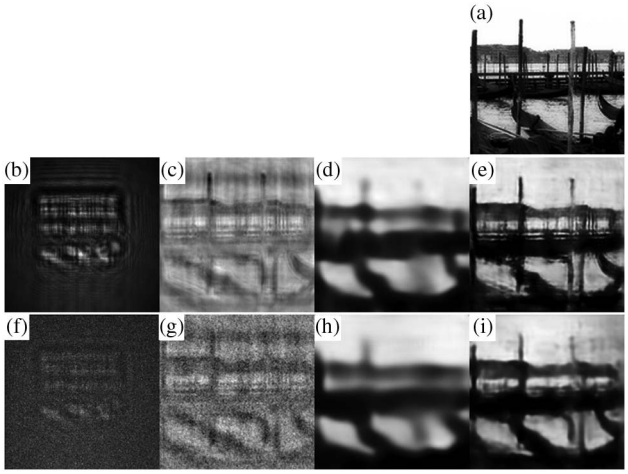

Fig. 18. ML applied to quantitative phase retrieval on a severely photon-limited signal (after [265], Fig. 2; reprinted with permission). (a) Ground truth f for a test example; (b), (f) raw intensity signal at photon flux of 1,050 photons/pixel; (c), (g) corresponding outputs $\hat{f}^{[0]}$ of the approximant H^* of Fig. 10(d), implemented as a single-iteration Gerchberg–Saxton algorithm; (d), (h) corresponding reconstructions by the end-to-end ML engine of Fig. 10(a); (e), (i) corresponding reconstructions by physics-informed single-pass ML engine according to Fig. 10(d). In (a) and reconstructions (c)–(e, (g)–(i), grayscale tone represents phase delay $[0,2\pi)$.

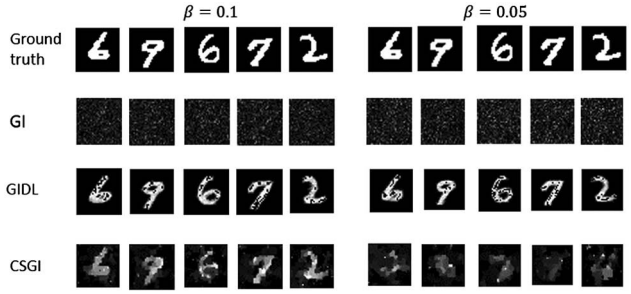

Fig. 19. Experimental computational ghost image reconstructions with sampling ratio $\beta = 0.1$ (left-hand side group) and $\beta = 0.05$ (right-hand side group) (after [279]; reprinted with permission from Springer Nature). Top row: ground truth; second-row: basic ghost reconstruction according to [273]; third row: deep learning ghost reconstructions using the images in the second row as approximants $\hat{f}^{[0]}$ to the physics-informed ML engine of Fig. 10(d); last row: compressive ghost reconstructions according to [278].

integrating (bucket) detector on one arm of the interferometer, while an SLM displays random phase patterns, and a pinhole detector is scanned laterally on the other arm. The image is obtained by correlating the two measurements and removing a background term. Following a dispute about the explanation of pseudo-thermal ghost imaging [271,272], it was shown that in the classical case, one arm of the interferometer can be removed and replaced by computation [273]. Because the bucket detector collects all available photons, ghost imaging methods are attractive for applications where the available illumination is extremely weak, much less than a few photons per pixel [274] and in harsh environments [275]. However, to reconstruct images with a good signal-to-noise-ratio (SNR), usually a long acquisition time is needed [276]. Therefore, the sampling ratio

$$\beta = \frac{\#\text{displayed SLM patterns}}{\#\text{reconstructed pixels}} \tag{35}$$

needs to be kept relatively low. Compressive priors have also been used with some success to improve the condition of the ghost inverse problem [277,278].

Lyu *et al.* [279] used deep learning with the single-pass physics-informed engine [Fig. 10(d)] and approximant H^* computed according to the original computational ghost imaging [273]. Due to the low sampling rate and the noisy nature of the raw measurements, the approximant reconstructions $\hat{f}^{[0]}$ were corrupted and unrecognizable. However, when these $\hat{f}^{[0]}$ were used as input to the DNN, high-quality final estimates $\hat{f}$ were obtained even with sampling rates β as low as 5%, as shown in Fig. 19.

D. Imaging in the Presence of Strong Scattering

Imaging through diffuse media [280,281] is a classical challenging inverse problem with significant practical applications ranging from non-invasive medical imaging through tissue to autonomous navigation of vehicles in foggy conditions. The noisy statistical inverse model formulation (2) must now be reinterpreted with the forward operator H itself becoming random. When f is the index of refraction of the strongly scattering medium itself, then H is also nonlinear. Not surprisingly, this topic has attracted considerable attention in the literature, with most attempts generally belonging to one of two categories. The first is to characterize the diffuse medium H, assuming it is accessible and static, through (incomplete) measurement of the operator H, which in this context is referred to as transmission matrix [282–284]. The alternative is to characterize statistical similarities between *moments* of H. The second-order moment, or speckle correlations, are known as the memory effect. The idea originated in the context of electron propagation in disordered conductors [285] and of course is also valid for the analogous problem of optical disordered media [286–290].

The first-ever, to our knowledge, use of a non-dictionary (Section 2.B) ML algorithm for imaging through diffusers was by Horisaki *et al.* [18]. The authors created an experimental configuration explicitly non-suited to the transmission matrix formulation by sandwiching two SLMs, both displaying the same image, between three acrylic diffuser plates. The optical system was illuminated coherently, and it was lensless, i.e., without imaging optics on the beam path. Examples for training were drawn from the Caltech Computer Vision Database [291]. The ML engine was not implemented as a DNN, but rather as an extension of support vector machines [292] known as support vector regression (SVR) [293].

In SVR, the image is expressed in linear form as

$$\hat{f} = \sum_{n=1}^{N} \gamma_n \mathcal{K}(g_{(n)}, g) + f_0, \tag{36}$$

where $\mathcal{K}(\cdot,\cdot)$ is the kernel function [294,295], f_0 is a bias vector obtained by an efficient form of sequential minimal optimization [296], and the coefficients γ_n are the Lagrange multipliers in a convex optimization problem [297] involving the ε-insensitive TLF:

$$\mathcal{E}_{\varepsilon\text{-ins}}(v,\tilde{v}) = \begin{cases} 0, & \text{if } |v-\tilde{v}| \le \varepsilon, \\ |v-\tilde{v}| - \varepsilon, & \text{otherwise.} \end{cases} \quad (37)$$

This metric is also known as a soft margin TLF. An early proposal for kernel functions was polynomials [298]. Horisaki *et al.* [18] instead used the shift-invariant Gaussian radial basis function kernel. The results are shown in Figs. 20 and 21. The SVR engine trained with faces does a good job reconstructing test objects that are faces also. However, when the test objects are not faces, the SVR still "hallucinates" face reconstructions.

Deep learning solutions to the problem were first presented in [299] and [155], using end-to-end fully connected and residual-convolutional (CNN) architectures, respectively. Results are shown in Figs. 22 and 23. The fully connected solution [299] is motivated by the physical fact that when light propagates through a strongly scattering medium, every object pixel influences every raw image pixel in shift non-invariant fashion. However, the large number of connections creates risks of undertraining and overfitting, and limits the space-bandwidth product (SBP) of the reconstructions due to limited computational resources. On the other hand, the CNN trained with NPCC loss function [155,300], despite being designed for situations when limited range of influence and shift invariance constraints are valid, Section 3.D, does a surprisingly good job at learning shift variance—through the ReLU nonlinearities and pooling operations, presumably—and achieves larger SBP. Both methods work well with spatially sparse datasets, e.g., handwritten numerical digits, and Latin and Chinese characters. Compared to Horisaki *et al.* [18], the deep architectures perform comparably well with spatially dense datasets of restricted content, e.g., faces, and also hallucinate when tested outside their learned priors.

Approaches [18,155,299,300] are all meant to be used with the specific diffuser they were trained with. It was recently shown that for spatially sparse datasets, it is possible to train a single DNN to image through arbitrary diffusers belonging to a class of similar statistics, by adopting a cross-entropy TLF (27), Section 3.E, and with the training procedure including samples

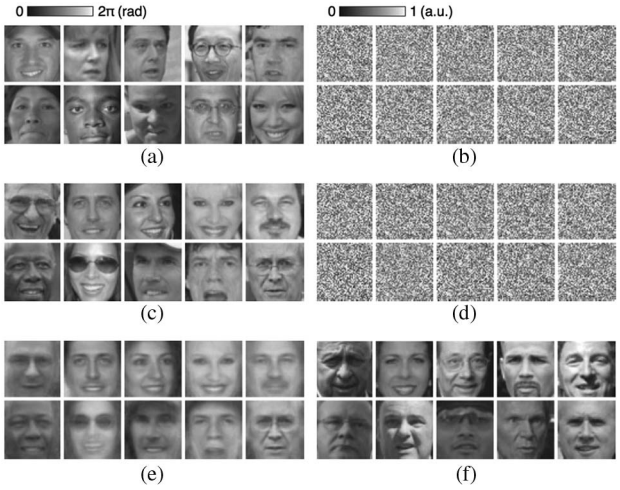

Fig. 20. Learning-based imaging through diffusers using support vector regression (SVR) (after [18], Fig. 3; reprinted with permission). (a) Training examples from a database of faces and (b) their corresponding speckle patterns; (c) test face examples and their corresponding (d) speckle patterns, (e) test reconstructions using SVR, and (f) test reconstructions using pattern matching.

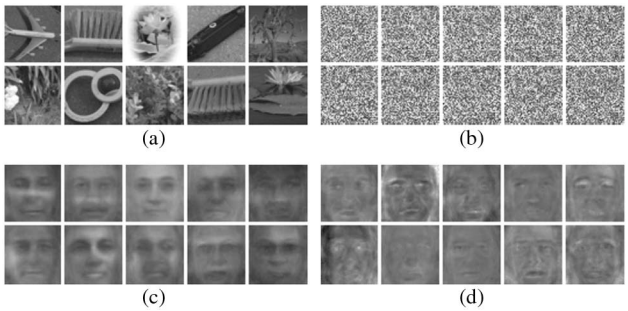

Fig. 21. Generalization study of the SVR method (after [18], Fig. 5; reprinted with permission). (a) Test non-face examples and their corresponding (b) speckle patterns, (c) test reconstructions using SVR, and (d) test reconstructions using pattern matching.

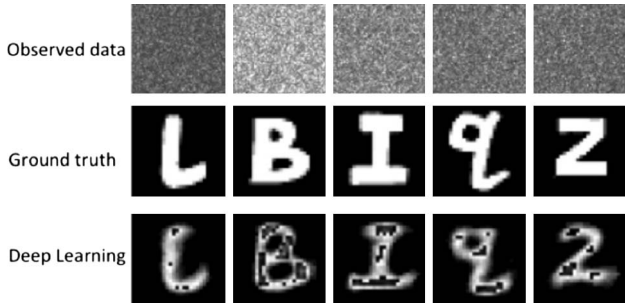

Fig. 22. Learning-based imaging through optically thick diffusers using a fully connected DNN with MAE TLF (after [299], Fig. 3; reprinted with permission). First row: speckle patterns input to the DNN; second row: corresponding ground-truth images selected among a database of English letters; third row: corresponding reconstructions by the DNN.

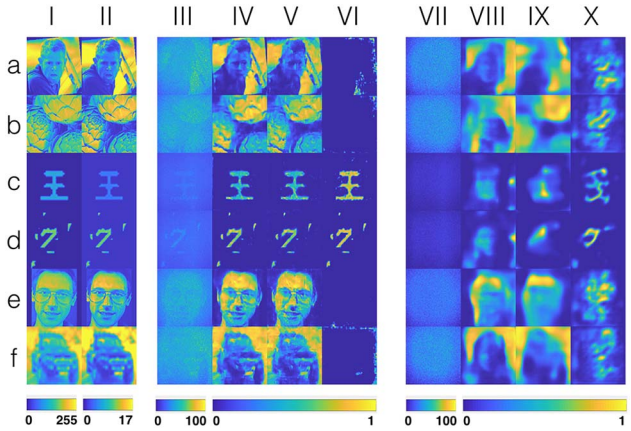

Fig. 23. Learning-based imaging through diffusers using IDiffNet: a residual-convolutional DNN with NPCC TLF (after [155], Fig. 6; reprinted with permission). Columns I, II; ground truth pixel values and fields modulated by the SLM, after calibration; III-VI: results with 600-grit diffuser. III are the raw images, IV-VI test reconstructions from IDiffNet trained with Faces-LFW [235], ImageNet [173], and MNIST [237] datasets, respectively. Rows correspond to the dataset the test image is drawn from, as (a) Faces-LFW, (b) ImageNet, (c) Characters [236], (d) MNIST, (e) Faces-ATT [238], (f) CIFAR [239].

of the same training object imaged with randomly chosen diffusers from among the class [163]. It has also been shown that imaging through multi-core fibers can be relatively tolerant to variability in the transmission matrix, in terms of both image reconstruction quality and recognition of simple patterns, e.g., characters from the MNIST database [23].

Non-line-of-sight (NLOS) imaging, recognition, and tracking belong to a related class of problems, because capturing details about objects in such cases must rely on scattering, typically of light pulses [301–309] or spatially incoherent light [310–313]. Convolutional DNNs have been found to be useful for improving gesture classification [314], and person identification and three-dimensional localization [315]; in the latter case even with a single-photon, single-pixel detector only.

5. CONCLUDING REMARKS

The diverse collection of ML flavors adopted and problems tackled by the CI community in a relatively brief time period, mostly since ∼2010 [104], indicate that the basic idea of doing at least partially the job of Tikhonov–Wiener optimization by DNN holds much promise. A significant increase in the rate of related publications is evident—we had trouble keeping up while crafting the present review—and is likely to accelerate, at least in the near future. As we saw in Section 4, in many cases, ML algorithms have been discovered to offer new insights or substantial performance improvements on previous CI approaches, mostly compressive sensing based, whereas in other cases, particular challenges associated with acute CI problems have prompted innovations in ML architectures themselves. This productive interplay is likely to benefit both disciplines in the long run, especially because of the strong connection they share through optimization theory and practice.

Beyond these bidirectional improvements, we wish to single out and return to the open question of image interpretation versus image formation that we touched upon in Section 1. It is possible that image-forming instruments aided by ML will continue doing just that, producing high-quality images for consumption by humans and image interpretation algorithms that require high-quality images as their input. It is also possible that, to some degree, image formation and interpretation will fuse, resulting in algorithms that interpret the raw data directly, without bothering to form high-quality images; or other algorithms might turn out to be useful by falling somewhere in between these two extremes. We saw already that DH and imaging through multimode fiber bundles have emerged as fertile ground for this kind of crossover [19–23]. In our view, further developments in that direction will carry significant fundamental value, as existing algorithms and insights, originating as they have from either side of formation or interpretation, will likely have to be at least partially reconsidered. Practical implications are also to be expected if pattern recognition and image interpretation, rather than relying mostly on human-recognizable optical intensity patterns as it does presently, become more directly linked to the complete electromagnetic representation of the optical field and its interaction with matter, i.e., the broadest definition of CI.

Funding. Intelligence Advanced Research Projects Activity (IARPA) (FA8650-17-C-9113); Chinese Academy of Sciences (CAS) (QYZDB-SSW-JSC002); Chinesisch-Deutsche Zentrum für Wissenschaftsförderung (CDZ) (GZ1931); National Research Foundation Singapore (NRF) (SMART Centre).

Acknowledgment. The authors are grateful to Zhang Zhengyun, Alexandre Goy, Berthold K. P. Horn, and Lei Tian for helpful discussions and extensive comments on earlier versions of the paper.

REFERENCES

1. N. Wiener and E. Hopf, "Uber eine Klasse singulaerer Integralgeichungen," Sitzungsber. Preuss. Akad. Math.-Phys. Kl. **31**, 696–706 (1931).
2. N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series* (Wiley, 1949).
3. A. N. Tikhonov, "On the solution of ill-posed problems and the method of regularization," Dokl. Akad. Nauk SSSR **151**, 501–504 (1963).
4. A. N. Tikhonov, "On the regularization of ill-posed problems," Dokl. Akad. Nauk SSSR **153**, 49–52 (1963).
5. M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging* (Institute of Physics, 1998).
6. D. J. Brady, *Optical Imaging and Spectroscopy* (Wiley, 2009).
7. J. Mait, G. W. Euliss, and R. A. Athale, "Computational imaging," Adv. Opt. Photon. **10**, 409–483 (2018).
8. U. Grenander, *General Pattern Theory: A Mathematical Study of General Structures* (Clarendon, 1994).
9. M. L. Minsky, "Neural nets and the brain-model problem," Ph.D. thesis (Princeton University, 1954).
10. S. Agmon, "The relaxation method for linear inequalities," Can. J. Math. **6**, 382–392 (1954).
11. F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," Psychol. Rev. **65**, 386–408 (1958).
12. H. D. Block, "The perceptron: a model for brain functioning," Rev. Mod. Phys. **34**, 123–135 (1962).
13. M. L. Minsky and S. Papert, *Perceptrons* (MIT, 1969).
14. G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural Comput. **18**, 1527–1554 (2006).
15. Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Neural Information Processing Systems (NIPS)* (2006), Vol. **19**, pp. 153–160.
16. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature **521**, 436–444 (2015).
17. J. Schmidhuber, "Deep learning in neural networks: an overview," Neural Netw. **61**, 85–117 (2015).
18. R. Horisaki, R. Takagi, and J. Tanida, "Learning-based imaging through scattering media," Opt. Express **24**, 13738–13743 (2016).
19. A. Yevick, M. Hannel, and D. G. Grier, "Machine-learning approach to holographic particle characterization," Opt. Express **22**, 26884–26890 (2014).
20. M. D. Hannel, A. Abdulali, M. O'Brien, and D. G. Grier, "Machine-learning techniques for fast and accurate feature localization of holograms of colloidal particles," Opt. Express **26**, 15221–15231 (2018).
21. J. Yoon, Y.-J. Jo, M.-H. Kim, S. Y. Lee, S.-J. Kang, and Y. K. Park, "Identification of non-activated lymphocytes using three-dimensional refractive index tomography and machine learning," Sci. Rep. **7**, 6654 (2017).
22. R. Horstmeyer, R. Y. Chen, B. Kappes, and B. Judkewitz, "Convolutional neural networks that teach microscopes how to image," arXiv:1709.07223 (2017).
23. N. Borhani, E. Kakkava, C. Moser, and D. Psaltis, "Learning to see through multimode fibers," Optica **5**, 960–966 (2018).
24. S. Wang, Z. Su, L. Ying, X. Peng, S. Zhu, F. Liang, D. Feng, and D. Liang, "Accelerating magnetic resonance imaging via deep learning," in *IEEE 13th International Symposium on Biomedical Imaging (ISBI)* (2016), pp. 514–517.
25. K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," IEEE Trans. Image Process. **26**, 4509–4522 (2017).
26. D. Lee, J. Yoo, and J. C. Ye, "Deep residual learning for compressed sensing MRI," in *IEEE 14th International Symposium on Biomedical Imaging (ISBI)* (2017), pp. 15–18.

27. B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, "Image reconstruction by domain-transform manifold learning," Nature **555**, 487–492 (2018).
28. M. T. McCann, K. H. Jin, and M. Unser, "Convolutional neural networks for inverse problems in imaging," IEEE Signal Process. Mag. **34**(6), 85–95 (2017).
29. A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos, "Using deep neural networks for inverse problems in imaging," IEEE Signal Process. Mag. **35**(1), 20–36 (2018).
30. Y.-J. Jo, H. Cho, S. Y. Lee, G. Choi, G. Kim, H.-S. Min, and Y.-K. Park, "Quantitative phase imaging and artificial intelligence: a review," IEEE J. Sel. Top. Quantum Electron. **25**, 6800914 (2019).
31. A. Papoulis and M. S. Bertran, "Digital filtering and prolate functions," IEEE Trans. Circuit Theory **19**, 674–681 (1972).
32. F. Gori and G. Guattari, "Degrees of freedom of images from point-like element pupils," J. Opt. Soc. Am. **64**, 453–458 (1974).
33. D. Slepian, "Prolate spheroidal wave functions, Fourier analysis, and uncertainty—V: the discrete case," Bell Syst. Tech. J. **57**, 1371–1430 (1978).
34. M. Minsky, "Microscopy apparatus," U.S. patent 3,013,467 (Dec 19, 1961).
35. W. Lukosz and M. Marchand, "Optischen Abbidung unter Uberschreitung der Beugungsbedingten Aufl osungsgrenze," Opt. Acta **10**, 241–255 (1963).
36. C. J. R. Sheppard and A. Choudhury, "Image formation in the scanning microscope," Opt. Acta **24**, 1051–1073 (1977).
37. C. J. R. Sheppard and T. Wilson, "Fourier imaging of phase information in scanning and conventional microscopes," Philos. Trans. R. Soc. London **A295**, 513–536 (1980).
38. C. J. R. Sheppard and C. J. Cogswell, "Three-dimensional image formation in confocal microscopy," J. Microsc. **159**, 179–194 (1990).
39. M. A. A. Neil, R. Juskaitis, and T. Wilson, "Method of obtaining optical sectioning by using structured light in a conventional microscope," Opt. Lett. **22**, 1905–1907 (1997).
40. R. Heintzmann and C. G. Cremer, "Laterally modulated excitation microscopy: improvement of resolution by using a diffraction grating," Proc. SPIE **3568**, 185–196 (1999).
41. M. G. Gustafsson, "Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy," J. Microsc. **198**, 82–87 (2000).
42. E. Fuchs, J. S. Jaffe, R. A. Long, and F. Azam, "Thin laser light sheet microscope for microbial oceanography," Opt. Express **10**, 145–154 (2002).
43. J. Huisken, J. Swoger, F. Del Bene, J. Wittbrot, and E. H. Steltzer, "Optical sectioning deep inside live embryos by selective plane illumination microscopy," Science **305**, 1007–1009 (2004).
44. R. Heintzmann and P. A. Benedetti, "High-resolution image reconstruction in fluorescence microscopy with patterned excitation," Appl. Opt. **45**, 5037–5045 (2006).
45. P. J. Keller, A. Schmidt, J. Wittbrot, and E. H. Steltzer, "Reconstruction of zebrafish early embryonic development by scanning light sheet microscopy," Science **322**, 1065–1069 (2008).
46. D. Lim, K. K. Chu, and J. Mertz, "Wide-field fluorescence sectioning with hybrid speckle and uniform-illumination microscopy," Opt. Lett. **33**, 1819–1821 (2008).
47. S. Santos, K. K. Chu, D. Lim, N. Bozinovic, T. N. Ford, C. Hourtoule, A. Bartoo, S. K. Singh, and J. Mertz, "Optically sectioned fluorescence endomicroscopy with hybrid-illumination imaging through a flexible fiber bundle," J. Biomed. Opt. **14**, 030502 (2009).
48. P. A. Santi, S. B. Johnson, M. Hillenbrand, P. Z. GrandPre, T. J. Glass, and J. R. Leger, "Thin-sheet laser imaging microscopy for optical sectioning of thick tissues," Biotechniques **46**, 287–294 (2009).
49. J. Mertz and J. Kim, "Scanning light-sheet microscopy in the whole mouse brain with Hilo background rejection," J. Biomed. Opt. **15**, 016027 (2010).
50. J. Mertz, "Optical sectioning microscopy with planar or structured illumination," Nat. Methods **8**, 811–819 (2011).
51. R. Heintzmann and T. Huser, "Super-resolution structured illumination microscopy," Chem. Rev. **117**, 13890–13908 (2017).
52. L. Tian, X. Li, K. Ramchandran, and L. Waller, "Multiplexed coded illumination for Fourier ptychography with an LED array microscope," Biomed. Opt. Express **5**, 2376–2389 (2014).
53. Y. Yunhui, A. Shanker, L. Tian, L. Waller, and G. Barbastathis, "Low-noise phase imaging by hybrid uniform and structured illumination transport of intensity equation," Opt. Express **22**, 26696–26711 (2014).
54. D. J. Brady, A. Mrozack, K. MacCabe, and P. Llull, "Compressive tomography," Adv. Opt. Photon. **7**, 756–813 (2015).
55. P. Llull, X. Liao, X. Yuan, J. Yang, D. Kittle, L. Carin, G. Sapiro, and D. J. Brady, "Coded aperture compressive temporal imaging," Opt. Express **21**, 10526–10545 (2013).
56. G. R. Arce, D. J. Brady, L. Carin, H. Arguello, and D. S. Kittle, "Compressive coded aperture spectral imaging," IEEE Signal Process. Mag. **31**(1), 105–115 (2014).
57. M. Hassan, J. A. Greenberg, I. Odinaka, and D. J. Brady, "Snapshot fan beam coded aperture coherent scatter tomography," Opt. Express **24**, 18277–18289 (2016).
58. M. J. Cieslak, K. A. A. Gamage, and R. Glover, "Coded-aperture imaging systems: past, present and future development–a review," Radiat. Meas. **92**, 59–71 (2016).
59. R. Penrose, "A generalized inverse for matrices," Math. Proc. Cambridge Philos. Soc. **51**, 406–413 (1955).
60. R. Penrose, "On best approximate solutions of linear matrix equations," Math. Proc. Cambridge Philos. Soc. **52**, 17–19 (1956).
61. E. Candès and T. Tao, "Decoding by linear programming," IEEE Trans. Inform. Theory **51**, 4203–4215 (2005).
62. E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete Fourier information," IEEE Trans. Inform. Theory **52**, 489–509 (2006).
63. D. L. Donoho, "Compressed sensing," IEEE Trans. Inform. Theory **52**, 1289–1306 (2006).
64. E. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," Commun. Pure Appl. Math. **59**, 1207–1223 (2006).
65. Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications* (Cambridge University, 2012).
66. E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multi-scale transforms," IEEE Trans. Inf. Theory **38**, 587–607 (1992).
67. D. L. Donoho and I. M. Johnstone, "Ideal denoising in an orthonormal basis chosen from a library of bases," C. R. Acad. Sci. **A319**, 1317–1322 (1994).
68. R. Coifman and D. L. Donoho, "Translation invariant denoising," in *Wavelets and Statistics, Lecture Notes in Statistics* (Springer-Verlag, 1995), Vol. **103**, pp. 120–150.
69. M. A. T. Figueiredo and R. Nowak, "An EM algorithm for wavelet-based image restoration," IEEE Trans. Image Proc. **12**, 906–916 (2003).
70. R. Chan, T. Chan, L. Shen, and Z. Shen, "Wavelet algorithms for high-resolution image reconstruction," SIAM J. Sci. Comput. **24**, 1408–1432 (2003).
71. M. W. Marcellin, M. J. Gormish, A. Bilgin, and M. P. Boliek, "An overview of JPEG-2000," in *Data Compression Conference* (2000), pp. 523–541.
72. P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," IEEE Trans. Pattern Anal. Mach. Intell. **12**, 629–639 (1990).
73. L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," Physica D **60**, 259–268 (1992).
74. J. Weikert, "A review of nonlinear diffusion filtering," in *Scale-Space Theory in Computer Vision*, B. ter Haar Romey, L. Florack, J. Koendrink, and M. Viergever, eds., Lecture Notes in Computer Science (Springer, 1997), pp. 3–38.
75. L. Tian, J. C. Petruccelli, and G. Barbastathis, "Nonlinear diffusion regularization for transport of intensity phase imaging," Opt. Lett. **37**, 4131–4133 (2012).
76. J. Cai, B. Dong, S. Osher, and Z. Shen, "Image restoration: total variation, wavelet frames, and beyond," J. Am. Math. Soc. **25**, 1033–1089 (2012).
77. E. Candès and T. Tao, "Near optimal signal recovery from random projections: universal encoding strategies?" IEEE Trans. Inform. Theory **52**, 5406–5425 (2006).
78. N. Dey, L. Blanc-Feraud, C. Zimmer, P. Roux, Z. Kam, J.-C. Olivo-Marin, and J. Zerubia, "Richardson-Lucy algorithm with total variation regularization for 3D confocal microscope deconvolution," Microsc. Res. Tech. **69**, 260–266 (2006).

79. W. Zhang, L. Cao, D. J. Brady, H. Zhang, J. Cang, H. Zhang, and G. Jin, "Twin-image-free holography: a compressive sensing approach," Phys. Rev. Lett. **121**, 093902 (2018).
80. Y. Liu, L. Tian, J. W. Lee, H. Y. H. Huang, M. S. Triantafyllou, and G. Barbastathis, "Scanning-free compressive holography for object localization with subpixel accuracy," Opt. Lett. **37**, 3357–3359 (2012).
81. Y. Liu, L. Tian, C.-H. Hsieh, and G. Barbastathis, "Compressive holographic two-dimensional localization with $1/30^2$ subpixel accuracy," Opt. Express **22**, 9774–9782 (2014).
82. L. Tian, J. C. Petruccelli, Q. Miao, H. Kudrolli, V. Nagarkar, and G. Barbastathis, "Compressive x-ray phase tomography based on the transport of intensity equation," Opt. Lett. **38**, 3418–3421 (2013).
83. B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," Nature **381**, 607–609 (1996).
84. B. A. Olshausen and D. J. Field, "Natural image statistics and efficient coding," Netw. Comput. Neural Syst. **7**, 333–339 (1996).
85. B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: a strategy employed by V1?" Vision Res. **37**, 3311–3325 (1997).
86. M. S. Lewicki and B. A. Olshausen, "A probabilistic framework for the adaptation and comparison of image codes," J. Opt. Soc. Am. **16**, 1587–1601 (1999).
87. M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," Neural Comput. **12**, 337–365 (2000).
88. M. Elad and M. Aharon, "Image denoising via learned dictionaries and sparse representation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE, 2006), Vol. **1**, pp. 895–900.
89. M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," IEEE Trans. Signal Process. **54**, 4311–4322 (2006).
90. R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," Proc. IEEE **98**, 1045–1057 (2010).
91. C. Bao, H. Ji, Y. Quan, and Z. Shen, "Dictionary learning and sparse coding: algorithms and convergence analysis," IEEE Trans. Patt. Anal. Mach. Intel. **38**, 1356–1369 (2016).
92. J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," IEEE Trans. Image Proc. **19**, 2861–2873 (2010).
93. J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," IEEE Trans. Image Proc. **21**, 3467–3478 (2012).
94. J. Bect, L. Blanc-Feraud, G. Aubert, and A. Chambolle, "A l_1-unified variational framework for image restoration," in *European Conference on Computer Vision (ECCV)* (2004), Vol. **3024**, pp. 1–13.
95. J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," Math. Program. **55**, 293–318 (1992).
96. R. Tibshirani, "Regression shrinkage and selection via the lasso," J. R. Stat. Soc. Ser. B **58**, 267–288 (1996).
97. I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," Commun. Pure Appl. Math. **57**, 1413–1457 (2004).
98. A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," SIAM J. Imag. Sci. **2**, 183–202 (2009).
99. J. M. Bioucas-Dias and M. A. Figueiredo, "A new twist: two-step iterative shrinkage/thresholding algorithms for image restoration," IEEE Trans. Image Process. **16**, 2992–3004 (2007).
100. D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithm for compressed sensing," Proc. Nat. Acad. Sci. USA **106**, 18914–18919 (2009).
101. Y. Li and S. Osher, "Coordinate descent optimization for l11 minimization with application to compressed sensing; a greedy algorithm," Inverse Probl. Imaging **3**, 487–503 (2009).
102. S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," Found. Trends Mach. Learn. **3**, 1–122 (2011).
103. D. P. Kingma and J. Lei Ba, "Adam: a method for stochastic optimization," in *International Conference on Learning Representations (ICLR)* (2015).
104. K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *27th International Conference on International Conference on Machine Learning (ICML)* (2010), pp. 399–406.
105. A. Mousavi, A. B. Patel, and R. G. Baraniuk, "A deep learning approach to structured signal recovery," in *53rd Annual Allerton Conference on Communication, Control, and Computing* (2015), pp. 1336–1343.
106. A. Mousavi and R. G. Baraniuk, "Learning to invert: signal recovery via deep convolutional networks," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (2017), pp. 2272–2276.
107. 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.
108. K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2009), pp. 2146–2153.
109. V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *International Conference on Machine Learning (ICML)* (2010), p. 432.
110. X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *14th International Conference on Artificial Intelligence and Statistics* (2011), pp. 315–323.
111. M. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds. (Curran Associates, 2017), pp. 700–708.
112. T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain correlations with generative adversarial networks," in *34th International Conference on Machine Learning* (2017), Vol. **70**, pp. 1857–1865.
113. J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 2223–2232.
114. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT, 2016).
115. D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," Nature **323**, 533–536 (1986).
116. L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification* (MIT, 1983).
117. A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations* (Springer-Verlag, 1990).
118. B. Delyon and A. Juditsky, "Accelerated stochastic approximation," SIAM J. Optim. **3**, 868–881 (1993).
119. L. Bottou, "Online algorithms and stochastic approximations," in *Online Learning and Neural Networks*, D. Saad, ed. (Cambridge University, 1998).
120. S. Ruder, "An overview of gradient descent optimization algorithms," arXiv:1609.04747 (2017).
121. 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. Mane, 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. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: large-scale machine learning on heterogeneous distributed systems," in *12th USENIX Conference on Operating Systems Design and Implementation (OSDI)* (2016), pp. 265–283.
122. LISA Lab, 2017, https://github.com/Theano/Theano.
123. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarma, and T. Darrell, "Caffe: convolutional architecture for fast feature embedding," in *22nd ACM International Conference on Multimedia* (ACM, 2014), pp. 675–678.
124. "Microsoft cognitive toolkit," https://github.com/Microsoft/cntk.
125. F. Chollet, "Keras: the Python deep learning library," 2015, https://keras.io.
126. J. Hertz, A. Krogh, and R. G. Palmer, "Introduction to the theory of neural computation," in *Santa Fe Institute Studies in the Sciences of Complexity* (Addison-Wesley, 1991).

127. C. M. Bishop, *Neural Networks for Pattern Recognition* (Clarendon, 1995).
128. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. (Wiley, 2001).
129. C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, 2006).
130. K. P. Murphy, *Machine Learning: A Probabilistic Perspective* (MIT, 2012).
131. C. C. Aggarwal, *Neural Networks and Deep Learning* (Springer, 2018).
132. E. Charniak, *Introduction to Deep Learning* (MIT, 2018).
133. A. N. Kolmogorov, "Three approaches to the quantitative definition of information," Int. J. Comput. Math. **2**, 157–168 (1968).
134. T. M. Cover and J. A. Thomas, *Information Theory* (Wiley, 1991), chap. 7, pp. 144–182.
135. V. N. Vapnik and A. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," Theory Prob. Appl. **16**, 264–280 (1971).
136. V. N. Vapnik, *Estimation of Dependences based on Empirial Data* (Springer-Verlag, 1982).
137. V. N. Vapnik, "Three fundamental concepts of the capacity of learning machines," Physica A **200**, 538–544 (1993).
138. Y. Abu-Mostafa, "The complexity of information extraction," IEEE Trans. Inf. Theory **32**, 513–525 (1986).
139. G. E. Hinton, "Learning translation invariant recognition in massively parallel networks," in *PARLE Conference on Parallel Architectures and Languages Europe* (Springer-Verlag, 1987), pp. 1–13.
140. M. C. Mozer and P. Smolensky, "Skeletonization: a technique for trimming the fat from a network via relevance assessment," in *Neural Information Processing Systems (NIPS)* (1989), Vol. **1**, pp. 107–115.
141. S. J. Hanson and L. Y. Pratt, "Comparing biases for minimal network construction with back-propagation," in *Neural Information Processing Systems (NIPS)* (1989), Vol. **1**, pp. 177–185.
142. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv:1207.0580 (2012).
143. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," J. Mach. Learn. Res. **15**, 1929–1958 (2014).
144. L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus, "Regularization of neural networks using DropConnect," in *30th International Conference on Machine Learning* (2013), Vol. **28**.
145. R. M. Neal, *Bayesian Learning for Neural Networks*, Lecture Notes in Statistics (Springer-Verlag, 1996), Vol. **118**.
146. Y. LeCun, "Generalization and network design strategies," Tech. Rep. CRG-TR-89-4 (University of Toronto, 1989).
147. Y. LeCun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, and W. Hubbard, "Handwritten digit recognition: applications of neural network chips and automatic learning," IEEE Commun. Mag. **27**(11), 41–46 (1989).
148. J. W. Goodman, *Introduction to Fourier Optics*, 2nd ed. (McGraw-Hill, 1996).
149. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 770–778.
150. K. Pearson, "Contributions to the mathematical theory of evolution. Note on reproductive selection," Proc. R. Soc. London **59**, 300–305 (1896).
151. J. L. Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," Am. Statist. **42**, 59–66 (1988).
152. E. K. Yen and R. G. Johnston, "The ineffectiveness of the correlation coefficient for image comparisons," Tech. Rep. LA-UR-96-2474 (Los Alamos National Laboratory, 1996).
153. A. M. Neto, L. Rittner, N. Leite, D. E. Zampieri, R. Lotufo, and A. Mendeleck, "Pearson's correlation coefficient for discarding redundant information in real time autonomous navigation systems," in *IEEE Multi-Conference on Autonomous Systems and Control (MSC)* (2007), pp. 426–431.
154. A. M. Neto, A. C. Victorino, I. Fantoni, D. E. Zampieri, J. V. Ferreira, and D. A. Lima, "Image processing using Pearson's correlation coefficient: applications on autonomous robotics," in *13th International Conference on Autonomous Robot Systems (Robotica)* (2013), pp. 14–19.
155. S. Li, M. Deng, J. Lee, A. Sinha, and G. Barbastathis, "Imaging through glass diffusers using densely connected convolutional networks," Optica **5**, 803–813 (2018).
156. 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**, 29340–29352 (2018).
157. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE Trans. Image Proc. **13**, 600–612 (2004).
158. T. M. Cover and J. A. Thomas, *Elements of Information Theory* (Wiley, 1991).
159. R. Y. Rubinstein, "Optimization of computer simulation models with rare events," Eur. J. Oper. Res. **99**, 89–112 (1997).
160. R. Y. Rubinstein, "The cross-entropy method for combinatorial and continuous optimization," Methodol. Comput. Appl. Probab. **1**, 127–190 (1999).
161. R. Y. Rubinstein, "Combinatorial optimization, cross-entropy, ants, and rare events," in *Stochastic Optimization: Algorithms and Applications*, S. Uryasev and P. M. Pardalos, eds. (Kluwer, 2001), pp. 304–358.
162. P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Rubinstein, "A tutorial on the cross-entropy method," Ann. Oper. Res. **134**, 19–67 (2005).
163. Y. Li, Y. Xue, and L. Tian, "Deep speckle correlation: a deep learning approach toward scalable imaging through scattering media," Optica **5**, 1181–1190 (2018).
164. T.-Y. Lin, P. Goyal, R. Girschik, K. He, and P. Dollár, "Focal loss for dense object detection," in *IEEE International Conference on Computer Vision* (2017), pp. 2999–3007.
165. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Neural Information Processing Systems (NIPS)* (2014), Vol. **27**.
166. A. M. Eskicioglu and P. S. Fisher, "Image quality measures and their performance," IEEE Trans. Commun. **43**, 2959–2965 (1995).
167. R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," arXiv:1801.03924 (2018).
168. Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *International Conference on Computer Vision (ICCV)* (2017), pp. 1511–1520.
169. C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to see in the dark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), pp. 3291–3300.
170. 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*, B. Leide, J. Matas, N. Sebe, and M. Welling, eds. (2016), vol. **9906**, pp. 694–711.
171. 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 4681–4690.
172. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR* (2015), p. 66.
173. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein, "ImageNet large scale visual recognition challenge," Int. J. Comput. Vis. **115**, 211–252 (2015).
174. A. Mahendran and A. Vebaldi, "Understanding deep image representations by inverting them," in *Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 5188–5196.
175. L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," arXiv:1508.06576 (2018).
176. M. Mardani, E. Gong, J. Y. Cheng, S. Vasanawala, G. Zaharchuk, M. Alley, N. Thakur, S. Han, W. Daly, J. M. Pauly, and L. Xing, "Deep generative adversarial networks for compressed sensing automates MRI," arXiv:1706.00051 (2017).
177. M. Mardani, H. Monajemi, V. Papyan, S. Vasanawala, D. Donoho, and J. Pauly, "Recurrent generative residual networks for proximal learning and automated compressive image recovery," arXiv:1711.10046 (2017).
178. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009), pp. 248–255.

179. G. B. Airy, "On the diffraction of an object-glass with circular aperture," Trans. Cambridge Philos. Soc. **5**, 283–291 (1834).
180. L. Rayleigh, "Investigations in optics, with special reference to the spectroscope," Philos. Mag. **8**(49), 261–274 (1879).
181. C.-Y. Yang, C. Ma, and M.-H. Yang, "Single-image super-resolution: a benchmark," in *European Conference on Computer Vision (ECCV)/Lecture Notes on Computer Science*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds. (2014), Vol. **8692**, pp. 372–386.
182. 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)/Lecture Notes on Computer Science Part IV* (2014), Vol. **8692**, pp. 184–199.
183. C. Dong, C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," IEEE Trans. Pattern Anal. Mach. Intel. **38**, 295–307 (2015).
184. J. Caballero, C. Ledig, A. Aitken, A. Acosta, J. Totz, Z. Wang, and W. Shi, "Real-time video super-resolution with spatio-temporal networks and motion compensation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 4778–4787.
185. C. J. Schuler, H. Christopher Burger, S. Harmeling, and B. Scholkopf, "A machine learning approach for non-blind image deconvolution," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2013).
186. A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009).
187. J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).
188. M. Sarikaya, "Evolution of resolution in microscopy," Ultramicroscopy **47**, 1–14 (1992).
189. W. E. Moerner and L. Kador, "Optical detection and spectroscopy of single molecules in a solid," Phys. Rev. Lett. **62**, 2535–2538 (1989).
190. S. W. Hell and J. Wichmann, "Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy," Opt. Lett. **19**, 780–782 (1994).
191. E. Betzig, "Proposed method for molecular optical imaging," Opt. Lett. **20**, 237–239 (1995).
192. R. M. Dickson, A. B. Cubitt, R. Y. Tsien, and W. E. Moerner, "On/off blinking and switching behaviour of single molecules of green fluorescent protein," Nature **388**, 355–358 (1997).
193. M. J. Rust, M. Bates, and X. Zhuang, "Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM)," Nat. Methods **3**, 793–796 (2006).
194. E. Betzig, G. H. Patterson, R. Sougrat, O. W. Lindwasser, S. Olenych, J. S. Bonifacino, M. W. Davidson, J. Lippincott-Schwarz, and H. F. Hess, "Imaging intracellular fluorescent proteins at nanometer resolution," Science **313**, 1642–1645 (2006).
195. S. T. Hess, T. P. Girirajan, and M. D. Mason, "Ultra-high resolution imaging by fluorescence photoactivation localization microscopy," Biophys. J. **91**, 4258–4272 (2006).
196. T. G. Stockham, T. M. Cannon, and R. B. Ingebretsen, "Blind deconvolution through digital signal processing," Proc. IEEE **63**, 678–692 (1975).
197. G. R. Ayers and J. C. Dainty, "Iterative blind deconvolution method and its applications," Opt. Lett. **13**, 547–549 (1988).
198. T. Kenig, Z. Kam, and A. Feuer, "Blind image deconvolution using machine learning for three-dimensional microscopy," IEEE Trans. Pattern Anal. Mach. Intel. **32**, 2191–2204 (2010).
199. Y. Rivenson, Z. Gorocs, H. Gunaydin, Y. Zhang, H. Wang, and A. Ozcan, "Deep learning microscopy," Optica **4**, 1437–1443 (2017).
200. H. Wang, Y. Rivenson, Z. Wei, H. Gunaydin, L. Bentolila, and A. Ozcan, "Deep learning achieves super-resolution in fluorescence microscopy," Nat. Methods (2018).
201. E. Nehme, L. E. Weiss, T. Michaeli, and Y. Shechtman, "Deep-STORM: super-resolution single-molecule microscopy by deep learning," Optica **5**, 458–464 (2018).
202. N. Boyd, E. Jonas, H. P. Babcock, and B. Recht, "DeepLoco: fast 3D localization microscopy using neural networks," bioRxiv.
203. R. W. Gerchberg and W. O. Saxton, "Phase determination from image and diffraction plane pictures in electron-microscope," Optik **34**, 275–284 (1971).
204. R. W. Gerchberg and W. O. Saxton, "Practical algorithm for the determination of phase from image and diffraction plane pictures," Optik **35**, 237–246 (1972).
205. J. R. Fienup, "Reconstruction of an object from the modulus of its Fourier transform," Opt. Lett. **3**, 27–29 (1978).
206. J. Fienup and C. Wackerman, "Phase-retrieval stagnation problems and solutions," J. Opt. Soc. Am. A **3**, 1897–1907 (1986).
207. 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**, 1334–1345 (2002).
208. V. Elser, "Phase retrieval by iterated projections," J. Opt. Soc. Am. A **20**, 40–55 (2003).
209. J. R. Fienup, "Phase retrieval algorithms: a comparison," Appl. Opt. **21**, 2758–2769 (1982).
210. M. R. Hestenes and E. Stiefel, "Method of conjugate gradients for solving linear systems," J. Res. Natl. Bur. Stand. **49**, 409–436 (1952).
211. 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).
212. G. Popescu, T. Ikeda, R. R. Dasari, and M. S. Feld, "Diffraction phase microscopy for quantifying cell structure and dynamics," Opt. Lett. **31**, 775–777 (2006).
213. 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).
214. 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).
215. M. R. Teague, "Deterministic phase retrieval: a Green's function solution," J. Opt. Soc. Am. **73**, 1434–1441 (1983).
216. N. Streibl, "Phase imaging by the transport-equation of intensity," Opt. Commun. **49**, 6–10 (1984).
217. J. W. Goodman and R. Lawrence, "Digital image formation from electronically detected holograms," Appl. Phys. Lett. **11**, 77–79 (1967).
218. W. Xu, M. H. Jericho, I. A. Meinertzhagen, and H. J. Kreuzer, "Digital in-line holography for biological applications," Proc. Nat. Acad. Sci. USA **98**, 11301–11305 (2001).
219. J. H. Milgram and W. Li, "Computational reconstruction of images from holograms," Appl. Opt. **41**, 853–864 (2002).
220. S. L. Friedman and J. M. Rodenburg, "Optical demonstration of a new principle of far-field microscopy," J. Phys. D **25**, 147–154 (1992).
221. B. C. McCallum and J. M. Rodenburg, "Two-dimensional demonstration of Wigner phase-retrieval microscopy in the STEM configuration," Ultramicroscopy **45**, 371–380 (1992).
222. J. M. Rodenburg and R. H. T. Bates, "The theory of super-resolution electron microscopy via Wigner-distribution deconvolution," Philos. Trans. R. Soc. London A **339**, 521–553 (1992).
223. A. M. Maiden and J. M. Rodenburg, "An improved ptychographical phase retrieval algorithm for diffractive imaging," Ultramicroscopy **109**, 1256–1262 (2009).
224. P. Li, T. B. Edo, and J. M. Rodenburg, "Ptychographic inversion via wigner distribution deconvolution: noise suppression and probe design," Ultramicroscopy **147**, 106–113 (2014).
225. G. Zheng, R. Horstmeyer, and C. Yang, "Wide-field, high-resolution Fourier ptychographic microscopy," Nat. Photonics **7**, 739–745 (2013).
226. X. Ou, R. Horstmeyer, and C. Yang, "Quantitative phase imaging via Fourier ptychographic microscopy," Opt. Lett. **38**, 4845–4848 (2013).
227. R. Horstmeyer, "A phase space model for Fourier ptychographic microscopy," Opt. Express **22**, 338–358 (2014).
228. D. J. Brady, K. Choi, D. L. Marks, R. Horisaki, and S. Lim, "Compressive holography," Opt. Express **17**, 13040–13049 (2009).
229. Y. Rivenson, A. Stern, and B. Javidi, "Compressive Fresnel holography," J. Disp. Technol. **6**, 506–509 (2010).
230. A. Pan, L. 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).
231. Y. Zhang, W. Jiang, L. Tian, L. Waller, and Q. Dai, "Self-learning based Fourier ptychographic microscopy," Opt. Express **23**, 18471–18486 (2015).

232. J. Lee and G. Barbastathis, "Denoised Wigner distribution deconvolution via low-rank matrix completion," Opt. Express **24**, 20069–20079 (2016).
233. A. Sinha, J. Lee, S. Li, and G. Barbastathis, "Lensless computational imaging through deep learning," arXiv:1702.08516 (2017).
234. A. Sinha, J. Lee, S. Li, and G. Barbastathis, "Lensless computational imaging through deep learning," Optica **4**, 1117–1125 (2017).
235. G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: a database for studying face recognition in unconstrained environments," Tech. Rep. 07-49 (University of Massachusetts, 2007).
236. C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, "CASIA online and offline Chinese handwriting databases," in *International Conference on Document Analysis and Recognition* (IEEE Computer Society, 2011), pp. 37–41.
237. Y. LeCun, C. Cortes, and C. J. Burges, "MNIST database of handwritten digits," AT&T Labs (2010), http://yann.lecun.com/exdb/mnist.
238. F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *2nd IEEE Workshop on Applications of Computer Vision* (IEEE, 1994), pp. 138–142.
239. A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep. TR-2009 (University of Toronto, 2009).
240. Y. Rivenson, Y. Zhang, H. Gunaydin, D. Teng, and A. Ozcan, "Phase recovery and holographic image reconstruction using deep learning in neural networks," Light Sci. Appl. **7**, 17141 (2018).
241. T. Shimobaba, T. Kakue, and T. Ito, "Convolutional neural network-based regression for depth prediction in digital holography," arXiv:1802.00664 (2018).
242. Z. Ren, Z. Xu, and E. Y. Lam, "Autofocusing in digital holography using deep learning," Proc. SPIE **10499**, 104991V (2018).
243. H. Wang, M. Lyu, and G. Situ, "Eholonet: a learning-based point-to-point approach for in-line digital holographic reconstruction," Opt. Express **26**, 22603–22614 (2018).
244. Y. Wu, Y. Rivenson, Y. Zhang, Z. Wei, H. Gunaydin, X. Lin, and A. Ozcan, "Extended depth-of-field in holographic image reconstruction using deep learning based auto-focusing and phase-recovery," Optica **5**, 704–710 (2018).
245. T. Pitkäaho, A. Manninen, and T. J. Naughton, "Performance of autofocus capability of deep convolutional neural networks in digital holographic microscopy," in *Digital Holography and Three-Dimensional Imaging* (OSA, 2017), paper W2A.5.
246. T. Nguyen, V. Bui, and G. Nehmetallah, "Computational optical tomography using 3-D deep convolutional neural networks," Opt. Eng. **57**, 043111 (2018).
247. T. Nguyen, V. Bui, V. Lam, C. B. Raub, L.-C. Chang, and G. Nehmetallah, "Automatic phase aberration compensation for digital holographic microscopy based on deep learning background detection," Opt. Express **25**, 15043–15057 (2017).
248. T. Liu, K. de Haan, Y. Rivenson, Z. Wei, X. Zeng, Y. Zhang, and A. Ozcan, "Deep learning-based super-resolution in coherent imaging systems," arXiv:1810.06611 (2018).
249. T. Nguyen, Y. Xue, Y. Li, L. Tian, and G. Nehmetallah, "Deep learning approach for Fourier ptychography microscopy," Opt. Express **26**, 26470–26484 (2018).
250. L. Tian and L. Waller, "3D intensity and phase imaging from light field measurements in an LED array microscope," Optica **2**, 104–111 (2015).
251. 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**, 517–522 (2015).
252. J. Van Roey, J. Van der Donk, and P. E. Lagasse, "Beam propagation method: analysis and assessment," J. Opt. Soc. Am. **71**, 803–810 (1981).
253. S. Jiang, K. Guo, J. Liao, and G. Zheng, "Solving Fourier ptychographic imaging problems via neural network modeling and TensorFlow," Biomed. Opt. Express **9**, 3306–3319 (2018).
254. A. Van der Schaaf and J. H. van Hateren, "Modelling the power spectra of natural images: statistics and information," Vision Res. **36**, 2759–2770 (1996).
255. J. Pan, S. Liu, D. Sun, J. Zhang, Y. Liu, J. Ren, Z. Li, J. Tang, H. Lu, Y.-W. Tai, and M.-H. Yang, "Learning dual convolutional neural networks for low-level vision," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).
256. M. Deng, S. Li, and G. Barbastathis, "Learning to synthesize: splitting and recombining low and high spatial frequencies for image recovery," arXiv:1811.07945 (2018).
257. H. Malm, M. Oskarsson, E. Warrant, P. Clarberg, J. Hasselgren, and C. Lejdfors, "Adaptive enhancement and noise reduction in very low light-level video," in *International Conference on Computer Vision (ICCV)* (2007), pp. 1631–1638.
258. X. Zhang, P. Shen, L. Luo, L. Zhang, and J. Song, "Enhancement and noise reduction of very low light level images," in *International Conference on Pattern Recognition* (2012), pp. 2034–2037.
259. A. Loza, D. R. Bull, P. R. Hill, and A. M. Achim, "Automatic contrast enhancement of low-light images based on local statistics of wavelet coefficients," Digital Signal Process. **23**, 1856–1866 (2013).
260. Y. Chen, W. Yu, and T. Peck, "On learning optimized reaction diffusion processes for effective image restoration," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 5261–5269.
261. S. W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. T. Barron, F. Kainz, J. Chen, and M. Levoy, "Burst photography for high dynamic range and low-light imaging on mobile cameras," ACM Trans. Graph. **35**, 192 (2016).
262. T. Plotz and S. Roth, "Benchmarking denoising algorithms with real photographs," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 1586–1595.
263. T. Remez, O. Litany, R. Giryes, and A. M. Bronstein, "Deep convolutional denoising of low-light images," arXiv:1701.01687v1 (2017).
264. K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," IEEE Trans. Image Proc. **16**, 2080–2095 (2007).
265. A. Goy, K. Arthur, S. Li, and G. Barbastathis, "Low photon count phase retrieval using deep learning," Phys. Rev. Lett. **121**, 243902 (2018).
266. T. B. Pittman, Y. H. Shih, D. V. Strekalov, and A. V. Sergienko, "Optical imaging by means of two-photon quantum entanglement," Phys. Rev. A **52**, R3429–R3432 (1995).
267. A. Gatti, E. Brambilla, M. Bache, and L. A. Lugiato, "Correlated imaging, quantum and classical," Phys. Rev. A **70**, 013802 (2004).
268. A. Gatti, E. Brambilla, M. Bache, and L. A. Lugiato, "Ghost imaging with thermal light: comparing entanglement and classical correlation," Phys. Rev. Lett. **93**, 093602 (2004).
269. A. Valencia, G. Scarcelli, M. D'Angelo, and Y. H. Shih, "Two-photon imaging with thermal light," Phys. Rev. Lett. **94**, 063601 (2005).
270. F. Ferri, D. Magatti, A. Gatti, M. Bache, E. Brambilla, and L. A. Lugiato, "High-resolution ghost image and ghost diffraction experiments with thermal light," Phys. Rev. Lett. **94**, 183602 (2005).
271. B. I. Erkmen and J. H. Shapiro, "Unified theory of ghost imaging with Gaussian-state light," Phys. Rev. A **77**, 043809 (2008).
272. R. Meyers, K. S. Deacon, and Y. Shih, "Ghost-imaging experiment by measuring reflected photons," Phys. Rev. A **77**, 041801 (2008).
273. J. H. Shapiro, "Computational ghost imaging," Phys. Rev. A **78**, 061802 (2008).
274. P. A. Morris, R. S. Aspden, J. E. Bell, R. W. Boyd, and M. J. Padgett, "Imaging with a small number of photons," Nat. Commun. **6**, 5913 (2015).
275. J. Cheng, "Ghost imaging through turbulent atmosphere," Opt. Express **17**, 7916–7921 (2009).
276. B. I. Erkmen and J. H. Shapiro, "Ghost imaging: from quantum to classical to computational," Adv. Opt. Photon. **2**, 405–450 (2010).
277. O. Katz, Y. Bromberg, and Y. Silberberg, "Compressive ghost imaging," Appl. Phys. Lett. **95**, 131110 (2009).
278. C. Zhao, W. Gong, M. Chen, E. Li, H. Wang, W. Xu, and S. Han, "Ghost imaging lidar via sparsity constraints," Appl. Phys. Lett. **101**, 141123 (2012).
279. M. Lyu, W. Wang, H. Wang, H. Wang, G. Li, N. Chen, and G. Situ, "Deep-learning-based ghost imaging," Sci. Rep. **7**, 17865 (2017).
280. V. I. Tatarski, *Wave Propagation in a Turbulent Medium* (McGraw-Hill, 1961).
281. A. Ishimaru, *Wave Propagation and Scattering in Random Media* (Academic Press 1978; reissued by Oxford University Press and IEEE Press, 1997).
282. S. Popoff, G. Lerosey, R. Carminati, M. Fink, A. Boccara, and S. Gigan, "Measuring the transmission matrix in optics: an approach to the study and control of light propagation in disordered media," Phys. Rev. Lett. **104**, 100601 (2010).

283. S. Popoff, G. Lerosey, M. Fink, A. Boccara, and S. Gigan, "Image transmission through an opaque material," Nat. Commun. **1**, 81 (2010).
284. A. Drémeau, A. Liutkus, D. Martina, O. Katz, C. Schülke, F. Krzakala, S. Gigan, and L. Daudet, "Reference-less measurement of the transmission matrix of a highly scattering material using a DMD and phase retrieval techniques," Opt. Express **23**, 11898–11911 (2015).
285. S. Feng, C. Kane, P. A. Lee, and A. D. Stone, "Correlations and fluctuations of coherent wave transmission through disordered media," Phys. Rev. Lett. **61**, 834–837 (1988).
286. I. Freund, M. Rosenbluh, and S. Feng, "Memory effects in propagation of optical waves through disordered media," Phys. Rev. Lett. **61**, 2328–2331 (1988).
287. J. Bertolotti, E. G. van Putten, C. Blum, A. Lagendijk, W. L. Vos, and A. P. Mosk, "Non-invasive imaging through opaque scattering layers," Nature **491**, 232–234 (2012).
288. O. Katz, P. Heidmann, M. Fink, and S. Gigan, "Non-invasive single-shot imaging through scattering layers and around corners via speckle correlations," Nat. Photonics **8**, 784–790 (2014).
289. N. Stasio, C. Moser, and D. Psaltis, "Calibration-free imaging through a multicore fiber using speckle scanning microscopy," Opt. Lett. **41**, 3078–3081 (2016).
290. A. Porat, E. R. Andresen, H. Rigneault, D. Oron, S. Gigan, and O. Katz, "Widefield lensless imaging through a fiber bundle via speckle correlations," Opt. Express **24**, 16835–16855 (2016).
291. "Caltech computer vision database," http://www.vision.caltech.edu/archive.html.
292. V. N. Vapnik, *The Nature of Statistical Learning Theory* (Springer, 1995).
293. A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," Stat. Comput. **14**, 199–222 (2004).
294. M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer, "Theoretical foundations of the potential function method in pattern recognition learning," Autom. Remote Control **25**, 917–936 (1964).
295. T. Hofmann, B. Scholkoff, and A. J. Smola, "Kernel methods in machine learning," Ann. Statist. **36**, 1171–1220 (2008).
296. S. S. Keerthi, S. K. Shevade, C. Bhattacharya, and K. R. K. Murty, "Improvements to Platt's SMO algorithm for SVM classifier design," Neural Comput. **13**, 637–649 (2001).
297. B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Annual Conference on Computational Learning Theory* (ACM, 1992), pp. 144–152.
298. T. Poggio, "On optimal nonlinear associative recall," Biol. Cybern. **19**, 201–209 (1975).
299. M. Lyu, H. Wang, G. Li, and G. Situ, "Exploit imaging through opaque wall via deep learning," arXiv:1708.07881 (2017).
300. M. Lyu, H. Wang, G. Li, S. Zheng, and G. Situ, "Learning-based lensless imaging through optically thick scattering media," Adv. Photon. **1**, 036002 (2019).
301. A. Kirmani, T. Hutchinson, J. Davis, and R. Raskar, "Looking around the corner using ultrafast transient imaging," Int. J. Comput. Vision **95**, 13–28 (2011).
302. O. Gupta, T. Willwacher, A. Velten, A. Veeraraghavan, and R. Raskar, "Reconstruction of hidden 3d shapes using diffuse reflections," Opt. Express **20**, 19096–19108 (2012).
303. A. Velten, "Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging," Nat. Commun. **3**, 745 (2012).
304. M. Buttafava, J. Zeman, A. Tosi, K. Eliceiri, and A. Velten, "Non-line-of-sight imaging using a time-gated single photon avalanche diode," Opt. Express **23**, 20997–21011 (2015).
305. G. Gariepy, F. Tonolini, R. Henderson, J. Leach, and D. Faccio, "Detection and tracking of moving objects hidden from view," Nat. Photonics **10**, 23–26 (2016).
306. J. Klein, C. Peters, J. Martin, M. Laurenzis, and M. B. Hullin, "Tracking objects outside the line of sight using 2D intensity images," Sci. Rep. **6**, 32491 (2016).
307. A. Kadambi, H. Zhao, B. Shi, and R. Raskar, "Occluded imaging with time-of-flight sensors," ACM Trans. Graph. **35**, 1–12 (2016).
308. S. Chan, R. E. Warburton, G. Gariepy, J. Leach, and D. Faccio, "Non-line-of-sight tracking of people at long range," Opt. Express **25**, 10109–10117 (2017).
309. M. O'Toole, D. B. Lindell, and G. Wetzstein, "Confocal non-line-of-sight imaging based on the light-cone transform," Nature **555**, 338–341 (2018).
310. A. L. Cohen, "Anti-pinhole imaging," Opt. Acta **29**, 63–67 (1982).
311. O. Katz, E. Small, and Y. Silberberg, "Looking around corners and through thin turbid layers in real time with scattered incoherent light," Nat. Photonics **6**, 549–553 (2012).
312. A. Torralba and W. T. Freeman, "Accidental pinhole and pinspeck cameras: revealing the scene outside the picture," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2012), pp. 374–381.
313. K. L. Bouman, "Turning corners into cameras: principles and methods," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 2270–2278.
314. G. Satat, M. Tancik, O. Gupta, B. Heshmat, and R. Raskar, "Object classification through scattering media with deep learning on time resolved measurement," Opt. Express **25**, 17466–17479 (2017).
315. P. Caramazza, A. Boccolini, D. Buschek, M. Hullin, C. F. Higham, R. Henderson, R. Murray-Smith, and D. Faccio, "Neural network identifcation of people hidden from view with a single-pixel, single-photon detector," Sci. Rep. **8**, 11945 (2018).